Doctoral thesis at NTNU, 2011:70

Pål Liljebäck

Modelling, Development, and Control of Snake Robots

NTNU – Trondheim
Norwegian University of
Science and Technology

Doctoral Thesis

Pål Liljebäck

NTNU
Norwegian University of Science and Technology
Thesis for the degree of Philosophiae Doctor
Faculty of Information Technology,
Mathematics and Electrical Engineering
Department of Engineering Cybernetics

Doctoral thesis at NTNU, 2011:70

NTNU

NTNU – Trondheim
Norwegian University of
Science and Technology

ISBN 978-82-471-2667-7 (printed version)
ISBN 978-82-471-2668-4 (electronic version)
ISSN 1503-8181

Pål Liljebäck

Modelling, Development, and Control of Snake Robots

Thesis for the degree of Philosophiae Doctor

Trondheim, February 2011

Norwegian University of Science and Technology
Faculty of Information Technology,
Mathematics and Electrical Engineering
Department of Engineering Cybernetics

NTNU – Trondheim
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology,
Mathematics and Electrical Engineering
Department of Engineering Cybernetics

ISBN 978-82-471-2667-7 (printed version)
ISBN 978-82-471-2668-4 (electronic version)
ISSN 1503-8181

ITK report 2011-3-W

Doctoral thesis at NTNU, 2011:70

Printed by Skipnes Kommunikasjon as

Summary

This thesis is motivated by the long-term goal of developing snake robots which can move in unknown and challenging environments in order to support human intervention tasks. Inspired by biological snakes, snake robots typically consist of a large number of serially connected joint modules which provide stable and robust locomotion skills. The aim of this thesis is to increase our basic understanding of snake robot locomotion and to present new control strategies for these mechanisms. The thesis contains two parts.

Part I

Part I considers planar snake robot locomotion across horizontal and flat surfaces. The treatment of this control problem is based on two different mathematical models of the snake robot. The first model is developed from first principles, while the second model is developed based on simplifying assumptions.

The models are analysed using several techniques. A stabilizability and controllability analysis reveals fundamental properties of snake robot dynamics which have been assumed in the snake robot literature, but never formally proven. Several properties are also derived directly from the equations of motion of the robot, some of which explain why anisotropic ground friction properties are important during flat surface locomotion. An averaging analysis reveals important relationships between specific gait pattern parameters of the robot and the resulting velocity during locomotion.

Two controllers for straight line path following control are proposed. By analysing a Poincaré map, all state variables of a snake robot, except for the position along the desired straight path, are shown to trace out an exponentially stable periodic orbit with the first controller. Using cascaded systems theory, the second controller is proved to $\mathcal{K}$-exponentially stabilize a snake robot to any desired straight path under the assumption that the forward velocity is contained in some nonzero and positive interval. More-

over, a waypoint guidance strategy is proposed for steering a snake robot along more general paths defined by waypoints interconnected by straight lines. The guidance strategy is guaranteed to steer the position of the robot into the acceptance region of each waypoint. Furthermore, it is outlined how the controllers can be extended to path following of general curved paths.

The development of a snake robot for flat surface motion is presented, and the robot is employed to experimentally validate many of the theoretical results.

Part II

Part II considers snake robot locomotion in environments containing external objects (or obstacles), which is in line with practical applications of these mechanisms. A hybrid model of a planar snake robot interacting with obstacles is presented, where obstacle interaction is modelled by introducing a unilateral velocity constraint on each contacted link of the robot. The existence and uniqueness properties of the hybrid model are investigated based on the theory of linear complementarity problems.

The control problem in this obstacle environment is attacked based on the notion of obstacle-aided locomotion, where the snake robot is propelled by active use of the contact forces from the obstacles. A general control principle is proposed which suggests that obstacle-aided locomotion is achieved by producing body shape changes where the links in contact with obstacles are rotated to increase the propulsive forces on the robot. This control principle is employed to propose a hybrid controller aimed at resolving situations where the snake robot is jammed between obstacles. The control principle is also instrumental in the development of a straight line path following controller in the obstacle environment, where the motion of the snake robot is specified in terms of a body wave component, an environment adaptation component, and a heading control component.

The development of a snake robot for motion in unstructured environments is presented. The robot, which can sense its environment due to a novel force sensing system, is employed to experimentally investigate the proposed control strategies.

Preface

I am inevitably, relentlessly, and inescapably fascinated by snake robots. A driving force of my research interests is the dream of one day, and at first hand, witnessing their potential being realized. I feel privileged to have been given the opportunity to study and work with these mechanisms.

This thesis presents results from my PhD research on snake robots, which has been carried out in the period January 2008 to December 2010 at the Department of Engineering Cybernetics at the Norwegian University of Science and Technology (NTNU) in Trondheim, Norway. The research has been conducted in close cooperation with the Norwegian research organization SINTEF. Funding for the research has been provided by a PhD scholarship at NTNU, combined with financial support from SINTEF. I gratefully acknowledge the support from these institutions.

I am indebted to a number of people who have been integral to the completion of this PhD study. I express my deepest gratitude to my supervisors Professor Kristin Ytterstad Pettersen, Dr. Øyvind Stavdahl, and Professor Jan Tommy Gravdahl for their invaluable support and encouragement. I thank Kristin for teaching me the value and importance of a solid theoretical basis when faced with practical problems, for her always brilliant and skilful contributions during my research, and for her amazing ability to provide accurate feedback in order to raise the quality of scientific publications. I thank Øyvind for introducing me to the field of snake robotics during my MSc studies, for contributing with his many ingenious ideas, and for never ceasing to amaze me with his ability to instantly see a solution to (almost) any given problem. I thank Jan Tommy for his encouragement and positivity, and for his bright ideas and inputs provided during our meetings.

I express my sincere gratitude to Professor Shigeo Hirose and Dr. Hiroya Yamada at Tokyo Institute of Technology in Japan for their kind hospitality during my visit at their university for two months in 2008, which was a truly inspiring experience for me.

I gratefully acknowledge all the support I've received from my colleagues at the Department of Engineering Cybernetics at NTNU. In particular, I thank Idar Haugstuen for our cooperation in conjunction with his MSc project on snake robots in 2009/2010. I greatly appreciate our many interesting and fruitful discussions during this period. Furthermore, all my past and current PhD colleagues that I've had the privilege of getting to know are thanked for creating a diverse and stimulating work environment, and for our many social interactions. I also thank Alexey Pavlov for our talks and for his many useful suggestions for my research. For their untiring help and efforts with the experimental systems employed in my PhD work, I thank Terje Haugen, Per Inge Snildal, and Glenn Angel at the mechanical workshop of the department. I also thank Stefano Bertelli for his positive spirit and for his help with documenting my experimental results.

I owe many thanks to my dear colleagues at SINTEF Applied Cybernetics for their support and for contributing to a positive and stimulating work environment. In particular, I thank the Research Director, Sture Holmstrøm, for motivating and supporting my decision of taking on this PhD study, and for his willingness to financially support research on snake robots. I am thankful to Aksel A. Transeth, Sigurd Fjerdingen, and Erik Kyrkjebø for their friendship, their enthusiasm, and for our many interesting discussions related to snake robots over the last years. My good friend Aksel deserves special recognition for sharing much of his experience from his own doctoral studies, and for supporting my research with his knowledge and expertise of snake robots. I thank Geir Mathisen, Espen Helle, and Knut Vidar Skjersli for their work on circuit boards and software for the experimental systems that I have employed in my studies. I am grateful to Anders Beitnes for conceiving the idea of a self-propelled fire hose, which initiated research on snake robots at SINTEF and NTNU. I also thank Wheeko and Kulko for their outstanding and obedient performance during my experiments, and for never complaining about late work hours.

Finally, this thesis never would have taken form without the loving support of my family. I am eternally indebted to my parents, Anna and Jan Liljebäck, my sister, Hanne, and my two brothers, Lars and Kjetil, for always being there for me and for helping me pursue my dreams. My final and most profound gratitude goes to my dear girlfriend and fiancée, Tonje Smedbråten, for her love and support, which mean everything to me. You are my favorite person in the whole world.

Trondheim, December 2010 Pål Liljebäck

Publications

The work underlying this thesis has produced the following publications.

Journal Papers (First Author)

- P. Liljebäck, Ø. Stavdahl, K. Y. Pettersen, and J. T. Gravdahl, "Two new design concepts for snake robot locomotion in unstructured environments," *Paladyn. Journal of Behavioral Robotics*, 2011. To appear.

- P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "Snake robot locomotion in environments with obstacles," *IEEE Transactions on Mechatronics*, 2010. Submitted.

- P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "A review on modelling, implementation, and control of snake robots," *Robotics and Autonomous Systems*, 2010. Submitted.

- P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "A simplified model and fundamental properties of snake robot locomotion," *Robotica*, 2011. To be submitted.

- P. Liljebäck, I. U. Haugstuen, and K. Y. Pettersen, "Path following control of planar snake robots using a cascaded approach," *IEEE Transactions on Control Systems Technology*, 2011. To appear.

- P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "Experimental investigation of obstacle-aided locomotion with a snake robot," *IEEE Transactions on Robotics*, 2010. Conditionally accepted.

- P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "Controllability and stability analysis of planar snake robot locomotion," *IEEE Transactions on Automatic Control*, 2011. To appear.

- P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "Hybrid modelling and control of obstacle-aided snake robot locomotion," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 781-799, Oct 2010.

- P. Liljebäck, Ø. Stavdahl, and K. Y. Pettersen, "Modular pneumatic snake robot: 3D modelling, implementation and control," *Modeling, Identification and Control (MIC)*, vol. 29, no. 1, pp. 21-28, Jan 2008.

Conference Papers (First Author)

- P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "Path following control of snake robots in unstructured environments," in *Proc. IEEE Int. Conf. Robotics and Automation*, Shanghai, China, 2011. Accepted.

- P. Liljebäck and K. Y. Pettersen, "Waypoint guidance control of snake robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, Shanghai, China, 2011. Accepted.

- P. Liljebäck, I. U. Haugstuen, and K. Y. Pettersen, "Experimental investigation of a path following controller for planar snake robots," in *Proc. IEEE Int. Conf. Control, Automation, Robotics, and Vision (ICARCV)*, Singapore, 2010, pp. 2325-2332.

- P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "Experimental investigation of fundamental properties of snake robot locomotion," in *Proc. IEEE Int. Conf. Control, Automation, Robotics, and Vision (ICARCV)*, Singapore, 2010, pp. 187-194. Finalist for the Best Paper Award.

- P. Liljebäck, I. U. Haugstuen, and K. Y. Pettersen, "Path following control of planar snake robots using a cascaded approach," in *Proc. IEEE Conf. Decision and Control*, Atlanta, GA, USA, 2010, pp. 1969-1976.

- P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "Stability analysis of snake robot locomotion based on averaging theory," in *Proc. IEEE Conf. Decision and Control*, Atlanta, GA, USA, 2010, pp. 1977-1984.

- P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "Fundamental properties of snake robot locomotion," in *Proc. IEEE/RSJ*

Int. Conf. Intelligent Robots and Systems, Taipei, Taiwan, 2010, pp. 2876-2883.

- P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "A simplified model of planar snake robot locomotion," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Taipei, Taiwan, 2010, pp. 2868-2875.

- P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "A hybrid model of obstacle-aided snake robot locomotion," in *Proc. IEEE Int. Conf. Robotics and Automation*, Anchorage, AK, USA, 2010, pp. 675-682.

- P. Liljebäck, K. Y. Pettersen, and Ø. Stavdahl, "A snake robot with a contact force measurement system for obstacle-aided locomotion," in *Proc. IEEE Int. Conf. Robotics and Automation*, Anchorage, AK, USA, 2010, pp. 683-690.

- P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "Controllability analysis of planar snake robots influenced by viscous ground friction," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, St. Louis, MO, USA, 2009, pp. 3615-3622.

- P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "Stability analysis of snake robot locomotion based on Poincaré maps," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, St. Louis, MO, USA, 2009, pp. 3623-3630.

- P. Liljebäck, S. Fjerdingen, K. Y. Pettersen, and Ø. Stavdahl, "A snake robot joint mechanism with a contact force measurement system," in *Proc. IEEE Int. Conf. Robotics and Automation*, Kobe, Japan, 2009, pp. 3815-3820.

- P. Liljebäck, K. Y. Pettersen, and Ø. Stavdahl, "Modelling and control of obstacle-aided snake robot locomotion based on jam resolution," in *Proc. IEEE Int. Conf. Robotics and Automation*, Kobe, Japan, 2009, pp. 3807-3814.

- P. Liljebäck, Ø. Stavdahl, and A. Beitnes, "SnakeFighter - Development of a water hydraulic fire fighting snake robot," in *Proc. IEEE Int. Conf. Control, Automation, Robotics, and Vision (ICARCV)*, Singapore, Dec 2006.

- P. Liljebäck, Ø. Stavdahl, and K. Y. Pettersen, "Modular pneumatic snake robot: 3D modelling, implementation and control," in *Proc. 16th IFAC World Congress*, Prague, Czech Republic, July 2005.

Journal Papers (Co-author)

- A. A. Transeth, K. Y. Pettersen, and P. Liljebäck, "A survey on snake robot modeling and locomotion," *Robotica*, vol. 27, pp. 999-1015, 2008.

- A. A. Transeth, R. I. Leine, C. Glocker, K. Y. Pettersen, and P. Liljebäck, "Snake robot obstacle aided locomotion: Modeling, simulations, and experiments," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 88-104, Feb 2008.

Conference Papers (Co-author)

- A. A. Transeth, P. Liljebäck, S. Fjerdingen, E. Kyrkjebø, and T. Mugaas, "New possibilities - Next generation robotic systems for inspection and maintenance operations," in Proc. EuroMaintenance, Fiera di Verona, Italy, 2010, pp. 229-232.

- S. A. Fjerdingen, P. Liljebäck, and A. A. Transeth, "A snake-like robot for internal inspection of complex pipe structures (PIKo)," in Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, St. Louis, MO, USA, 2009, pp. 5665-5671.

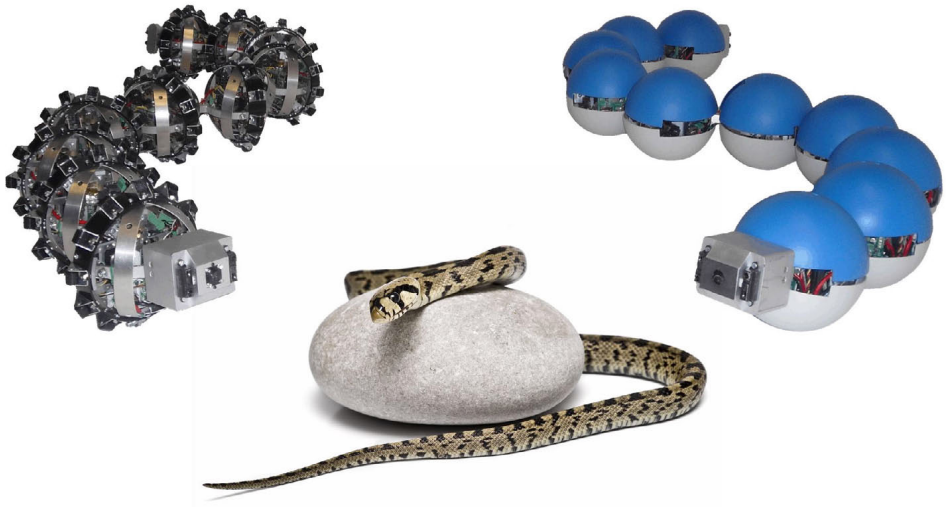
- E. Kyrkjebø, P. Liljebäck, and A. A. Transeth, "A robotic concept for remote inspection and maintenance on oil platforms," in Proc. ASME 28th Int. Conf. Ocean, Offshore and Arctic Engineering (OMAE), Honolulu, HI, USA, 2009.

- A. A. Transeth, P. Liljebäck, and K. Y. Pettersen, "Snake robot obstacle aided locomotion: An experimental validation of a non-smooth modeling approach," in Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, San Diego, CA, USA, Oct-Nov 2007, pp. 2582-2589.

Contents

Chapter 1

Introduction



1.1 Background and Motivation

Research on snake robots at NTNU has spawned from a research project at SINTEF[1]. The project was initiated in 2003 after several major city fires in Trondheim, which launched an initiative to bring the fire department in closer relation with the research community in Trondheim to stimulate efforts that would improve fire safety. A specific idea which spurred from this initiative was the vision of a self-propelled fire hose as a robotic tool to

[1]SINTEF is a Norwegian research organization which is tightly coupled with NTNU both geographically and through joint research activities.

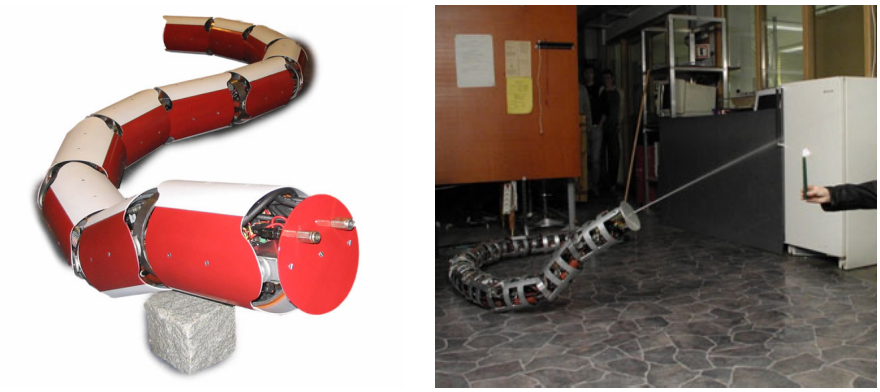


Figure 1.1: The water hydraulic snake robot *Anna Konda*.

aid human firefighters. This idea is clever in that the high-pressure water inside the hose can be employed as a *hydraulic* medium in the propulsion mechanism, a *fire extinguishing* medium, and a *cooling* medium for cooling the robot in environments with extreme temperatures. The resulting system would be a robotic fire hose that could move in extreme environments with the agility of a biological snake, or, in other words, a water hydraulic snake robot. The Applied Cybernetics department at SINTEF was brought in to investigate this idea further, and so began the research activity on snake robots at SINTEF and NTNU.

Researchers at SINTEF and NTNU quickly realized that developing such a mechanism represents a highly interdisciplinary task with challenges ranging from heat resistant materials and water hydraulic joint actuation to control design and human-machine interaction. To show the feasibility of the concept, it was decided to develop a simple technology demonstrator in the form of a water hydraulic snake robot. The robot, which was named *Anna Konda*, is shown in Fig. 1.1 and is described in more detail in Liljebäck *et al.* (2006). Anna Konda can move over relatively flat surfaces and can spray water through nozzles in its head. The robot is, however, far from ready for operating in harsh environments.

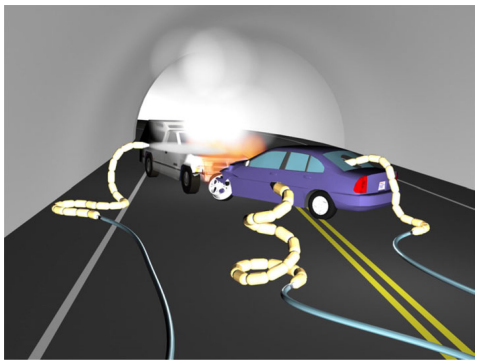
The work on the Anna Konda robot helped us identify several major research challenges. The critical and most significant research challenge was (and still is) the serpentine propulsion mechanism of this system. After the development of Anna Konda, the research on snake robots at NTNU and SINTEF has therefore targeted snake robot locomotion in general without concern about the specific application of the robot. Although fire fighting was the initial motivation behind this research, the scope of the current

research activities extends beyond merely fire intervention tasks since snake robots can be used in many other applications where robust robotic mobility is required. Fig. 1.2 illustrates a few potential applications of snake robots.

In a global perspective, research on snake robots has been conducted for several decades. These mechanisms typically consist of serially connected joint modules capable of bending in one or more planes. The world's first snake robot was developed by the Japanese Professor Shigeo Hirose as early as 1972 (see Hirose, 1993). Since that time, several agile and impressive mechanisms have been developed by research communities around the world in efforts to mimic the motion capabilities of their biological counterpart. However, the locomotive capabilities of current snake robots are still limited to controlled lab environments, and the world has not yet seen practical applications of snake robot locomotion. Nonetheless, researchers working with snake robots know that harnessing and overcoming the challenges of snake robot locomotion means developing a robotic propulsion mechanism which far surpasses the mobility of the more conventional wheeled, tracked and legged forms of robotic mobility.

Development and control of snake robots is generally quite challenging for two primary reasons. First of all, a snake robot has many degrees of freedom, which means that the physical mechanism will contain a complex interconnection of sensors, actuators, and control logic. Moreover, the many degrees of freedom represent complex nonlinear dynamics which is challenging to analyse from a control design perspective. Second of all, the dependence on environment interaction is more complicated for a snake robot than for more conventional mobile robots. In particular, the propulsion mechanism of a wheeled, tracked or legged robot is achieved with a separate and dedicated part of the robot. A snake robot, on the other hand, has no separate part which is dedicated to propulsion. Being essentially a smooth and flexible manipulator arm, the propulsion mechanism of a snake robot is rather an integrated part of the entire body, which means that propulsion requires synchronized motion of the entire robot in order to produce appropriate environment interaction forces. Motion based on such environment interaction is challenging both with respect to control design and mechanical design.

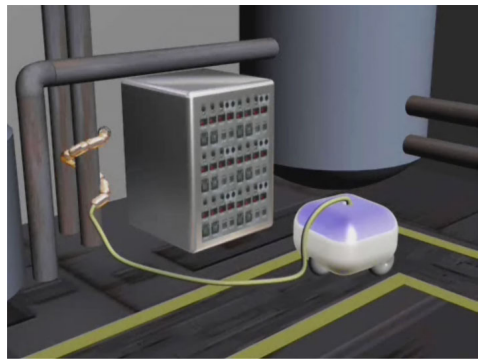
This thesis targets some of the challenges discussed above, and is motivated by the long-term goal of developing snake robots which can move in unknown and challenging environments in order to support human intervention tasks. The main goal of the thesis is to increase our basic understanding of snake robot locomotion. To this end, the focus of the thesis



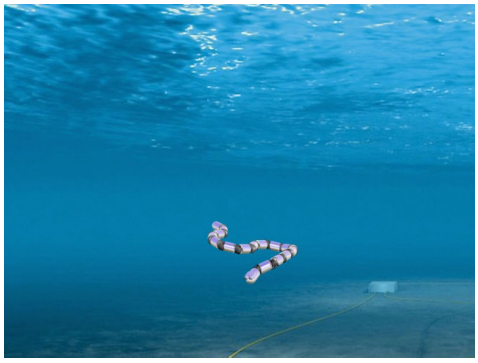
(a) Fire fighting.

(b) Explosion prevention.

(c) Search and rescue operations.



(d) Inspection and maintenance.



(e) Subsea operations.

(f) Domestic applications.

Figure 1.2: Applications of snake robots.

is primarily directed towards control design. Efficient control strategies are vital to future applications of snake robots, and are also instrumental in the development of these mechanisms. In particular, a control strategy with a solid mathematical foundation will immediately reveal what sensory capabilities, ground friction properties, actuator forces, etc., that are required in the physical robot to achieve a specific control objective. It is our hope that the results presented in this thesis will stimulate and support future research on these fascinating mechanisms.

1.2 Biological Snakes

This thesis is inspired by the robust motion capabilities of biological snakes. These amazing creatures are optimal in the sense that they have emerged through millions of years of evolution. In the following, we present aspects of biological snakes that we consider relevant to modelling, development, and control of snake robots. The material is based on Bauchot (1994); Hirose (1993); Hu *et al.* (2009); Mattison (2002).

1.2.1 The Anatomy of Snakes

The typical appearance of the skeletal structure of a snake is shown in Fig. 1.3 and consists of vertebrae, ribs, and a skull. Snakes can have between 130 and 500 vertebrae, with ribs attached to each one. The vertebrae constitute a column of movable joints that run through the body of the snake, and protects the spinal cord, which runs through a channel along the top of the vertebral column. The ribs attached to each side of a vertebra protect the internal organs.

The mechanical interconnection of the vertebrae is interesting. As illustrated in Fig. 1.4, two vertebrae are connected in a ball and socket arrangement. The magnitude of the relative rotational motion between two vertebrae is quite limited. In particular, the relative rotation between two vertebrae about the vertical axis ranges between 10° and 20°, while the relative rotation about the horizontal axis is limited to only a few degrees. These limitations may appear contradictory to the flexibility that snakes are known for, but this flexibility is, in fact, produced by the sum of the small movements of many vertebrae. Moreover, limiting the range of the relative movements leads to increased strength in the connection between the vertebrae. To prevent damage to the spinal cord due to twisting of the vertebrae about the axis tangential to the body, each vertebra has a num-

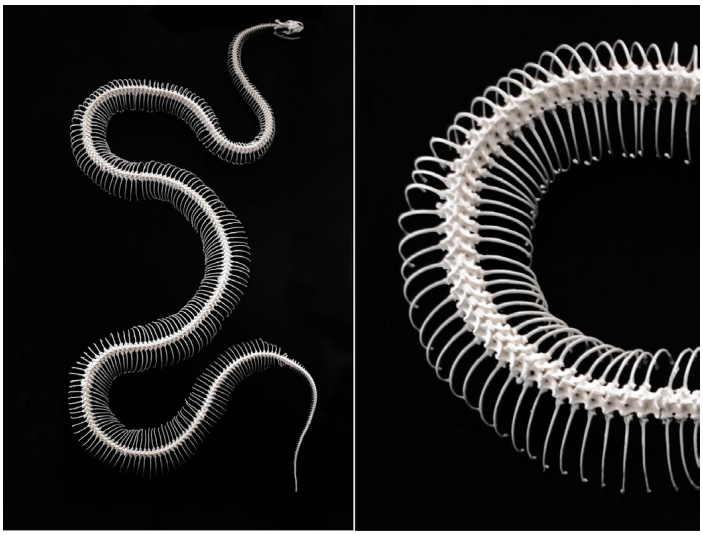


Figure 1.3: The skeleton of a snake consisting of vertebrae, ribs, and a skull. Image source: www.shutterstock.com.

ber of wing-like projections that interlock loosely with their counterparts on the adjacent vertebrae. This limits the amount of twisting.

The body shape of a snake is changed with the help of muscles that are arranged diagonally along each side of the snake. The ends of these muscles are attached to ribs, sometimes joining adjacent ribs, but mostly joining ribs that are some distance apart. The pattern of contraction and relaxation of these muscles determines the type of locomotion that is performed. For instance, if muscles on one side of the snake are contracted at the same time as the equivalent muscles on the other side are relaxed, then the body will be bent. If, on the other hand, opposite sets of muscles are contracted or relaxed simultaneously, then the snake will, to some extent, be able to shorten or extend its body at this location.

The skin of a snake is completely covered with scales. The scales are formed from thickened areas of the skin and are therefore integrated with the skin. The typical appearance of snake scales is shown to the left of Fig. 1.5, while the right shows the skin when it is stretched, thereby pulling the scales apart. The areas of skin between the scales allow the snake to flex its body while maintaining a smooth coverage of the scales. An important purpose of the scales is to form a physical protection from general wear and tear when the snake moves across rough surfaces. At the same time, the

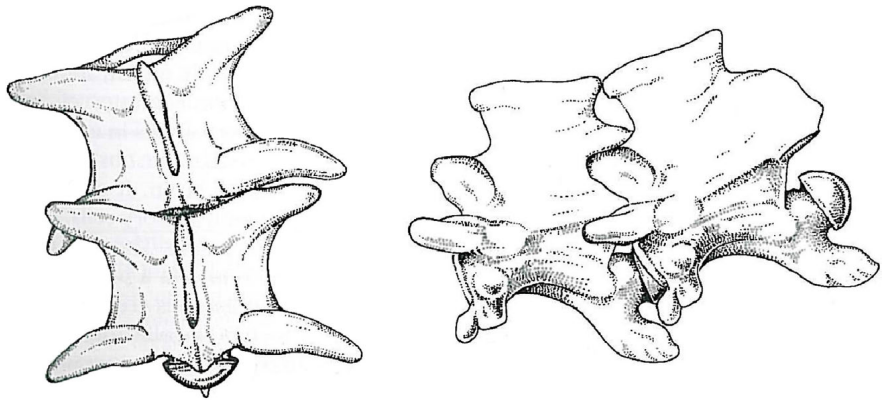


Figure 1.4: Two loosely interlocked vertebrae of a snake. Image source: Mattison (2002). By permission of Chris Mattison.

use of small units of armour allows greater flexibility than would large bony plates. Another feature of the scales is that they give the snake *anisotropic* ground friction properties, i.e. the scales give the snake a larger friction coefficient in the transversal direction of the snake body compared to in the tangential direction. Studies of biological snakes and simulation studies have indicated that this difference in the friction coefficients is important during forward gliding motion. The assumption on the importance of this friction property is proved in this thesis (Section 4.3).

1.2.2 The Locomotion of Snakes

Snakes are almost unique among the terrestrial vertebrates in their lack of legs. However, the lack of legs do not appear to have placed restrictions on the ability of snakes to move around. On the contrary, snake locomotion is stable, robust, and versatile. The speed of snake locomotion is, however, relatively slow, although certain species can move at speeds up to 11 km/h. Snakes display four basic types of locomotion, which are described below. More specialized forms of motion also exist. For instance, certain snakes can jump to heights of up to 1 m by curving their body into a vertical S-shape to serve as a spring, and then jump by stretching their body. Other snakes are able to fly through the air by throwing themselves from trees and forming their body in an aerodynamically favourable manner. The four most common types of snake locomotion are now presented.

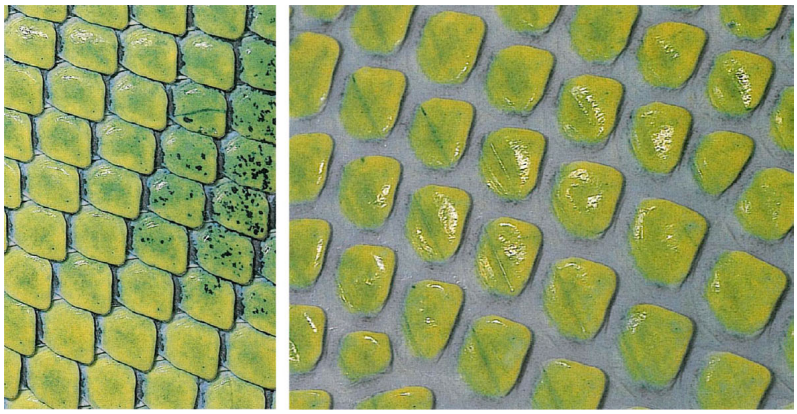


Figure 1.5: The skin of a snake is completely covered with scales, which are formed from thickened areas of the skin. In the picture on the right, the skin is stretched, pulling the scales apart. Image source: Mattison (2002). By permission of Chris Mattison.

Lateral Undulation

Lateral undulation, also called serpentine crawling, is the fastest and most common form of snake locomotion, and is illustrated in Fig. 1.6(a). During lateral undulation, continuous waves are propagated backwards along the snake body from head to tail. During this wave motion, the sides of the snake body push against irregularities in the surface, thereby pushing the snake forward. This form of locomotion is therefore not suitable on slippery surfaces. As the snake progresses, every point along the body passes the same point on the ground, and there is never any static contact between the ground and any point along the body. During swimming, the same wave motion is produced, but the body then pushes against the resistance of the water. The weight distribution of a snake during lateral undulation is not uniform, but rather distributed so that the peaks of the body wave curve are slightly lifted from the ground.

Concertina Locomotion

Concertina locomotion, which is illustrated in Fig. 1.6(b), is often employed in narrow spaces where the available range of motion is limited. The motion is carried out by first extending the front part of the body forward while the back part is curved several times to provide an anchor against the narrow

environment. Once the head and front part of the body are fully extended, they are subsequently used to provide an anchor in the same way so that the back part of the body can be drawn up. The sequence is then repeated.

The principle behind concertina locomotion relies on the difference between the large static friction forces at the anchor points and the low kinetic friction forces in the part of the body which is extended. The motion pattern is not very efficient in terms of energy consumption, but is often needed in order to traverse tight spaces.

Rectilinear Crawling

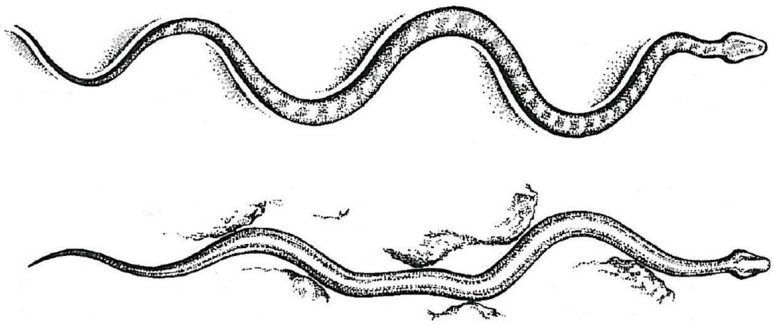
Rectilinear crawling is a slow form of locomotion often employed by heavy-bodied snakes. Also snakes in the final stages of stalking their pray use rectilinear crawling to avoid alerting their intended victim. During rectilinear crawling, the snake uses the edges of the scales on its underside as anchor points to pull itself forward in a more or less straight line. The operation consists of stretching forward and hooking the edges of the scales over small irregularities, then pulling the body up to this point. Alternate parts of the body will be stretching and pulling at the same time. The motion pattern is illustrated in Fig. 1.6(c).

Sidewinding

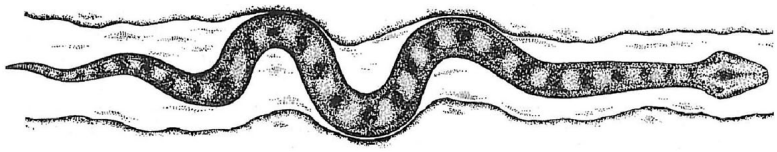
Sidewinding is a form of locomotion which is usually employed by snakes that live in areas of loose sand, e.g. desert snakes. The motion resembles concertina motion in that one part of the body acts as an anchor while another part is moved forward. Starting from a resting position, the head and neck are raised off the ground and thrown sideways while the rest of the body provides an anchor against the ground. Once the head and fore part of the body are again on the ground, they in turn act as an anchor while rest of the body repeats the same motion. The snake moves at about 45° with respect to its heading and leaves a trail of characteristic markings in the sand, as illustrated in Fig. 1.6(d).

The Control System of Snakes

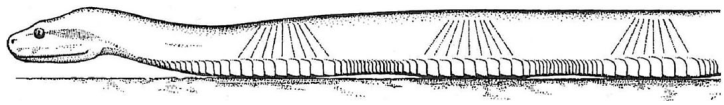
The employed locomotion method of snakes sometimes depends on the size of the snake and sometimes on the substrate over (or through) which it is moving. In fact, an interesting difference between snake locomotion and legged forms of locomotion is that the basic repeating motion that leads to propulsion of legged animals to a large extent depends on the progression



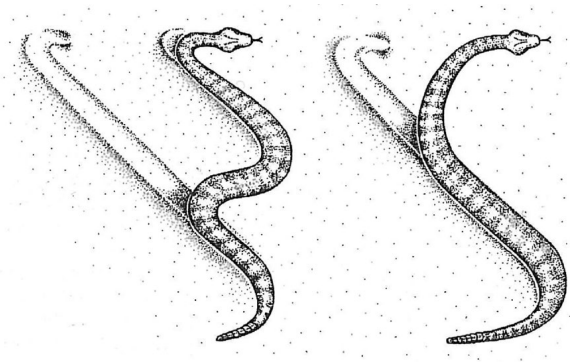
(a) Lateral undulation.



(b) Concertina locomotion.



(c) Rectilinear crawling.



(d) Sidewinding.

Figure 1.6: Different forms of biological snake locomotion. Image source: Mattison (2002). By permission of Chris Mattison.

speed of the animal. On the other hand, the basic repeating motion that leads to propulsion of snakes largely depends on the environment, and not the speed.

Considering the large number of muscles involved in the motion of a snake, and also the large number of contact points that are sensed by its nervous system, it is fair to say that the coordination of snake movements is both impressive and complex. Investigations of the electrical activity that accompanies the muscular contraction during movement show that the motor response is segmentary. Nerve impulses are propagated backwards along the snake body through the bone marrow. These impulses successively activate local muscle groups, which bend the snake body. Musculature is, in other words, successively, and not simultaneously, active, and only for a few elements at a time. The bending motion at a point along the snake body is also influenced by the sensory information transmitted by the skin. Simply speaking, the snake produces a relatively simple motor command which is modulated by local reflexes. This explains how every point in the body is able to follow the same trajectory.

1.3 Previous Work on Modelling, Development and Control of Snake Robots

This section provides an overview of previous literature on snake robot locomotion based on the review presented in Liljebäck *et al.* (2010*i*). The review is structured according to the title of this thesis by first considering research efforts related to modelling and analysis of snake robots, followed by research on physical development of these mechanisms, and finally considering previous control design efforts for snake locomotion. The scope of this thesis, which we present in Section 1.4, is motivated and justified based on this literature review.

1.3.1 Modelling and Analysis of Snake Robots

Previous literature on modelling and analysis of snake robot locomotion is summarized in Table 1.1. The table separates between works that consider snake locomotion from a planar (2D) perspective and works that also include three-dimensional aspects of the motion. A more detailed description of this literature is presented in the following.

Table 1.1: Previous work on modelling and analysis of snake robots.

Biomechanical studies of biological snakes	
2D perspective	Gray (1946), Moon and Gans (1998), Ma (1999).
3D perspective	Hirose (1993), Hu *et al.* (2009).
Flat surface locomotion with sideslip constraints	
2D perspective	Hirose (1993), Krishnaprasad and Tsakiris (1994), Kelly and Murray (1995), Ostrowski (1996), Ostrowski and Burdick (1998), Ishikawa (2009), Hatton and Choset (2009*b*), Prautsch and Mita (1999), Ute and Ono (2002), Matsuno and Mogi (2000), Matsuno and Sato (2005).
3D perspective	Ma *et al.* (2003), Tanaka and Matsuno (2008*b*), Date and Takita (2005).
Flat surface locomotion without sideslip constraints	
2D perspective	Ma (2001), Ma and Tadokoro (2006), Saito *et al.* (2002), Li and Shan (2008), Kane and Lecison (2000), Grabec (2002), Hicks (2003), Mehta *et al.* (2008), Chernousko (2005), Nilsson (2004), Hu *et al.* (2009).
3D perspective	Shapiro *et al.* (2007), Ma *et al.* (2004), Transeth *et al.* (2008*b*).
Robotic fish and eel-like mechanisms	
2D perspective	McIsaac and Ostrowski (2003*a*), Kanso *et al.* (2005).
3D perspective	Boyer *et al.* (2006), Zuo *et al.* (2008), Morgansen *et al.* (2001), Morgansen *et al.* (2002), Vela *et al.* (2002), Morgansen *et al.* (2007).
Locomotion in environments with obstacles	
2D perspective	Shan and Koren (1993), Bayraktaroglu and Blazevic (2005), Date and Takita (2007).
3D perspective	Chirikjian (1992), Chirikjian and Burdick (1995), Yamada and Hirose (2006*a*), Shan and Koren (1995), Tanev *et al.* (2005), Transeth *et al.* (2008*a*).

Biomechanical Studies of Biological Snakes

A complete treatment of previous studies of biological snakes is beyond the scope of this thesis. The biomechanical studies that we consider to be most relevant to this thesis are summarized in the following.

One of the earliest analytical studies of snake locomotion was given in Gray (1946), where mathematical descriptions of the forces acting on a snake are proposed and used to derive properties of snake locomotion. One of Gray's conclusions was that forward motion of a planar snake requires the existence of external forces acting in the normal direction of the snake body.

Hirose (1993) studied biological snakes and modelled the snake body as a continuous curve that could not move sideways (sideslip constraints). A well-known result by Hirose is the formulation of the *serpenoid curve*, which is a mathematical description of lateral undulation (the most common form of snake locomotion). Hirose also investigated adaptive functions of biological snakes (i.e. sinus-lifting, the α-adaptive principle, and the l-adaptive principle) and proposed mathematical descriptions of how external factors, such as ground friction and temperature, affect the shape of a snake during locomotion. Furthermore, Hirose investigated locomotion efficiency inside a maze, i.e. when the snake touches a wall on each side.

An alternative description of lateral undulation, named the *serpentine curve*, was proposed in Ma (1999), where a mathematical model of the muscle characteristics of snakes is employed to derive the resulting form of the body shape during lateral undulation. Ma showed that snake locomotion according to the serpentine curve has a higher locomotive efficiency than locomotion according to the serpenoid curve. The locomotive efficiency during slip-free motion was defined as the ratio between the tangential and normal directon friction forces on the snake body.

Other interesting studies of snake locomotion include the work in Moon and Gans (1998), which considers the mechanism by which muscular activity of a snake produces curvature and propulsion. In particular, the muscular activity is studied as a snake interacts with pegs in order to push itself forward. A more recent study given in Hu *et al.* (2009) experimentally investigates the frictional properties of snake skin. In particular, the study shows that the friction coefficient of a snake in the transversal direction of the body is larger than the friction coefficient in the tangential direction. This property is important during forward gliding motion. The study also shows that the weight distribution of a snake during lateral undulation is not uniform, but rather distributed so that the peaks of the body

wave curve are slightly lifted from the ground. This is often referred to as sinus-lifting.

Modelling of Flat Surface Locomotion with Sideslip Constraints

As noted in e.g. Gray (1946), each part of a biological snake conducting lateral undulation follows the path traced out by the head. This phenomenon is partially explained by the frictional anisotropy of snake skin studied in e.g. Hu *et al.* (2009), but is also caused by irregularities on the surface that provide grip and enable the snake to glide forward without slipping sideways. To mimic this motion, many models of snake robots have been developed under the explicit assumption that the body cannot move sideways (sideslip constraints). This assumption introduces nonholonomic constraints (Bloch *et al.*, 2003) in the equations of motion of the robot. In practice, such conditions are usually achieved by installing passive wheels along the body of the snake robot.

Several works attack the motion control problem of wheeled snake robots with tools from differential geometry. Early approaches of such form are presented in Kelly and Murray (1995); Krishnaprasad and Tsakiris (1994), which model the kinematics of wheeled snake robots and analyse the relationship between body shape changes and the resulting displacement of the robot. These works also assess the controllability of such mechanisms. Similar approaches are considered in Ostrowski and Burdick (1998); Ostrowski (1996), where also the dynamics of wheeled snake robots is considered, and where system symmetries are utilized to arrive at reduced forms of the model. Modelling and controllability analysis of the kinematics of a three-linked wheeled snake robot is also considered in Ishikawa (2009). Furthermore, Hatton and Choset (2009*b*) introduce the concept of a body velocity integral in order to easily approximate the net displacement of a snake robot during a gait. The method requires that the system coordinates are properly chosen.

A model of the 2D dynamics of a wheeled snake robot is developed in Prautsch and Mita (1999) from Lagrange's equations of motion, and in Ute and Ono (2002) from first principles. The works in Matsuno and Sato (2005); Matsuno and Mogi (2000) present models of the 2D kinematics and dynamics of snake robots, respectively, where some, but not all of the links are wheeled. The wheel-less links correspond to links that are lifted from the ground. Lifting some of the wheeled links is sometimes desirable from a control perspective to make the motion of the robot less constrained. A model of the 3D kinematics of a snake robot that describe the lifting of

the links more accurately is presented in Ma *et al.* (2003). Furthermore, Tanaka and Matsuno (2008*b*) present a model of the 3D dynamics of a snake robot consisting of a grounded base part and a lifted head part (for manipulation purposes), where some, but not all, of the links in the base part are wheeled.

Continuum models of snake robot dynamics, where the snake is treated as a continuous curve that cannot move sideways, are presented in Date and Takita (2005); Hirose (1993). The model in Hirose (1993) is planar, while the model in Date and Takita (2005) considers the 3D dynamics of the continuous snake robot.

Modelling of Flat Surface Locomotion without Sideslip Constraints

In addition to the many models of snake robots with sideslip constraints, there are also many models that do not enforce such constraints, but instead only assume that the links exhibit *anisotropic* ground friction properties similar to biological snakes. With anisotropic ground friction properties, the friction coefficients describing the friction force in the tangential and normal direction of a link, respectively, are different. Models based on such ground friction properties are generally more complex to analyse than models based on sideslip constraints since there is no longer a direct connection between the body shape changes and the resulting displacement of the robot.

Ma (2001) employs the Newton-Euler formulation to develop a 2D model of the dynamics of a snake robot with anisotropic ground friction properties. The ground friction model include both static and dynamic Coulomb ground friction forces. The model of the robot is formulated in two ways, where the first form gives the propulsion of the robot and the joint torques based on knowledge of the body shape changes, whereas the second form gives the propulsion and body shape changes of the robot based on knowledge of the joint torques. The model is extended in Ma and Tadokoro (2006) to also describe snake locomotion on a slope.

Another model of planar wheel-less snake robot dynamics is developed in Saito *et al.* (2002) from first principles. The model considers both viscous and Coulomb ground friction forces. Simulations with the model are carried out to derive properties of snake robot dynamics. The model from Saito *et al.* (2002) is employed in Li and Shan (2008) to study the controllability of the joints of a snake robot under the assumption that one joint is passive. However, the analysis does not consider the position of the robot.

Models of planar snake robot dynamics with anisotropic viscous ground friction are presented in Grabec (2002); Hicks (2003); Kane and Lecison

(2000). The work in Hicks (2003) exploits symmetries in the system (cyclic coordinates) to transform the model to a reduced form where the shape dynamics is decoupled from the displacement dynamics of the snake robot, and investigates general requirements for the propulsion of a three-linked snake robot. A friction model that includes both viscous and Coulomb friction forces is proposed and analysed in Mehta *et al.* (2008).

A model that considers *isotropic* Coulomb ground friction forces (both static and dynamic friction) is presented in Chernousko (2005). Isotropic ground friction is also assumed in Nilsson (2004), where a continuum approach along with energy arguments are employed to analyse planar snake locomotion under isotropic friction conditions. Shapiro *et al.* (2007) model the frictional contact forces between a snake robot and a compliant surface. The dynamics of planar snake locomotion is described in terms of a continuum model in Hu *et al.* (2009), where the snake is treated as a continuous curve influenced by Coulomb friction forces from the ground. The model is employed to study the effect of anisotropic ground friction properties on the propulsion of snakes.

The 3D dynamics of a snake robot during locomotion across flat surfaces is considered in Ma *et al.* (2004); Transeth *et al.* (2008*b*). The model in Ma *et al.* (2004) is developed from the Newton-Euler formulation and includes both static and dynamic Coulomb ground friction forces. The model is employed to study sinus-lifting during lateral undulation. Transeth *et al.* (2008*b*) model snake robot dynamics by use of the framework of nonsmooth dynamics. The model, which represents a hybrid system, describes the normal direction contact forces from the ground and the Coulomb ground friction forces by use of set-valued force laws.

Modelling of Robotic Fish and Eel-like Mechanisms

Research on robotic fish and eel-like mechanisms is relevant to research on snake robots since these mechanisms are very similar. A complete treatment of robotic underwater locomotion is beyond the scope of this review. However, a representative part of previous research related to modelling of such mechanisms is presented in the following.

A model of eel-like motion is developed in McIsaac and Ostrowski (2003*a*) based on tools from differential geometry that were also considered in some of the works concerning wheeled snake robots described above. However, the model does not place sideslip constraints on the robot. Instead, the eel-like mechanism is propelled by hydrodynamic forces modelled by a viscous friction model. The dynamics of eel-like motion is also con-

sidered in Kanso *et al.* (2005), where model reductions are proposed to allow the net motion of the robot to be described as a sum of geometric and dynamic phases over closed curves in the shape space, and in Boyer *et al.* (2006), where a continuum model is formulated based on beam theory, and in Zuo *et al.* (2008), where first principles are employed to model the dynamics of a swimming snake robot. The works in Morgansen *et al.* (2007, 2002, 2001); Vela *et al.* (2002) model the dynamics of a robotic fish influenced by lift and drag forces in an inviscous fluid. The controllability of the fish-like mechanism is also assessed in these works.

Modelling of Locomotion in Environments with Obstacles

In Chirikjian (1992); Chirikjian and Burdick (1995), the kinematics of snake robots is modelled in terms of a continuous backbone curve that captures the macroscopic geometry of the robot. Gaits for the backbone curve, which determine the shape of the snake robot, are specified with respect to environment constraints and the desired locomotion trajectory of the robot. The approach is original in that the problem of locomotion in cluttered environments is attacked at a purely kinematic level. The work by Chirikjian and Burdick is extended in Yamada and Hirose (2006*a*), where a continuum kinematics model is presented that explicitly handles the case of backbone curves that can be bent, but not twisted. This condition is in line with most physical snake robots, which are generally able to bend, but not twist their body. The kinematic constraints imposed on a snake robot due to external obstacles are modelled in Shan and Koren (1993, 1995). These works also analyse how obstacles around a snake robot affect its degrees of freedom.

The only known works that consider the dynamics of snake robots in environments with obstacles (i.e. where obstacle contact forces are considered) are presented in Bayraktaroglu and Blazevic (2005); Date and Takita (2007); Tanev *et al.* (2005); Transeth *et al.* (2008*a*). In Bayraktaroglu and Blazevic (2005), a dynamic simulation software called *WorkingModel* is used to simulate a planar snake robot interacting with circular obstacles. Contact forces are calculated from a spring-damper approximation. A similar approach is employed in Tanev *et al.* (2005), where the simulation software *Open Dynamics Engine* (ODE) is used to model a snake robot interacting with various forms of obstacles. Date and Takita (2007) use the multi-body dynamics simulation software *Autolev* to study the motion of a snake robot during contact with a single peg, where the contact with the peg is modelled as a spring-damper system. The works in Bayraktaroglu

and Blazevic (2005); Date and Takita (2007); Tanev *et al.* (2005) do not provide the equations underlying the dynamics of the snake robot due to the use of general-purpose simulation software. On the other hand, the model proposed in Transeth *et al.* (2008*a*) is, to our best knowledge, the only work which explicitly presents the equations of motion underlying the obstacle interaction dynamics of a snake robot. The model, which represents a hybrid system, is formulated within the framework of nonsmooth dynamics. A timestepping method is used to simulate the dynamics of the robot, which means that the system equations are discretized with a time step determined by a fixed error criterion, and trajectories of the system are approximated without tracking events (i.e. obstacle impacts). Transeth *et al.* (2008*a*) also introduce the term *obstacle-aided locomotion*, which involves using external obstacles as push points to aid the propulsion instead of avoiding them.

1.3.2 Development of Physical Snake Robots

Previous literature that considers development of physical snake robots is summarized in the following. The review is structured according to the focus of this thesis on snake robot locomotion based on measurements of environment contact forces, which we consider important for body shape adaptation in cluttered environments. In particular, to illustrate that previous research on environment sensing for snake robots is limited, we have chosen to separate the works that consider snake robtos *with* contact force sensors from the works that do *not* include such sensor capabilities in the robot design. The referred works are summarized in Table 1.2, which separates between snake robots with passive wheels, which are advantageous during motion across flat surfaces, snake robots without such passive wheels, and snake robots equipped with active propulsion.

Snake Robots without Contact Force Sensors

Hirose developed the world's first snake robot as early as 1972 (Hirose, 1993). This robot was equipped with passive wheels to realize the anisotropic ground friction property that enables forward locomotion on flat surfaces. Several other snake robots with passive wheels have been proposed over the years, such as the robots presented in Endo *et al.* (1999), Togawa *et al.* (2000), Ma *et al.* (2001), Wiriyacharoensunthorn and Laowattana (2002), Mori and Hirose (2002), Miller (2002), Ye *et al.* (2004*a*), Yamada *et al.* (2005), Ye *et al.* (2007), Crespi and Ijspeert (2008), Yu *et al.* (2008), Yu *et*

Table 1.2: Previous work on implementation of physical snake robots.

Snake Robots without Contact Force Sensors	
With passive wheels	Endo *et al.* (1999), Togawa *et al.* (2000), Ma *et al.* (2001), Wiriyacharoensunthorn and Laowattana (2002), Mori and Hirose (2002), Miller (2002), Ye *et al.* (2004*a*), Yamada *et al.* (2005), Ye *et al.* (2007), Crespi and Ijspeert (2008), Yu *et al.* (2008), Yu *et al.* (2009), Kamegawa *et al.* (2009).
Without passive wheels	Yim (1994), Yim *et al.* (2002), Worst and Linnemann (1996), Dowling (1997), Dowling (1999), Nilsson (1998), Ohno and Hirose (2001), Saito *et al.* (2002), Brunete *et al.* (2006), Chen *et al.* (2007), Wright *et al.* (2007), Kuwada *et al.* (2008), Yamada and Hirose (2009), Ohashi and Hirose (2010), Yamada and Hirose (2008).
With active propulsion	Kimura and Hirose (2002), Yamada and Hirose (2006*b*), Taal *et al.* (2009), Kamegawa *et al.* (2004), Masayuki *et al.* (2004), Granosik *et al.* (2006), Gao *et al.* (2008), McKenna *et al.* (2008), Hara *et al.* (2007).
Snake Robots with Contact Force Sensors	
With passive wheels	Hirose (1993), Chen *et al.* (2008).
Without passive wheels	Bayraktaroglu (2008), Gonzalez-Gomez *et al.* (2010), Liljebäck *et al.* (2006), Fjerdingen *et al.* (2008).
With active propulsion	Taal *et al.* (2009)

al. (2009), and Kamegawa *et al.* (2009). Some of the robots can only display planar motion, while other robots can move their links both horizontally and vertically. Some robots have shielded joint modules that enable motion in environments with e.g. mud and dust, and even motion under water, while other robots have modules with exposed electronic components which only allow them to move in clean lab environments. A common feature of these mechanisms, however, is that they are generally only able to move across relatively flat surfaces since passive wheels do not move very well in a cluttered environment. Such mechanisms are therefore suitable for motion control experiments on relatively flat surfaces, but not for practical applications of snake robots in more challenging environments.

Snake robots without passive wheels, i.e. robots that basically consist of straight links interconnected by motorized joints, are presented in Yim (1994), Yim *et al.* (2002), Worst and Linnemann (1996), Dowling (1997), Dowling (1999), Nilsson (1998), Ohno and Hirose (2001), Saito *et al.* (2002), Brunete *et al.* (2006), Chen *et al.* (2007), Wright *et al.* (2007), Kuwada *et al.* (2008), Yamada and Hirose (2009), and Ohashi and Hirose (2010). Despite its lack of wheels, the snake robot in Saito *et al.* (2002) maintains an anisotropic ground friction property since the underside of each link has edges, or grooves, that run parallel to the link. This robot can therefore move forward by lateral undulation through purely planar motion. Robots whose ground friction properties are isotropic, on the other hand, can move forward during lateral undulation by resorting to sinus-lifting, i.e. by slightly lifting the peaks of the body wave curve from the ground (see e.g. Ohno and Hirose, 2001; Yamada and Hirose, 2008). However, robots with isotropic friction are mostly used for studying gaits other than lateral undulation, such as gaits based on sidewinding, inchworm motion, or lateral rolling.

There are also works that consider active propulsion along the body of a snake robot, for example by equipping each link with motorized wheels (Kimura and Hirose, 2002; Taal *et al.*, 2009; Yamada and Hirose, 2006*b*), or by installing tracks along the body of the snake robot (Gao *et al.*, 2008; Granosik *et al.*, 2006; Kamegawa *et al.*, 2004; Masayuki *et al.*, 2004; McKenna *et al.*, 2008), or by employing a screw drive mechanism (Hara *et al.*, 2007).

Snake Robots with Contact Force Sensors

Previous research on environment sensing for snake robots is limited. The wheeled snake robot developed by Hirose already in 1972 (Hirose, 1993) was equipped with contact switches, which enabled the robot to demon-

strate lateral inhibition with respect to external obstacles. Snake robots with cylindrical modules covered by force sensors are proposed by our research group in Fjerdingen *et al.* (2008); Liljebäck *et al.* (2006). The force sensor systems are able to detect and, to some extent, assess the magnitude of external forces applied at certain areas of the joint modules. A snake robot with active wheels, where each wheel axis is equipped with a 3-axial force sensor, is presented in Taal *et al.* (2009). The force sensor measures the translational forces on the wheel axis based on optical range measurements. Bayraktaroglu (2008) presents a wheel-less snake robot with contact switches and presents experimental results where the robot is propelled forward by pushing against pegs that are detected by the contact switches. A snake robot with passive wheels and strain gauge sensors is proposed in Chen *et al.* (2008), where the strain gauge sensors are shown to successfully measure the constraint forces on the wheels. Ideas relating to environment sensing for snake robots are considered in Gonzalez-Gomez *et al.* (2010), where the preliminary design of a capacitive contact sensor is proposed that can be wrapped around each module of a snake robot.

1.3.3 Control of Snake Robots

In the following, we provide an overview of previous control design efforts for snake robots. The review is structured according to Tables 1.3 and 1.4, which summarize all papers referred to in this section. The two tables separate between works that present gait patterns without explicitly controlling the position or heading of the snake robot and works that present gait patterns along with position and/or heading controllers. The review focuses on controllers based on lateral undulation, which is the most common form of snake robot locomotion and which is also most relevant to the results presented in this thesis.

Remark 1.1 *Stability analysis of control laws for snake robots is challenging due to the complexity of existing models of these mechanisms. For this reason, applications of formal stability analysis tools in previous snake robot literature are very limited. Simulations and experimental investigations are instead the common approach in the literature for providing support of proposed control strategies.*

Control of Flat Surface Locomotion with Sideslip Constraints

A majority of previous control design efforts for snake robots has focused on locomotion where the links are subjected to nonholonomic constraints,

Table 1.3: Previous work on control of snake robot locomotion (1 of 2).

Flat surface locomotion with sideslip constraints	
Without position or heading control	Shan and Koren (1993), Kelly and Murray (1995), Ostrowski and Burdick (1998), Date and Takita (2005), Tanaka and Matsuno (2009), Ute and Ono (2002), Sato *et al.* (2010), Wang *et al.* (2010).
With position and/or heading control	Prautsch *et al.* (2000), Date *et al.* (2000), Date *et al.* (2001*b*), Date *et al.* (2001*a*), Yamakita *et al.* (2003), Matsuno and Mogi (2000), Ma *et al.* (2003), Matsuno and Suenaga (2003), Ye *et al.* (2004*b*), Matsuno and Sato (2005), Tanaka and Matsuno (2008*a*), Tanaka and Matsuno (2008*b*), Wiriyacharoensunthorn and Laowattana (2002), Watanabe *et al.* (2008), Ishikawa (2009), Ishikaway *et al.* (2010), Paap *et al.* (1999), Linnemann *et al.* (1999), Murugendran *et al.* (2009).
Flat surface locomotion without sideslip constraints	
Without position or heading control	Dowling (1997), Dowling (1999), Ma (2001), Ma *et al.* (2004), Saito *et al.* (2002), Chernousko (2003), Chernousko (2005), Transeth *et al.* (2007), Burdick *et al.* (1995), Gonzalez-Gomez *et al.* (2007), Yu *et al.* (2008), Chirikjian and Burdick (1995), Poi *et al.* (1998), Yim (1994), Yim *et al.* (2002), Ohno and Hirose (2001), Rincon and Sotelo (2003), Hatton and Choset (2010), Yamada and Hirose (2010), Mori and Hirose (2002), Chen *et al.* (2004), Ohashi and Hirose (2010).
With position and/or heading control	Hicks (2003), Hicks and Ito (2005).

Table 1.4: Previous work on control of snake robot locomotion (2 of 2).

Robotic fish and eel-like mechanisms	
Without position or heading control	Morgansen *et al.* (2001), Melli *et al.* (2006), Crespi and Ijspeert (2008).
With position and/or heading control	McIsaac and Ostrowski (2003*b*), McIsaac and Ostrowski (2003*a*), Morgansen *et al.* (2002), Vela *et al.* (2002), Morgansen *et al.* (2007).
Locomotion in environments with obstacles	
Without position or heading control	Hirose (1993), Andruska and Peterson (2008), Kuwada *et al.* (2008), Kulali *et al.* (2002), Greenfield *et al.* (2005), Kamegawa *et al.* (2009), Zarrouk *et al.* (2010), Nilsson (1997), Chen *et al.* (2007), Lipkin *et al.* (2007), Hatton and Choset (2009*a*).
With position and/or heading control	Bayraktaroglu and Blazevic (2005), Bayraktaroglu (2008), Date and Takita (2007), Sfakiotakis and Tsakiris (2007).

i.e. where each link is constrained from moving sideways. Shan and Koren (1993) consider a snake robot that uses solenoids for attachment to the environment and proposes gaits for forward and turning motion of this mechanism. Tools from differential geometry are employed in Kelly and Murray (1995); Ostrowski and Burdick (1998) to demonstrate that sinusoidal shape inputs to wheeled snake robots lead to propulsion.

A position and path following controller for a wheeled snake robot is proposed in Prautsch *et al.* (2000), where also Lyapunov analysis is employed to analyse the controller. The work also considers approaches for preventing the snake robot from attaining a straight shape, which is singular with respect to propulsion. The works in Date *et al.* (2001*a*, 2000, 2001*b*) propose path following controllers for wheeled snake robots aimed at minimizing the lateral constraint forces on the wheels during lateral undulation. The controllers are based on a measure of dynamic manipulability, which describes the ability of the robot to generate propulsive force. A similar approach is employed in Yamakita *et al.* (2003), which proposes a gait pattern aimed at minimizing the lateral constraint forces on the wheels, and in Date and Takita (2005), which formulates and solves an optimization problem in order to minimize the torque input. The optimization problem is solved using a 3D continuum model of the snake robot.

In Ma *et al.* (2003); Matsuno and Sato (2005); Matsuno and Mogi (2000); Matsuno and Suenaga (2003), position and path following controllers are proposed for the case where some, but not all, of the snake robot links are wheeled. The wheel-less links correspond to links that are lifted from the ground, which give the system more degrees of freedom that can be utilized to follow a trajectory while simultaneously maintaining a high manipulability. Similar approaches are considered in Tanaka and Matsuno (2008*a*, 2009, 2008*b*), where also strategies for sinus-lifting during lateral undulation are proposed.

Ute and Ono (2002) propose a gait based on a self-excitation principle where joint angle information determines the winding motion of a snake robot. Directional control during lateral undulation is considered in Wiriyacharoensunthorn and Laowattana (2002); Ye *et al.* (2004*b*). Watanabe *et al.* (2008) propose a position controller for a wheeled snake robot that takes ground friction forces into account. A similar approach is employed in Sato *et al.* (2010), where deviations of the joint angles from their setpoints are used to modify the oscillatory joint motion, thereby enabling the snake robot to automatically adapt its motion to variations in the ground friction conditions. The works in Ishikawa (2009); Ishikaway *et al.* (2010)

propose position and path following controllers for three-linked and four-linked wheeled snake robots based on Lie bracket calculations and controllability analysis results. Furthermore, a Poincaré map is employed to study the motion of the robot. The concept of passive creeping is considered in Wang *et al.* (2010), which involves adjusting the motion of a snake robot based on a measure of the dissipated energy, thereby achieving adaptation of the motion to different surface conditions. Local orbital stability of state trajectories during the motion is concluded based on recurrence plots.

A snake robot with *active* wheels is considered in Linnemann *et al.* (1999); Paap *et al.* (1999), where an optimization scheme is employed to make the robot follow the path that minimizes energy dissipation due to friction forces. Active wheels are also assumed in Murugendran *et al.* (2009), where a path following controller for such snake robots is proposed on a kinematic level.

Remark 1.2 *The works in e.g. Date* et al. *(2001*a*, 2000); Ma* et al. *(2003); Matsuno and Sato (2005); Matsuno and Mogi (2000); Prautsch* et al. *(2000), which were described above, all employ a common approach for motion control in that the nonholonomic constraints on the links are used to establish an explicit connection between body shape changes and propulsion, which allows the control input to be specified directly in terms of the desired propulsion of the robot. These approaches are, to our best knowledge, the only known approaches for motion control of wheeled snake robots which infer some formal and model-based conclusions on the propulsion of the robot.*

Control of Flat Surface Locomotion without Sideslip Constraints

Gait patterns of a wheel-less snake robot are specified in Dowling (1999, 1997) in terms of Fourier series coefficients, and certain learning techniques are employed for determining these parameters. In Ma (2001), computer simulations are employed to study properties of lateral undulation related to the optimality of the motion. Ma *et al.* (2004) propose a control strategy for sinus-lifting during lateral undulation by solving a quadratic optimization problem. Saito *et al.* (2002) consider snake robots influenced by anisotropic ground friction, and optimizes the gait parameters of lateral undulation based on simulations. The work also proposes a forward velocity controller for wheel-less snake robots. The works in Chernousko (2003, 2005) consider several elementary motions for planar snake robots and derive conditions for the feasibility of these motions, such as required actuator strength. In Hicks and Ito (2005); Hicks (2003), methods based

on numerical optimal control are considered for determining optimal gaits during positional control of snake robots influenced by anisotropic viscous ground friction. Gonzalez-Gomez *et al.* (2007) use *Open Dynamics Engine* (ODE) to propose and simulate various 3D gaits for translational and turning motion of snake robots, including a gait for rotation with very little displacement. Transeth *et al.* (2007) propose a controller for the joints of a planar snake robot influenced by anisotropic Coulomb ground friction, and prove that the resulting translational and rotational velocity of the robot is bounded.

The following works consider other gait patterns than lateral undulation, and the gaits are carried out in open-loop without explicitly controlling the position and orientation of the snake robot. Gaits for sidewinding motion, which is a sideways rolling type of motion, are proposed in Burdick *et al.* (1995); Gonzalez-Gomez *et al.* (2007); Hatton and Choset (2010); Yu *et al.* (2008). Inchworm locomotion gaits are proposed in Chirikjian and Burdick (1995); Gonzalez-Gomez *et al.* (2007); Ohno and Hirose (2001); Poi *et al.* (1998); Rincon and Sotelo (2003); Yamada and Hirose (2010); Yim (1994); Yim *et al.* (2002). Lateral rolling, which is achieved by continuously forming the snake body into a vertical U-shape that tips over, is considered in Chen *et al.* (2004); Dowling (1997); Gonzalez-Gomez *et al.* (2007); Mori and Hirose (2002); Ohno and Hirose (2001). Furthermore, gaits for loop forming motion are proposed in Ohashi and Hirose (2010); Yim (1994); Yim *et al.* (2002), where the head and tail of the snake robot are connected to turn the robot into a rolling wheel.

Remark 1.3 *To our best knowledge, previous literature has not presented any formal mathematical proofs regarding the propulsion of wheel-less snake robots.*

Control of Robotic Fish and Eel-like Mechanisms

A complete treatment of robotic underwater locomotion is beyond the scope of this review. However, we consider the above works to be representative of previous research related to control of such mechanisms.

Eel-like motion is considered in McIsaac and Ostrowski (2003*a*,*b*), where controllers for tracking straight and curved trajectories are proposed. The works in Morgansen *et al.* (2007, 2002, 2001); Vela *et al.* (2002) consider motion control of robotic fish. Lie bracket calculations based on the dynamics of the robotic fish are used to derive gaits for forward motion and various forms of turning motion. Algorithms for closed-loop heading and

depth control are also considered. Melli *et al.* (2006) propose open-loop gaits for a robotic fish based on curvature plots of the mechanical connection between the shape space motion and the overall displacement of the robot. A swimming snake robot is considered in Crespi and Ijspeert (2008), where a gradient-free optimization method is employed to adjust the gait parameters online, i.e. while the robot is moving, in order to maximize the forward veloctiy.

Control of Locomotion in Environments with Obstacles

Only a few works in previous literature consider control strategies for snake robots where the surface is no longer assumed to be flat (i.e. in environments with obstacles). To our best knowledge, the works in Bayraktaroglu (2008); Bayraktaroglu and Blazevic (2005); Hirose (1993) are the only works in previous literature which present control strategies for snake robots that employ explicit contact sensing in the feedback loop. Hirose (1993) proposes a strategy for lateral inhibition that modifies the shape of a snake robot based on contact force sensing along the snake body in order to avoid obstacles. Bayraktaroglu and Blazevic (2005) propose an inverse dynamics approach by formulating and numerically solving an optimization problem in order to, for a given set of obstacle contacts, calculate the contact forces required to propel the robot in a desired direction. A strategy for calculating the actual torque inputs to the joints from the desired contacts was, however, not presented. A kinematic approach is proposed in Bayraktaroglu (2008), where a curve fitting procedure is used to determine the shape of the robot with respect to the detected obstacles. Subsequently, this shape is propagated backwards along the snake body under the assumption that this will push the robot forward.

Sensing the environment of a snake robot must not necessarily involve contact force sensing since the environment can be indirectly sensed through the joint angle measurements and/or the actuator torques. This approach is considered in Date and Takita (2007), where the joint torques of a snake robot are specified solely in terms of the measured joint angles to achieve motion through a winding corridor, and in Andruska and Peterson (2008), which presents a control strategy that uses motor current measurements to adjust the shape of a snake robot moving through an elastically deformable channel, and in Kuwada *et al.* (2008), where the deviations of the joint angles from their setpoints are used to adapt the body shape of a snake robot moving inside pipe structures.

The remaining works presented in the following consider controllers

aimed at locomotion in environments that are not flat, but do not appear to involve sensing of the *interaction* between the snake robot and its environment. Kulali *et al.* (2002) employ a fuzzy logic controller to switch between various predefined gaits during motion in an obstacle environment. The goal of the motion controller is to avoid the obstacles. In Greenfield *et al.* (2005), an algorithm is presented that takes contact constraints on a snake robot into account in order to compute the joint torques that produce the desired motion. The algorithm is applied to achieve climbing motion with a snake robot. A gait for climbing motion is also proposed in Kamegawa *et al.* (2009). Sfakiotakis and Tsakiris (2007) use range sensor measurements to centre a crawling snake robot between the walls of a corridor. Zarrouk *et al.* (2010) analyse the efficiency of earthworm-like motion on compliant surfaces motivated by biomedical applications of worm robots. Moreover, various gaits aimed at motion in unstructured environments, including climbing gaits, are proposed in Chen *et al.* (2007); Hatton and Choset (2009*a*); Lipkin *et al.* (2007); Nilsson (1997).

1.4 Scope of the Thesis

The work underlying this thesis has been carried out with the following scope.

1.4.1 An Analytical Approach

The motivation behind this thesis is highly practical and applied. However, this thesis is primarily a theoretical study, although experimental investigations are also considered. In our opinion, there are many aspects related to control of snake robots that have not yet been addressed. Moreover, even though research on snake robots has been conducted for several decades, our understanding of snake locomotion so far is largely based on empirical studies of biological snakes and simulation-based synthesis of properties of snake robots. In this thesis, we therefore take an analytical approach in an attempt to increase our basic understanding of snake robot locomotion. We hope this approach will contribute to the mathematical foundation of the control theory of snake robots.

1.4.2 A Planar Perspective

This thesis considers planar snake robot locomotion in the horizontal plane. Of course, snake locomotion is inherently a three-dimensional phenomenon,

and a snake robot capable of strict planar motion will generally not be able to operate in unstructured and challenging environments. However, we believe the essential control principles of snake robot locomotion are contained in a planar perspective. In particular, since the fully three-dimensional motion of a snake robot consists of motion components in a horizontal and vertical plane, respectively, we conjecture that control laws that fulfil some control objective in a planar perspective can be extended to fulfil a similar control objective in a fully three-dimensional perspective. Moreover, we believe the simpler case of planar locomotion should be fully understood before the more challenging problem of three-dimensional locomotion is attacked.

1.4.3 Locomotion Without Sideslip Constraints

As indicated by the literature review in Section 1.3, a majority of previous research has focused on snake robots where the links are subjected to nonholonomic constraints, i.e. where each link is constrained from moving sideways. Such conditions are usually obtained by installing passive wheels in the tangential direction of the snake robot links. However, it seems unrealistic to enforce a nonholonomic constraint on each link during motion in unknown and cluttered environments, which represents the long-term goal of our research. In this thesis, we therefore consider snake robots where the links are allowed to slip sideways, often referred to as *wheel-less* snake locomotion.

1.4.4 Motion based on Lateral Undulation

This thesis focuses on motion analysis and control strategies where the snake robot moves according to various forms of the gait pattern lateral undulation (see Section 1.2.2). We have chosen this scope since lateral undulation is the fastest and most common form of snake locomotion. Moreover, we believe that this is the gait pattern which is most relevant and most efficient in a planar perspective.

1.5 Contributions of the Thesis

The title of this thesis, *Modelling, Development, and Control of Snake Robots*, is reflected in its structure and organization, as illustrated in Table 1.5. The thesis has two parts, where the first part targets snake robot locomotion on flat surfaces, while the second part targets snake robot locomotion

Table 1.5: The problem areas treated in each chapter.

		Modelling	Development	Control
Part I	Chapter 2	X		
	Chapter 3		X	
	Chapter 4			X
	Chapter 5			X
	Chapter 6	X		
	Chapter 7			X
	Chapter 8			X
Part II	Chapter 9			
	Chapter 10	X		
	Chapter 11		X	
	Chapter 12			X
	Chapter 13			X

in unstructured environments, i.e. environments containing external obstacles. In the following, we present the topic and corresponding contributions of each individual chapter of the thesis.

1.5.1 Contributions of Part I - Snake Robot Locomotion on Planar Surfaces

Chapter 2

Topic: We present a mathematical model of the kinematics and dynamics of a snake robot moving on a horizontal and flat surface. The links of the robot are influenced by ground friction forces, which propel the robot.

Contributions: The initial form of the model is not novel to this work since similar models of snake robot locomotion based on first principles are presented in Ma (2001); Saito *et al.* (2002). The two main contributions of this chapter are a change of coordinates, which enables us to partition the model of the snake robot into an actuated and an unactuated part, and a subsequent partial feedback linearization of the model. Due to the complexity of the initial form of the model, much of the model analysis presented in this thesis would not

have been feasible without the model transformation.

Chapter 3

Topic: We present the design of the snake robot *Wheeko*, which was developed for motion control experiments on flat surfaces.

Contributions: The internal structure of Wheeko is identical to the internal structure of the snake robot *Kulko*, which is described in Chapter 11. The contributions of this chapter are therefore included in the contributions of Chapter 11.

Chapter 4

Topic: We employ nonlinear system analysis tools for investigating fundamental properties of snake robot dynamics.

Contributions: The first contribution is a stabilizability analysis that proves that any asymptotically stabilizing control law for a planar snake robot to an equilibrium point must be *time-varying*, i.e. not of pure-state feedback type.

The second contribution is a controllability analysis of planar snake robots influenced by viscous ground friction forces that shows that a snake robot is *not* controllable when the ground friction is *isotropic*, but that a snake robot becomes *strongly accessible* when the ground friction is *anisotropic*. The analysis also shows that the snake robot does *not* satisfy sufficient conditions for *small-time local controllability*. To our best knowledge, no formal controllability analysis has previously been reported for the position and link angles of a wheelless snake robot influenced by ground friction. The results from the controllability analysis are not sufficient to conclude that a snake robot with anisotropic ground friction is controllable. However, the analysis proves that propulsion of a snake robot under viscous friction conditions requires the friction to be *anisotropic*, and also that the joint angles of a snake robot should be *out of phase* during snake locomotion. These claims have been assumed in the snake robot literature, but have never before been formally proven.

The third contribution is the development of a simple relationship between link velocities *normal* to the direction of motion and *propulsive forces* in the direction of motion that explains how snake robots influenced by anisotropic ground friction are able to locomote forward

on a planar surface. In our opinion, previously published research on snake robots has not presented an explicit mathematical description that *easily* explains how a snake robot achieves forward propulsion. As a fourth and final contribution, we present mathematical arguments that support the empirically derived mathematical description of lateral undulation proposed in Hirose (1993). Moreover, we identify an important property concerning the turning motion of a snake robot, and a property related to the relative displacements of the links during lateral undulation.

Chapter 5

Topic: We consider straight line path following control of snake robots.

Contributions: The contribution of this chapter is a control law that enables snake robots to track a straight path. We also employ a *Poincaré map* to show that all state variables of the snake robot, except for the position along the path, trace out an *exponentially stable* periodic orbit during path following with the proposed controller. To our best knowledge, this is the first time a *Poincaré map* is used to study the stability properties of snake robot locomotion.

Chapter 6

Topic: We consider an approach for simplifying the mathematical model of the snake robot.

Contributions: The main contribution of this chapter is a *simplified model* of planar snake robot locomotion, which is intended for control design and stability analysis purposes. Moreover, we provide support of the claim that the simplified model captures the essential part of the dynamics of planar snake robot locomotion by showing that the stabilizability and controllability properties of the simplified model are similar to the corresponding properties of the more complex model of the snake robot.

Chapter 7

Topic: We employ *averaging theory* to study the average effect of the joint motion that propels the snake robot during lateral undulation.

Contributions: The first contribution of this chapter is an averaged model of the velocity dynamics of a snake robot during lateral undulation. As a second contribution, we show that the average velocity of a snake robot during lateral undulation converges exponentially fast to a steady state velocity, and an analytical expression for calculating the steady state velocity is presented as a function of the gait pattern parameters. To our best knowledge, this is the first formal proof that a wheel-less snake robot with anisotropic ground friction properties achieves forward propulsion when it moves by lateral undulation.
The third contribution is a set of fundamental relationships between the gait pattern parameters of lateral undulation and the resulting forward velocity of a planar snake robot. In particular, the derived properties state that the average forward velocity of a snake robot 1) is proportional to the squared amplitude of the sinusoidal motion of each joint, 2) is proportional to the angular frequency of the sinusoidal motion of each joint, 3) is proportional to a particular function of the constant phase shift between the joints, and 4) is maximized by the phase shift between the joints that also maximizes the particular phase shift function. To our best knowledge, these fundamental properties of snake locomotion have never before been derived analytically. We also present experimental results based on the snake robot Wheeko that support the derived properties.

Chapter 8

Topic: We return to the problem of straight line path following control of snake robots, but this time based on the simplified model.

Contributions: The first and main contribution of this chapter is a straight line path following controller, which, using cascaded systems theory, is proved to $\mathcal{K}$-exponentially stabilize a snake robot to any desired straight path. The proof relies on the assumption that the forward velocity of the robot is contained in some nonzero and positive interval. To our best knowledge, this is the first time the stability properties of a path following controller for a snake robot without nonholonomic constraints are formally proved. We also present experimental results where the proposed controller successfully steers the snake robot Wheeko towards and along the desired straight path.
The second contribution is a description of how the straight line path following controller can be extended to path following of general

curved paths by employing an approach previously proposed in the marine control literature for path following control of marine vessels. The third contribution is a waypoint guidance strategy for steering a snake robot along a path defined by waypoints interconnected by straight lines, and a proof that the guidance strategy is guaranteed to steer the position of the robot into the acceptance region of each waypoint. Waypoint guidance has, to our best knowledge, not previously been considered for motion control of snake robots.

1.5.2 Contributions of Part II - Snake Robot Locomotion in Unstructured Environments

Chapter 9

Topic: This chapter gives an introduction to the second part of the thesis.

Contributions: We regard the thoughts and ideas presented in this chapter as a contribution.

Chapter 10

Topic: We extend the mathematical model of the snake robot to include contact forces from external obstacles in the environment around the robot.

Contributions: The contribution of this chapter is a *hybrid model* of a planar snake robot interacting with obstacles in its environment. The obstacle interaction is modelled by introducing a *unilateral velocity constraint* on each contacted link of the snake robot, which is a novel approach. In particular, the conventional approach for modelling the obstacle interaction would be to assume that the obstacle constraint force points in the normal direction of the *obstacle* (see Brogliato, 1999). With the approach described in this chapter, the shape of the obstacles does not have to be considered explicitly as we instead calculate constraint forces with respect to the normal direction of the *contacted links*, which simplifies the equations of motion. The investigations of the existence and uniqueness properties of the hybrid model based on the theory of *linear complementarity problems* is also considered to be a contribution of this chapter.

Chapter 11

Topic: We present the design of the snake robot *Kulko*, which was developed for the purpose of experiments related to obstacle-aided locomotion in unstructured environments.

Contributions: Previous snake robot design efforts have given very limited attention to the exterior gliding surface of such robots, and to methods for enabling snake robots to sense their environment. The contribution of this chapter is therefore the design of a ball-shaped joint mechanism for a snake robot that 1) allows the joint modules to be covered by shells, thereby giving the robot a smooth outer surface independently of how the joints are flexed, and that 2) allows contact force sensors to be installed underneath the shells, thereby enabling the robot to sense its environment. Experimental results are presented that validate the function of the contact force measurement system. To our best knowledge, this is the first reported snake robot that can measure the magnitude of external forces applied along its body. This chapter also proposes an alternative approach for sensing environment contact forces, which has the advantage that force measurements are only required at the locations of the joints, and that the sensor system can be well protected inside the snake robot.

Chapter 12

Topic: We consider motion control of snake robots on surfaces with irregularities in the form of external obstacles.

Contributions: The first contribution of this chapter is a general control principle for snake robots which suggests that obstacle-aided locomotion is achieved by producing body shape changes where the links in contact with obstacles are rotated to increase the propulsive forces on the robot.

As a second contribution, we use the control principle to propose a *hybrid controller* for obstacle-aided locomotion aimed at resolving situations where the snake robot is jammed between obstacles. The concept of detecting and resolving snake robot jams has, to our best knowledge, not been treated in previous literature, but is a genuine challenge during snake robot locomotion in cluttered environments. Moreover, this is the first control strategy for a snake robot involving feedback and explicit use of measured contact forces to achieve

propulsion. The work in Hirose (1993) also considers snake locomotion based on measured contact forces. However, the contact forces in Hirose (1993) are employed to *avoid* obstacles, whereas the contact forces in this chapter are employed to push the snake robot forward. We present experimental results based on the snake robot Kulko, where the hybrid controller is shown to maintain the propulsion of the snake robot in different obstacle environments. Also to our best knowledge, this is the first reported experiment where a snake robot is propelled forward based on measurements of the contact force amplitudes along the robot body. The works in Bayraktaroglu (2008); Hirose (1993) also report experiments where a snake robot is propelled by obstacle contact forces. However, the control strategies in these experiments do not consider the amplitude of the contact forces.

Chapter 13

Topic: We consider straight line path following control of snake robots in environments containing obstacles.

Contributions: The first contribution of this chapter is a general framework for motion control of snake robots, where the motion is specified in terms of a *body wave* component, an *environment adaptation* component, and a *heading control* component.

The second contribution is a control law (based on the controller framework) for straight line path following control of snake robots in environments with obstacles. A significant contribution of this controller is the idea of a continuous jam resolution action that is performed in parallel with the cyclic wave motion of the robot to continuously adapt the body shape to the environment and prevent the motion from being jammed. A formal analysis of the performance of the path following controller remains a topic of future work.

As a third contribution of this chapter, we present experimental results where the snake robot Kulko is successfully propelled through three different obstacle environments with the proposed controller.

Part I

Snake Robot Locomotion on Planar Surfaces

Chapter 2

A Complex Model of Snake Robot Locomotion on Planar Surfaces

The underlying theme of this thesis is analytical approaches aimed at increasing our understanding of snake robot locomotion. The mathematical model of the snake robot is the basis for these analytical studies, which means that the analysis relies heavily on the form and complexity of the model. In this chapter, we employ first principles to develop a mathematical model of the kinematics and dynamics of a snake robot with N links moving on a horizontal and flat surface. The links of the robot are influenced by ground friction forces, which propel the motion. Due to the many degrees of freedom of the robot and the dynamical couplings between the links, the resulting model will turn out to be quite complex. We will eliminate some of this complexity by partially linearizing the model. This is achieved by introducing a change of coordinates which enables us to partition the model into an actuated part (the joint angles of the snake robot) and an unactuated part (the position and orientation of the snake robot). Through an input transformation, we are then able to linearize the actuated part of the model. However, even the partially linearized model contains complex terms which make model-based controller design and analysis challenging. Throughout this thesis, we will therefore refer to the model developed in this chapter as the *complex* model of the snake robot.

In Chapter 4, the complex model will be analysed in order to deduce several fundamental properties of snake robot dynamics. Some of these properties will be instrumental in the development of a *simplified* model

of the snake robot in Chapter 6, where we propose a model that captures only the 'essential' part of the dynamics of the complex model. In Part II of this thesis, which considers snake robot locomotion in unstructured environments, the complex model will be extended to include contact forces from external obstacles in the environment around the snake robot.

Contributions of this Chapter: The non-linearized model of the snake robot is not novel to this work since similar models of snake robot locomotion based on first principles are presented in Ma (2001); Saito *et al.* (2002). The notation and the ground friction models considered in this chapter are, however, different from the works in Ma (2001); Saito *et al.* (2002). Moreover, the expression for the linear velocity of individual links given in (2.13) is novel to this work. The two main contributions of this chapter are the change of coordinates which enables us to partition the model into an actuated and an unactuated part, and the subsequent partial feedback linearization of the model. Due to the complexity of the non-linearized model of the snake robot, much of the model analysis presented in Chapter 4 would not have been feasible without the model transformation.

Organization of this Chapter: Section 2.1 introduces some basic notation that will be used throughout the thesis. The parameters that characterize the snake robot are presented in Section 2.2. The kinematics of the snake robot is described in Section 2.3, while two different ground friction models are presented in Section 2.4. The model of the snake robot dynamics is presented in Section 2.5, and is partitioned into an actuated and an unactuated part in Section 2.6, and is transformed to a simpler form through partial feedback linearization in Section 2.7. Finally, the chapter is summarized in Section 2.8.

Publications: The material in this chapter is based on the journal papers Liljebäck *et al.* (2011*b*) and Liljebäck *et al.* (2010*h*), and on the conference papers Liljebäck *et al.* (2009*a*) and Liljebäck *et al.* (2009*b*).

2.1 Basic Notation

The following notation is used throughout this thesis:

- The operator $\text{sgn}(\cdot)$ produces a vector containing the sign of the individual elements of its argument.

- The operator diag $(\cdot)$ produces a diagonal matrix with the elements of its argument along its diagonal.
- The sinus and cosine operators, $\sin(\cdot)$ and $\cos(\cdot)$, are vector operators when their argument is a vector and scalar operators when their argument is a scalar value.
- We will use subscript i to denote element i of a vector (see Table 2.1 below). When parameters of the links (joints) of the snake robot are assembled into a vector, we associate element i of this vector with link i (joint i).
- Symbols representing a vector or a matrix are indicated with a bold font.
- The matrix $\boldsymbol{I}_k$ represents the $k \times k$ identity matrix and $\boldsymbol{0}_{i\times j}$ represents the $i \times j$ matrix of zeros.
- A vector related to link i of the snake robot is either expressed in the global coordinate system or in the local coordinate system of the link (see Fig. 2.1). This is indicated by superscript *global* or *link*, i, respectively. If not otherwise specified, a vector with no superscript is expressed in the global coordinate system.

2.2 The Parameters of the Snake Robot

The snake robot consists of N rigid links of length $2l$ interconnected by $N-1$ motorized joints. The width of each link is not considered in the model. All N links have the same mass m and moment of inertia $J = \frac{1}{3}ml^2$. The total mass of the snake robot is therefore Nm. The mass of each link is uniformly distributed so that the link CM (centre of mass) is located at its centre point (at length l from the joint at each side). In the following subsections, the kinematics and dynamics of the snake robot will be modelled in terms of the mathematical symbols described in Table 2.1 and illustrated in Fig. 2.1 and Fig. 2.2. We will make use of the following vectors and matrices:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & & \\ & \ddots & \ddots & \\ & & 1 & 1 \end{bmatrix} \in \mathbb{R}^{(N-1)\times N}, \boldsymbol{D} = \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \end{bmatrix} \in \mathbb{R}^{(N-1)\times N},$$

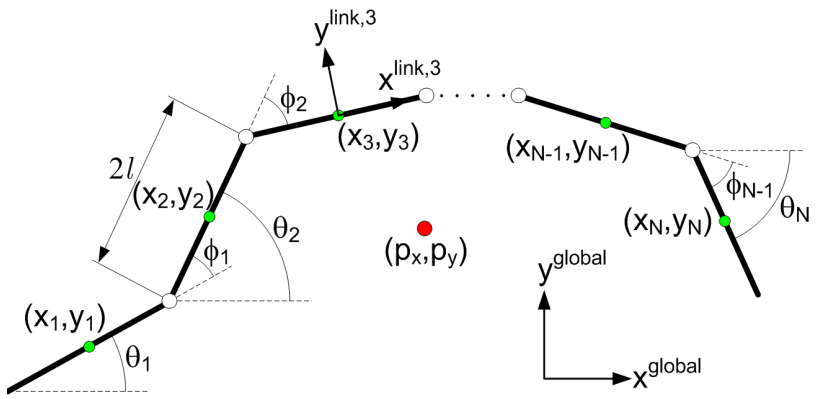


Figure 2.1: The kinematic parameters of the snake robot.

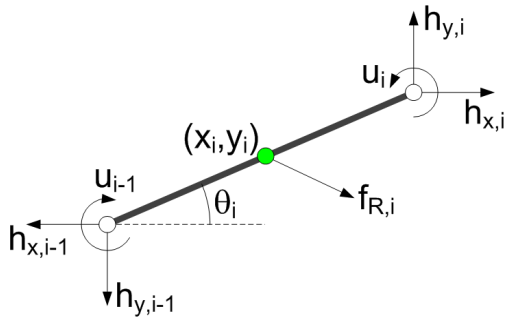


Figure 2.2: Forces and torques acting on each link of the snake robot.

$$\boldsymbol{e} = [1, \ldots, 1]^T \in \mathbb{R}^N, \quad \boldsymbol{E} = \begin{bmatrix} \boldsymbol{e} & \boldsymbol{0}_{N\times 1} \\ \boldsymbol{0}_{N\times 1} & \boldsymbol{e} \end{bmatrix} \in \mathbb{R}^{2N\times 2},$$
$$\sin\boldsymbol{\theta} = [\sin\theta_1, \ldots, \sin\theta_N]^T \in \mathbb{R}^N, \quad \boldsymbol{S}_\theta = \operatorname{diag}(\sin\boldsymbol{\theta}) \in \mathbb{R}^{N\times N},$$
$$\cos\boldsymbol{\theta} = [\cos\theta_1, \ldots, \cos\theta_N]^T \in \mathbb{R}^N, \quad \boldsymbol{C}_\theta = \operatorname{diag}(\cos\boldsymbol{\theta}) \in \mathbb{R}^{N\times N},$$
$$\operatorname{sgn}\boldsymbol{\theta} = [\operatorname{sgn}\theta_1, \ldots, \operatorname{sgn}\theta_N]^T \in \mathbb{R}^N, \quad \dot{\boldsymbol{\theta}}^2 = \left[\dot{\theta}_1^2, \ldots, \dot{\theta}_N^2\right]^T \in \mathbb{R}^N.$$

2.3 The Kinematics of the Snake Robot

The snake robot moves on a horizontal and flat surface, and has $N + 2$ degrees of freedom (N link angles and the planar position of the robot).

Table 2.1: Parameters that characterize the snake robot.

Symbol	Description	Associated vector
N	The number of links.	
l	Half the length of a link.	
m	Mass of each link.	
J	Moment of inertia of each link.	
θ_i	Angle between link i and the global x axis.	$\boldsymbol{\theta} \in \mathbb{R}^N$
ϕ_i	Angle of joint i.	$\boldsymbol{\phi} \in \mathbb{R}^{N-1}$
(x_i, y_i)	Global coordinates of the CM of link i.	$\boldsymbol{X}$, $\boldsymbol{Y} \in \mathbb{R}^N$
(p_x, p_y)	Global coordinates of the CM of the robot.	$\boldsymbol{p} \in \mathbb{R}^2$
u_i	Actuator torque exerted on link i from link $i+1$.	$\boldsymbol{u} \in \mathbb{R}^{N-1}$
u_{i-1}	Actuator torque exerted on link i from link $i-1$.	$\boldsymbol{u} \in \mathbb{R}^{N-1}$
$f_{R,x,i}$	Friction force on link i in the x direction.	$\boldsymbol{f}_{R,x} \in \mathbb{R}^N$
$f_{R,y,i}$	Friction force on link i in the y direction.	$\boldsymbol{f}_{R,y} \in \mathbb{R}^N$
$h_{x,i}$	Joint constraint force in x direction on link i from link $i+1$.	$\boldsymbol{h}_x \in \mathbb{R}^{N-1}$
$h_{y,i}$	Joint constraint force in y direction on link i from link $i+1$.	$\boldsymbol{h}_y \in \mathbb{R}^{N-1}$
$h_{x,i-1}$	Joint constraint force in x direction on link i from link $i-1$.	$\boldsymbol{h}_x \in \mathbb{R}^{N-1}$
$h_{y,i-1}$	Joint constraint force in y direction on link i from link $i-1$.	$\boldsymbol{h}_y \in \mathbb{R}^{N-1}$

The following definitions are illustrated in Fig. 2.1.

Definition 2.1 ***Link angle.***
The link angle *of link* $i \in \{1, \ldots, N\}$ *of the snake robot is denoted by* $\theta_i \in \mathbb{R}$ *and is defined as the angle that the link forms with the global* x *axis with counterclockwise positive direction.*

Definition 2.2 ***Joint angle.***
The joint angle *of joint* $i \in \{1, \ldots, N-1\}$ *of the snake robot is denoted* $\phi_i \in \mathbb{R}$ *and is defined as*

$$\phi_i = \theta_i - \theta_{i+1}. \tag{2.1}$$

Note the distinction between *link angles* and *joint angles*. A link angle is the orientation of a link with respect to the global x axis, while a joint angle is the difference between the link angles of two neighbouring links. We will quite frequently assemble the link angles and the joint angles in the vectors $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_N]^T \in \mathbb{R}^N$ and $\boldsymbol{\phi} = \begin{bmatrix}\phi_1, \ldots, \phi_{N-1}\end{bmatrix}^T \in \mathbb{R}^{N-1}$, respectively.

The snake robot has no explicitly defined orientation since there is an independent link angle associated with each link. We can still obtain a measure of the heading of the robot as follows (this approach is also considered in e.g. Hatton and Choset (2009*b*); Hu *et al.* (2009)):

Definition 2.3 ***Heading.***
The heading *(or* orientation*) of the snake robot is denoted by* $\overline{\theta} \in \mathbb{R}$ *and is defined as the average of the link angles, i.e. as*

$$\overline{\theta} = \frac{1}{N} \sum_{i=1}^{N} \theta_i. \tag{2.2}$$

The local coordinate system of each link is fixed in the CM of the link with x (tangential) and y (normal) axes oriented such that they are aligned with the global x and y axis, respectively, when the link angle is zero. The rotation matrix from the global frame to the frame of link i is given by

$$\boldsymbol{R}_{\text{link},i}^{\text{global}} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix}. \tag{2.3}$$

The global frame position $\boldsymbol{p} \in \mathbb{R}^2$ of the CM (centre of mass) of the robot is given by

$$\boldsymbol{p} = \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} \frac{1}{Nm} \sum_{i=1}^{N} m x_i \\ \frac{1}{Nm} \sum_{i=1}^{N} m y_i \end{bmatrix} = \frac{1}{N} \begin{bmatrix} \boldsymbol{e}^T \boldsymbol{X} \\ \boldsymbol{e}^T \boldsymbol{Y} \end{bmatrix}, \tag{2.4}$$

where $\boldsymbol{e}$ was defined in Section 2.2, (x_i, y_i) are the global frame coordinates of the CM of link i, $\boldsymbol{X} = [x_1, \ldots, x_N]^T \in \mathbb{R}^N$, and $\boldsymbol{Y} = [y_1, \ldots, y_N]^T \in \mathbb{R}^N$. We define the velocity of the snake robot along its *forward* direction as follows:

Definition 2.4 ***Forward velocity.***
The forward velocity *of the snake robot is denoted by* $\overline{v}_t \in \mathbb{R}$ *and is defined as the component of the CM velocity* $\dot{\boldsymbol{p}}$ *along the current heading* $\overline{\theta}$*, i.e. as*

$$\overline{v}_t = \dot{p}_x \cos\overline{\theta} + \dot{p}_y \sin\overline{\theta}. \tag{2.5}$$

Remark 2.1 *Subscript* t *in the forward velocity* $\overline{v}_t$ *denotes* tangential*. The simplified model of the snake robot presented in Chapter 6 makes a clear distinction between the forward velocity* v_t *and the sideways velocity* v_n *of the robot. We have chosen to denote the forward velocity in the complex model by* $\overline{v}_t$ *to maintain a similar notation as in the simplified model.*

The connection between link i and link $i+1$ at joint $i \in \{1, \ldots, N-1\}$ must comply with the two *holonomic* constraints

$$x_{i+1} - x_i = l\cos\theta_i + l\cos\theta_{i+1}, \tag{2.6a}$$
$$y_{i+1} - y_i = l\sin\theta_i + l\sin\theta_{i+1}. \tag{2.6b}$$

Using the notation from Section 2.2, the joint constraints for all the links of the robot can be written in matrix form as

$$\boldsymbol{D}\boldsymbol{X} + l\boldsymbol{A}\cos\boldsymbol{\theta} = \boldsymbol{0}, \tag{2.7a}$$
$$\boldsymbol{D}\boldsymbol{Y} + l\boldsymbol{A}\sin\boldsymbol{\theta} = \boldsymbol{0}. \tag{2.7b}$$

We can now express the position of the individual links as a function of the CM position and the link angles of the robot by combining (2.4) and (2.7) into

$$\boldsymbol{T}\boldsymbol{X} = \begin{bmatrix} -l\boldsymbol{A}\cos\boldsymbol{\theta} \\ p_x \end{bmatrix}, \quad \boldsymbol{T}\boldsymbol{Y} = \begin{bmatrix} -l\boldsymbol{A}\sin\boldsymbol{\theta} \\ p_y \end{bmatrix}, \tag{2.8}$$

where

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{D} \\ \frac{1}{N}\boldsymbol{e}^T \end{bmatrix} \in \mathbb{R}^{N\times N}. \tag{2.9}$$

It can be shown that

$$\boldsymbol{T}^{-1} = \begin{bmatrix} \boldsymbol{D}^T \left(\boldsymbol{D}\boldsymbol{D}^T\right)^{-1} & \boldsymbol{e} \end{bmatrix}, \tag{2.10}$$

which enables us to solve (2.8) for $\boldsymbol{X}$ and $\boldsymbol{Y}$ according to

$$\boldsymbol{X} = \boldsymbol{T}^{-1}\begin{bmatrix} -l\boldsymbol{A}\cos\boldsymbol{\theta} \\ p_x \end{bmatrix} = -l\boldsymbol{K}^T\cos\boldsymbol{\theta} + \boldsymbol{e}p_x, \tag{2.11a}$$

$$\boldsymbol{Y} = \boldsymbol{T}^{-1}\begin{bmatrix} -l\boldsymbol{A}\sin\boldsymbol{\theta} \\ p_y \end{bmatrix} = -l\boldsymbol{K}^T\sin\boldsymbol{\theta} + \boldsymbol{e}p_y, \tag{2.11b}$$

where $\boldsymbol{K} = \boldsymbol{A}^T\left(\boldsymbol{D}\boldsymbol{D}^T\right)^{-1}\boldsymbol{D} \in \mathbb{R}^{N\times N}$ and where $\boldsymbol{D}\boldsymbol{D}^T$ is nonsingular and thereby invertible. The linear velocities of the links are found by differentiating (2.11) with respect to time, which gives

$$\dot{\boldsymbol{X}} = l\boldsymbol{K}^T\boldsymbol{S}_\theta\dot{\boldsymbol{\theta}} + \boldsymbol{e}\dot{p}_x, \tag{2.12a}$$

$$\dot{\boldsymbol{Y}} = -l\boldsymbol{K}^T\boldsymbol{C}_\theta\dot{\boldsymbol{\theta}} + \boldsymbol{e}\dot{p}_y. \tag{2.12b}$$

By manually investigating the structure of each row in (2.12), it can be verified that the linear velocity of the CM of link i in the global x and y directions is given by

$$\dot{x}_i = \dot{p}_x - \sigma_i\boldsymbol{S}_\theta\dot{\boldsymbol{\theta}}, \tag{2.13a}$$

$$\dot{y}_i = \dot{p}_y + \sigma_i\boldsymbol{C}_\theta\dot{\boldsymbol{\theta}}, \tag{2.13b}$$

where

$$\sigma_i = \left[a_1, a_2, \ldots, a_{i-1}, \frac{a_i + b_i}{2}, b_{i+1}, b_{i+2}, \ldots, b_N\right] \in \mathbb{R}^N, \tag{2.14a}$$

$$a_i = \frac{l\,(2i-1)}{N}, b_i = \frac{l\,(2i-1-2N)}{N}. \tag{2.14b}$$

2.4 Ground Friction Models

2.4.1 The Friction Models and their Role in this Thesis

As will become apparent in Chapter 4, a planar snake robot achieves forward propulsion on a flat surface by continuously changing its body shape

to induce ground friction forces that propel the robot forward. The ground friction model is therefore an important part of the dynamics of the snake robot.

During planar locomotion, it is of great advantage to the propulsion of the snake robot that the ground friction forces on the links are *anisotropic*, which means that the friction coefficients describing the friction force in the tangential (along link x axis) and normal (along link y axis) direction of a link, respectively, are different. As described in the literature review of Section 1.3, anisotropic ground friction is assumed in the majority of published research on snake robots. This friction property is also exhibited by biological snakes, as was explained in the description of biological snakes in Section 1.2. We therefore include anisotropic friction conditions in the friction model of the snake robot. The importance of this friction property will be investigated in more detail in Chapter 4.

We consider two different ground friction models in this thesis, i.e. a *Coulomb* friction model and a *viscous* friction model. The Coulomb friction model, which assumes the ground friction on a link to be proportional to the weight of the link, is more accurate (from a physical point of view) than the viscous friction model, which assumes the ground friction on a link to be proportional to the velocity of the link. However, during planar locomotion, we conjecture that the anisotropic friction property of the links, which is independent of the choice of Coulomb or viscous friction, is the decisive factor of the motion. In other words, we conjecture that the motion of the snake robot is *qualitatively* (although not *quantitatively*) similar with anisotropic viscous friction as with anisotropic Coulomb friction. The viscous friction model is, however, less complex than the Coulomb friction model, which makes the viscous model more suitable for control design and analysis purposes. In this thesis, we will therefore mostly assume that the ground friction is viscous.

In the following, we first present the Coulomb friction model, and subsequently the viscous friction model. In both models, the ground friction force on link i is assumed to act on the CM of the link only, and is denoted by

$$\boldsymbol{f}_{R,i} = \boldsymbol{f}_{R,i}^{\text{global}} = \begin{bmatrix} f_{R,x,i} \\ f_{R,y,i} \end{bmatrix} \in \mathbb{R}^2. \tag{2.15}$$

The friction forces on all links are written in matrix form as

$$\boldsymbol{f}_R = \begin{bmatrix} \boldsymbol{f}_{R,x} \\ \boldsymbol{f}_{R,y} \end{bmatrix} \in \mathbb{R}^{2N}, \tag{2.16}$$

where $\boldsymbol{f}_{R,x} = [f_{R,x,1}, \ldots, f_{R,x,N}]^T \in \mathbb{R}^N$ and $\boldsymbol{f}_{R,y} = [f_{R,y,1}, \ldots, f_{R,y,N}]^T \in \mathbb{R}^N$ contain the friction forces on the links in the global x and y direction, respectively.

2.4.2 Coulomb Friction Model

The coefficients describing the Coulomb friction force in the tangential (along link x axis) and normal (along link y axis) direction of a link, respectively, are denoted by μ_t and μ_n, respectively. We define the Coulomb friction force on link i in the local link frame, $\boldsymbol{f}_{R,i}^{\text{link},i} \in \mathbb{R}^2$, as

$$\boldsymbol{f}_{R,i}^{\text{link},i} = -mg \begin{bmatrix} \mu_t & 0 \\ 0 & \mu_n \end{bmatrix} \operatorname{sgn}\left(\boldsymbol{v}_i^{\text{link},i}\right), \tag{2.17}$$

where $\boldsymbol{v}_i^{\text{link},i} \in \mathbb{R}^2$ is the link velocity expressed in the local link frame, and g is the gravitational acceleration constant. Using (2.3), we can express the global frame Coulomb friction force on link i in the form of (2.15) as

$$\begin{aligned} \boldsymbol{f}_{R,i} &= \boldsymbol{f}_{R,i}^{\text{global}} = \boldsymbol{R}_{\text{link},i}^{\text{global}} \boldsymbol{f}_{R,i}^{\text{link},i} \\ &= -mg\boldsymbol{R}_{\text{link},i}^{\text{global}} \begin{bmatrix} \mu_t & 0 \\ 0 & \mu_n \end{bmatrix} \operatorname{sgn}\left(\boldsymbol{v}_i^{\text{link},i}\right) \\ &= -mg\boldsymbol{R}_{\text{link},i}^{\text{global}} \begin{bmatrix} \mu_t & 0 \\ 0 & \mu_n \end{bmatrix} \operatorname{sgn}\left(\left(\boldsymbol{R}_{\text{link},i}^{\text{global}}\right)^T \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix}\right). \end{aligned} \tag{2.18}$$

By performing the matrix multiplication in (2.18) and assembling the forces on all links in matrix form, the global frame Coulomb friction forces on the links can be written in the form of (2.16) as

$$\boldsymbol{f}_R = \begin{bmatrix} \boldsymbol{f}_{R,x} \\ \boldsymbol{f}_{R,y} \end{bmatrix} = -mg \begin{bmatrix} \mu_t \boldsymbol{C}_\theta & -\mu_n \boldsymbol{S}_\theta \\ \mu_t \boldsymbol{S}_\theta & \mu_n \boldsymbol{C}_\theta \end{bmatrix} \operatorname{sgn}\left(\begin{bmatrix} \boldsymbol{C}_\theta & \boldsymbol{S}_\theta \\ -\boldsymbol{S}_\theta & \boldsymbol{C}_\theta \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{X}} \\ \dot{\boldsymbol{Y}} \end{bmatrix}\right) \in \mathbb{R}^{2N}. \tag{2.19}$$

2.4.3 Viscous Friction Model

Similar to the Coulomb friction model, we assume that the viscous ground friction forces act on the CM of the links only. We present the viscous friction model for the different cases of isotropic versus anisotropic viscous friction since these two cases are analysed separately in Chapter 4.

Isotropic Viscous Friction

The isotropic viscous friction force on link i in the global x and y direction is proportional to the global frame velocity of the link given by (2.13), and is written in the form of (2.15) as

$$\boldsymbol{f}_{R,i} = \boldsymbol{f}_{R,i}^{\text{global}} = -c \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = -c \begin{bmatrix} \dot{p}_x - \sigma_i \boldsymbol{S}_\theta \dot{\boldsymbol{\theta}} \\ \dot{p}_y + \sigma_i \boldsymbol{C}_\theta \dot{\boldsymbol{\theta}} \end{bmatrix}, \tag{2.20}$$

where c is the viscous friction coefficient. The friction forces on all links are easily expressed in the form of (2.16) as

$$\boldsymbol{f}_R = \begin{bmatrix} \boldsymbol{f}_{R,x} \\ \boldsymbol{f}_{R,y} \end{bmatrix} = -c \begin{bmatrix} \dot{\boldsymbol{X}} \\ \dot{\boldsymbol{Y}} \end{bmatrix} = -c \begin{bmatrix} l\boldsymbol{K}^T \boldsymbol{S}_\theta \dot{\boldsymbol{\theta}} + \boldsymbol{e}\dot{p}_x \\ -l\boldsymbol{K}^T \boldsymbol{C}_\theta \dot{\boldsymbol{\theta}} + \boldsymbol{e}\dot{p}_y \end{bmatrix}, \tag{2.21}$$

where we have used the expression for the link velocities given by (2.12).

Anisotropic Viscous Friction

Under anisotropic friction conditions, a link has two viscous friction coefficients, c_t and c_n, describing the friction force in the tangential (along link x axis) and normal (along link y axis) direction of the link, respectively. We define the viscous friction force on link i in the local link frame, $\boldsymbol{f}_{R,i}^{\text{link},i} \in \mathbb{R}^2$, as

$$\boldsymbol{f}_{R,i}^{\text{link},i} = -\begin{bmatrix} c_t & 0 \\ 0 & c_n \end{bmatrix} \boldsymbol{v}_i^{\text{link},i}, \tag{2.22}$$

where $\boldsymbol{v}_i^{\text{link},i} \in \mathbb{R}^2$ is the link velocity expressed in the local link frame. Using (2.3), we can express the global frame viscous friction force on link i in the form of (2.15) as

$$\begin{aligned} \boldsymbol{f}_{R,i} = \boldsymbol{f}_{R,i}^{\text{global}} &= \boldsymbol{R}_{\text{link},i}^{\text{global}} \boldsymbol{f}_{R,i}^{\text{link},i} = -\boldsymbol{R}_{\text{link},i}^{\text{global}} \begin{bmatrix} c_t & 0 \\ 0 & c_n \end{bmatrix} \boldsymbol{v}_i^{\text{link},i} \\ &= -\boldsymbol{R}_{\text{link},i}^{\text{global}} \begin{bmatrix} c_t & 0 \\ 0 & c_n \end{bmatrix} \left(\boldsymbol{R}_{\text{link},i}^{\text{global}}\right)^T \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix}, \end{aligned} \tag{2.23}$$

By performing the matrix multiplication in (2.23) and assembling the forces on all links in matrix form, the global frame viscous friction forces on the links can be written in the form of (2.16) as

$$\boldsymbol{f}_R = \begin{bmatrix} \boldsymbol{f}_{R,x} \\ \boldsymbol{f}_{R,y} \end{bmatrix} = -\begin{bmatrix} c_t \left(\boldsymbol{C}_\theta\right)^2 + c_n \left(\boldsymbol{S}_\theta\right)^2 & \left(c_t - c_n\right) \boldsymbol{S}_\theta \boldsymbol{C}_\theta \\ \left(c_t - c_n\right) \boldsymbol{S}_\theta \boldsymbol{C}_\theta & c_t \left(\boldsymbol{S}_\theta\right)^2 + c_n \left(\boldsymbol{C}_\theta\right)^2 \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{X}} \\ \dot{\boldsymbol{Y}} \end{bmatrix} \in \mathbb{R}^{2N}. \tag{2.24}$$

Note that (2.24) reduces to (2.21) in the case of isotropic friction, i.e. when $c_t = c_n = c$.

2.5 The Dynamics of the Snake Robot

The $N + 2$ degrees of freedom of the snake robot are defined by the link angles $\boldsymbol{\theta} \in \mathbb{R}^N$ and the CM position $\boldsymbol{p} \in \mathbb{R}^2$. We now present the equations of motion of the robot in terms of the acceleration of the link angles, $\ddot{\boldsymbol{\theta}}$, and the acceleration of the CM position, $\ddot{\boldsymbol{p}}$.

As illustrated in Fig. 2.2, link $i \in \{1, \ldots, N\}$ is influenced by the ground friction force $\boldsymbol{f}_{R,i} \in \mathbb{R}^2$, which acts on the CM of the link, and also the joint constraint forces $h_{x,i-1}$, $h_{y,i-1}$, $h_{x,i}$, and $h_{y,i}$, which keep the link connected to link $i - 1$ and link $i + 1$. The joint constraint forces are described in Table 2.1. Using first principles, the force balance for link i in global frame coordinates is given by

$$m\ddot{x}_i = f_{R,x,i} + h_{x,i} - h_{x,i-1}, \tag{2.25a}$$
$$m\ddot{y}_i = f_{R,y,i} + h_{y,i} - h_{y,i-1}. \tag{2.25b}$$

The force balance equations for all links may be expressed in matrix form as

$$m\ddot{\boldsymbol{X}} = \boldsymbol{f}_{R,x} + \boldsymbol{D}^T \boldsymbol{h}_x, \tag{2.26a}$$
$$m\ddot{\boldsymbol{Y}} = \boldsymbol{f}_{R,y} + \boldsymbol{D}^T \boldsymbol{h}_y. \tag{2.26b}$$

where $\boldsymbol{h}_x = [h_{x,1}, \ldots, h_{x,N}]^T \in \mathbb{R}^N$ and $\boldsymbol{h}_y = [h_{y,1}, \ldots, h_{y,N}]^T \in \mathbb{R}^N$. The link accelerations may also be expressed by differentiating (2.7) twice with respect to time, which gives

$$\boldsymbol{D}\ddot{\boldsymbol{X}} = l\boldsymbol{A}\left(\boldsymbol{C}_\theta \dot{\boldsymbol{\theta}}^2 + \boldsymbol{S}_\theta \ddot{\boldsymbol{\theta}}\right), \tag{2.27a}$$
$$\boldsymbol{D}\ddot{\boldsymbol{Y}} = l\boldsymbol{A}\left(\boldsymbol{S}_\theta \dot{\boldsymbol{\theta}}^2 - \boldsymbol{C}_\theta \ddot{\boldsymbol{\theta}}\right), \tag{2.27b}$$

where the square operator of $\dot{\boldsymbol{\theta}}^2$ means that each element of $\dot{\boldsymbol{\theta}}$ is squared ($\dot{\boldsymbol{\theta}}^2 = \mathrm{diag}(\dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}}$). We obtain the CM acceleration by differentiating (2.4) twice with respect to time, inserting (2.26), and noting that the joint constraint forces, $\boldsymbol{h}_x$ and $\boldsymbol{h}_y$, are eliminated when the link accelerations are summed (i.e. $\boldsymbol{e}^T\boldsymbol{D}^T = \boldsymbol{0}$). This gives

$$\begin{bmatrix} \ddot{p}_x \\ \ddot{p}_y \end{bmatrix} = \frac{1}{N}\begin{bmatrix} \boldsymbol{e}^T\ddot{\boldsymbol{X}} \\ \boldsymbol{e}^T\ddot{\boldsymbol{Y}} \end{bmatrix} = \frac{1}{Nm}\begin{bmatrix} \boldsymbol{e}^T\boldsymbol{f}_{R,x} \\ \boldsymbol{e}^T\boldsymbol{f}_{R,y} \end{bmatrix} = \frac{1}{Nm}\boldsymbol{E}^T\boldsymbol{f}_R. \tag{2.28}$$

This equation simply states, as would be expected, that the acceleration of the CM of the snake robot equals the sum of the external forces acting on the robot divided by its mass.

The torque balance for link i is given by

$$J\ddot{\theta}_i = u_i - u_{i-1} - l\sin\theta_i(h_{x,i} + h_{x,i-1}) + l\cos\theta_i(h_{y,i} + h_{y,i-1}), \tag{2.29}$$

where u_i and u_{i-1} are the actuator torques exerted on link i from link $i+1$ and link $i-1$, respectively. Hence, the torque balance equations for all links may be expressed in matrix form as

$$J\ddot{\boldsymbol{\theta}} = \boldsymbol{D}^T\boldsymbol{u} - l\boldsymbol{S}_\theta\boldsymbol{A}^T\boldsymbol{h}_x + l\boldsymbol{C}_\theta\boldsymbol{A}^T\boldsymbol{h}_y. \tag{2.30}$$

What now remains is to remove the joint constraint forces from (2.30). Premultiplying (2.26) by $\boldsymbol{D}$, solving for $\boldsymbol{h}_x$ and $\boldsymbol{h}_y$, and also inserting (2.27), give

$$\boldsymbol{h}_x = \left(\boldsymbol{D}\boldsymbol{D}^T\right)^{-1}\left(ml\boldsymbol{A}\left(\boldsymbol{C}_\theta\dot{\boldsymbol{\theta}}^2 + \boldsymbol{S}_\theta\ddot{\boldsymbol{\theta}}\right) - \boldsymbol{D}\boldsymbol{f}_{R,x}\right), \tag{2.31a}$$

$$\boldsymbol{h}_y = \left(\boldsymbol{D}\boldsymbol{D}^T\right)^{-1}\left(ml\boldsymbol{A}\left(\boldsymbol{S}_\theta\dot{\boldsymbol{\theta}}^2 - \boldsymbol{C}_\theta\ddot{\boldsymbol{\theta}}\right) - \boldsymbol{D}\boldsymbol{f}_{R,y}\right). \tag{2.31b}$$

By inserting (2.31) into (2.30) and solving for $\ddot{\boldsymbol{\theta}}$, the model of the snake robot can finally be written as

$$\boldsymbol{M}_\theta\ddot{\boldsymbol{\theta}} + \boldsymbol{W}\dot{\boldsymbol{\theta}}^2 - l\boldsymbol{S}_\theta\boldsymbol{K}\boldsymbol{f}_{R,x} + l\boldsymbol{C}_\theta\boldsymbol{K}\boldsymbol{f}_{R,y} = \boldsymbol{D}^T\boldsymbol{u}, \tag{2.32a}$$

$$Nm\ddot{\boldsymbol{p}} = Nm\begin{bmatrix}\ddot{p}_x\\ \ddot{p}_y\end{bmatrix} = \begin{bmatrix}\boldsymbol{e}^T\boldsymbol{f}_{R,x}\\ \boldsymbol{e}^T\boldsymbol{f}_{R,y}\end{bmatrix} = \boldsymbol{E}^T\boldsymbol{f}_R, \tag{2.32b}$$

where $\boldsymbol{f}_R$ is either the Coulomb friction force given by (2.19) or the viscous friction force given by (2.24), and where

$$\boldsymbol{M}_\theta = J\boldsymbol{I}_N + ml^2\boldsymbol{S}_\theta\boldsymbol{V}\boldsymbol{S}_\theta + ml^2\boldsymbol{C}_\theta\boldsymbol{V}\boldsymbol{C}_\theta, \tag{2.33a}$$

$$\boldsymbol{W} = ml^2\boldsymbol{S}_\theta\boldsymbol{V}\boldsymbol{C}_\theta - ml^2\boldsymbol{C}_\theta\boldsymbol{V}\boldsymbol{S}_\theta, \tag{2.33b}$$

$$\boldsymbol{V} = \boldsymbol{A}^T\left(\boldsymbol{D}\boldsymbol{D}^T\right)^{-1}\boldsymbol{A}, \tag{2.33c}$$

$$\boldsymbol{K} = \boldsymbol{A}^T\left(\boldsymbol{D}\boldsymbol{D}^T\right)^{-1}\boldsymbol{D}. \tag{2.33d}$$

By introducing the state variable $\boldsymbol{x} = \begin{bmatrix}\boldsymbol{\theta}^T & \boldsymbol{p}^T & \dot{\boldsymbol{\theta}}^T & \dot{\boldsymbol{p}}^T\end{bmatrix}^T \in \mathbb{R}^{2n+4}$, the model of the snake robot can be written compactly in state space form as

$$\dot{\boldsymbol{x}} = \begin{bmatrix}\dot{\boldsymbol{\theta}}\\ \dot{\boldsymbol{p}}\\ \ddot{\boldsymbol{\theta}}\\ \ddot{\boldsymbol{p}}\end{bmatrix} = \boldsymbol{F}\left(\boldsymbol{x}, \boldsymbol{u}\right), \tag{2.34}$$

where the elements of $\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})$ are easily found by solving (2.32a) and (2.32b) for $\ddot{\boldsymbol{\theta}}$ and $\ddot{\boldsymbol{p}}$, respectively.

2.6 Separating Actuated and Unactuated Dynamics

The model of the snake robot in (2.32) is rather complex for analysis and control design purposes. We therefore seek a transformation which allows us to write the model in a simpler form. Partial feedback linearization of underactuated systems (see e.g. Gu and Xu, 1993; Spong, 1994) consists of linearizing the dynamics corresponding to the actuated degrees of freedom of the system. We will employ this methodology in the next section. However, before partial feedback linearization can be carried out, the model of the snake robot in (2.32) must be partitioned into two parts representing the actuated and unactuated degrees of freedom, respectively. This partitioning is now carried out.

The acceleration of the CM of the snake robot, $\ddot{\boldsymbol{p}}$, belongs to the unactuated part since it is not directly influenced by the input, $\boldsymbol{u}$. The acceleration of the link angles, $\ddot{\boldsymbol{\theta}}$, represents one unactuated degree of freedom and $N-1$ actuated degrees of freedom since there are N link accelerations ($\boldsymbol{\theta} \in \mathbb{R}^N$) and only $N-1$ control inputs ($\boldsymbol{u} \in \mathbb{R}^{N-1}$). However, it is not possible to partition the equation for $\ddot{\boldsymbol{\theta}}$ in (2.32a) into an actuated and an unactuated part since the matrix $\boldsymbol{D}^T$ in front of the control input gives a direct influence between $\boldsymbol{u}$ and all the link accelerations. We therefore seek a form of the model where there is a direct influence between $\boldsymbol{u}$ and only $N-1$ link accelerations. This is achieved by modifying the choice of generalized coordinates from absolute link angles to relative joint angles. The generalized coordinates of the model in (2.32) are given by the link angles, $\boldsymbol{\theta}$, and the CM position of the snake robot, $\boldsymbol{p}$. We now replace these coordinates with

$$\boldsymbol{q}_\phi = \begin{bmatrix} \overline{\boldsymbol{\phi}} \\ \boldsymbol{p} \end{bmatrix} \in \mathbb{R}^{N+2}, \tag{2.35}$$

where $\overline{\boldsymbol{\phi}} = \begin{bmatrix} \phi_1, \ldots, \phi_{N-1}, \theta_N \end{bmatrix}^T \in \mathbb{R}^N$ contains the $N-1$ joint angles of the snake robot and the absolute link angle, $\theta_N \in \mathbb{R}$, of the head link. The joint angles were defined in Definition 2.2. The coordinate transformation

between link angles and joint angles is easily shown to be given by

$$\boldsymbol{\theta} = \boldsymbol{H}\overline{\boldsymbol{\phi}}, \quad \boldsymbol{H} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & 1 \\ 0 & 1 & 1 & \cdots & 1 & 1 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \in \mathbb{R}^{N \times N}. \tag{2.36}$$

The model of the snake robot in the new coordinates is found by inserting (2.36) into (2.32a). This gives

$$\boldsymbol{M}_\theta \boldsymbol{H}\ddot{\overline{\boldsymbol{\phi}}} + \boldsymbol{W} \operatorname{diag}\left(\boldsymbol{H}\dot{\overline{\boldsymbol{\phi}}}\right) \boldsymbol{H}\dot{\overline{\boldsymbol{\phi}}} - l\boldsymbol{S}_\theta \boldsymbol{K}\boldsymbol{f}_{R,x} + l\boldsymbol{C}_\theta \boldsymbol{K}\boldsymbol{f}_{R,y} = \boldsymbol{D}^T\boldsymbol{u}, \tag{2.37a}$$

$$Nm\ddot{\boldsymbol{p}} = \boldsymbol{E}^T\boldsymbol{f}_R, \tag{2.37b}$$

where we have used that $\dot{\boldsymbol{\theta}}^2 = \operatorname{diag}(\dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} = \operatorname{diag}(\boldsymbol{H}\dot{\overline{\boldsymbol{\phi}}})\boldsymbol{H}\dot{\overline{\boldsymbol{\phi}}}$. Finally, we premultiply (2.37a) with $\boldsymbol{H}^T$ in order to achieve the desired form of the input mapping matrix on the right-hand side by making the last of the N equations independent of the control input. This enables us to write the complete model of the snake robot as

$$\overline{\boldsymbol{M}}\left(\overline{\boldsymbol{\phi}}\right)\ddot{\boldsymbol{q}}_\phi + \overline{\boldsymbol{W}}\left(\overline{\boldsymbol{\phi}}, \dot{\overline{\boldsymbol{\phi}}}\right) + \overline{\boldsymbol{G}}\left(\overline{\boldsymbol{\phi}}\right)\boldsymbol{f}_R\left(\overline{\boldsymbol{\phi}}, \dot{\overline{\boldsymbol{\phi}}}, \dot{\boldsymbol{p}}\right) = \overline{\boldsymbol{B}}\boldsymbol{u}, \tag{2.38}$$

where

$$\boldsymbol{q}_\phi = \begin{bmatrix} \overline{\boldsymbol{\phi}} \\ \boldsymbol{p} \end{bmatrix}, \tag{2.39a}$$

$$\overline{\boldsymbol{M}}\left(\overline{\boldsymbol{\phi}}\right) = \begin{bmatrix} \boldsymbol{H}^T\boldsymbol{M}_\theta\left(\overline{\boldsymbol{\phi}}\right)\boldsymbol{H} & \boldsymbol{0}_{N\times 2} \\ \boldsymbol{0}_{2\times N} & Nm\boldsymbol{I}_2 \end{bmatrix}, \tag{2.39b}$$

$$\overline{\boldsymbol{W}}\left(\overline{\boldsymbol{\phi}}, \dot{\overline{\boldsymbol{\phi}}}\right) = \begin{bmatrix} \boldsymbol{H}^T\boldsymbol{W}\left(\overline{\boldsymbol{\phi}}\right)\operatorname{diag}\left(\boldsymbol{H}\dot{\overline{\boldsymbol{\phi}}}\right)\boldsymbol{H}\dot{\overline{\boldsymbol{\phi}}} \\ \boldsymbol{0}_{2\times 1} \end{bmatrix}, \tag{2.39c}$$

$$\overline{\boldsymbol{G}}(\boldsymbol{\phi}) = \begin{bmatrix} -l\boldsymbol{H}^T\boldsymbol{S}_{H\overline{\phi}}\boldsymbol{K} & l\boldsymbol{H}^T\boldsymbol{C}_{H\overline{\phi}}\boldsymbol{K} \\ -\boldsymbol{e}^T & \boldsymbol{0}_{1\times N} \\ \boldsymbol{0}_{1\times N} & -\boldsymbol{e}^T \end{bmatrix}, \tag{2.39d}$$

$$\overline{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{I}_{N-1} \\ \boldsymbol{0}_{3\times(N-1)} \end{bmatrix}, \tag{2.39e}$$

and where $\boldsymbol{S}_{H\overline{\phi}} = \boldsymbol{S}_\theta$ and $\boldsymbol{C}_{H\overline{\phi}} = \boldsymbol{C}_\theta$.

Remark 2.2 *It is interesting to note that premultiplying (2.37a) with* $\boldsymbol{H}^T$ *both causes the input mapping matrix* $\overline{\boldsymbol{B}}$ *to attain a desirable form and produces a symmetrical inertia matrix* $\overline{\boldsymbol{M}}$*. Had we left the model in the form of (2.37), the inertia matrix would not have been symmetrical.*

The first $N-1$ equations of (2.38) represent the dynamics of the relative joint angles of the snake robot, i.e. the *actuated* degrees of freedom of the snake robot. The last three equations represent the dynamics of the absolute orientation and position of the snake robot, i.e. the *unactuated* degrees of freedom. The model may therefore be partitioned as

$$\overline{\boldsymbol{M}}_{11}\ddot{\boldsymbol{q}}_a + \overline{\boldsymbol{M}}_{12}\ddot{\boldsymbol{q}}_u + \overline{\boldsymbol{W}}_1 + \overline{\boldsymbol{G}}_1\boldsymbol{f}_R = \boldsymbol{u}, \tag{2.40a}$$

$$\overline{\boldsymbol{M}}_{21}\ddot{\boldsymbol{q}}_a + \overline{\boldsymbol{M}}_{22}\ddot{\boldsymbol{q}}_u + \overline{\boldsymbol{W}}_2 + \overline{\boldsymbol{G}}_2\boldsymbol{f}_R = \boldsymbol{0}_{3\times 1}, \tag{2.40b}$$

where $\boldsymbol{q}_a = \begin{bmatrix}\phi_1, \ldots, \phi_{N-1}\end{bmatrix}^T \in \mathbb{R}^{N-1}$ represents the actuated degrees of freedom, $\boldsymbol{q}_u = [\theta_N, p_x, p_y]^T \in \mathbb{R}^3$ represents the unactuated degrees of freedom, $\overline{\boldsymbol{M}}_{11} \in \mathbb{R}^{(N-1)\times(N-1)}$, $\overline{\boldsymbol{M}}_{12} \in \mathbb{R}^{(N-1)\times 3}$, $\overline{\boldsymbol{M}}_{21} \in \mathbb{R}^{3\times(N-1)}$, $\overline{\boldsymbol{M}}_{22} \in \mathbb{R}^{3\times 3}$, $\overline{\boldsymbol{W}}_1 \in \mathbb{R}^{N-1}$, $\overline{\boldsymbol{W}}_2 \in \mathbb{R}^3$, $\overline{\boldsymbol{G}}_1 \in \mathbb{R}^{(N-1)\times 2N}$, and $\overline{\boldsymbol{G}}_2 \in \mathbb{R}^{3\times 2N}$.

Remark 2.3 $\overline{\boldsymbol{M}}\left(\overline{\phi}\right)$ *only depends on the relative joint angles of the snake robot and not on the absolute orientation of the head link,* θ_N*. Formally, this is a result of the fact that* θ_N *is a* cyclic *coordinate (Goldstein* et al.*, 2002). Less formally, this is quite obvious since it would not be reasonable that the inertial properties of a planar snake robot be dependent on how the snake robot is oriented in the plane. We therefore have that* $\overline{\boldsymbol{M}} = \overline{\boldsymbol{M}}\left(\boldsymbol{q}_a\right)$*.*

2.7 Partial Feedback Linearization of the Model

Based on the partitioned model in (2.40), we are now ready to transform the model of the snake robot to a simpler form through partial feedback linearization (see Gu and Xu, 1993; Spong, 1994) by introducing an input transformation which linearizes the dynamics of the actuated degrees of freedom in (2.40a). This conversion greatly simplifies the controllability and stabilizability analysis of the snake robot presented in Chapter 4. We will follow the approach presented in Reyhanoglu *et al.* (1999).

We begin by solving (2.40b) for $\ddot{\boldsymbol{q}}_u$ as

$$\ddot{\boldsymbol{q}}_u = -\overline{\boldsymbol{M}}_{22}^{-1}\left(\overline{\boldsymbol{M}}_{21}\ddot{\boldsymbol{q}}_a + \overline{\boldsymbol{W}}_2 + \overline{\boldsymbol{G}}_2\boldsymbol{f}_R\right), \tag{2.41}$$

where $\overline{\boldsymbol{M}}_{22}$ is an invertible 3×3 matrix as a consequence of the uniform positive definiteness of the system inertia matrix $\overline{\boldsymbol{M}}(\boldsymbol{q}_a)$. Inserting (2.41) into (2.40a) gives

$$\left(\overline{\boldsymbol{M}}_{11} - \overline{\boldsymbol{M}}_{12}\overline{\boldsymbol{M}}_{22}^{-1}\overline{\boldsymbol{M}}_{21}\right)\ddot{\boldsymbol{q}}_a + \overline{\boldsymbol{W}}_1 + \overline{\boldsymbol{G}}_1\boldsymbol{f}_R - \overline{\boldsymbol{M}}_{12}\overline{\boldsymbol{M}}_{22}^{-1}\left(\overline{\boldsymbol{W}}_2 + \overline{\boldsymbol{G}}_2\boldsymbol{f}_R\right) = \boldsymbol{u}. \tag{2.42}$$

Consequently, the following linearizing controller

$$\boldsymbol{u} = \left(\overline{\boldsymbol{M}}_{11} - \overline{\boldsymbol{M}}_{12}\overline{\boldsymbol{M}}_{22}^{-1}\overline{\boldsymbol{M}}_{21}\right)\overline{\boldsymbol{u}} + \overline{\boldsymbol{W}}_1 + \overline{\boldsymbol{G}}_1\boldsymbol{f}_R - \overline{\boldsymbol{M}}_{12}\overline{\boldsymbol{M}}_{22}^{-1}\left(\overline{\boldsymbol{W}}_2 + \overline{\boldsymbol{G}}_2\boldsymbol{f}_R\right), \tag{2.43}$$

where $\overline{\boldsymbol{u}} = [\overline{u}_1, \ldots, \overline{u}_{N-1}]^T \in \mathbb{R}^{N-1}$ is a new set of control inputs, enables us to rewrite (2.40a) and (2.40b) as

$$\ddot{\boldsymbol{q}}_a = \overline{\boldsymbol{u}}, \tag{2.44a}$$

$$\ddot{\boldsymbol{q}}_u = \mathcal{A}\left(\boldsymbol{q}_\phi, \dot{\boldsymbol{q}}_\phi\right) + \mathcal{B}\left(\boldsymbol{q}_a\right)\overline{\boldsymbol{u}}, \tag{2.44b}$$

where

$$\mathcal{A}\left(\boldsymbol{q}_\phi, \dot{\boldsymbol{q}}_\phi\right) = -\overline{\boldsymbol{M}}_{22}^{-1}\left(\overline{\boldsymbol{W}}_2 + \overline{\boldsymbol{G}}_2\boldsymbol{f}_R\right) \in \mathbb{R}^3, \tag{2.45a}$$

$$\mathcal{B}\left(\boldsymbol{q}_a\right) = -\overline{\boldsymbol{M}}_{22}^{-1}\overline{\boldsymbol{M}}_{21} \in \mathbb{R}^{3\times(N-1)}. \tag{2.45b}$$

This model may be written in the standard form of a control-affine system by defining $\boldsymbol{x}_1 = \boldsymbol{q}_a$, $\boldsymbol{x}_2 = \boldsymbol{q}_u$, $\boldsymbol{x}_3 = \dot{\boldsymbol{q}}_a$, $\boldsymbol{x}_4 = \dot{\boldsymbol{q}}_u$, and $\boldsymbol{x} = \left[\boldsymbol{x}_1^T, \boldsymbol{x}_2^T, \boldsymbol{x}_3^T, \boldsymbol{x}_4^T\right]^T \in \mathbb{R}^{2N+4}$. This gives

$$\dot{\boldsymbol{x}} = \begin{bmatrix}\dot{\boldsymbol{x}}_1\\ \dot{\boldsymbol{x}}_2\\ \dot{\boldsymbol{x}}_3\\ \dot{\boldsymbol{x}}_4\end{bmatrix} = \begin{bmatrix}\boldsymbol{x}_3\\ \boldsymbol{x}_4\\ \overline{\boldsymbol{u}}\\ \mathcal{A}\left(\boldsymbol{x}\right) + \mathcal{B}\left(\boldsymbol{x}_1\right)\overline{\boldsymbol{u}}\end{bmatrix} = \boldsymbol{f}\left(\boldsymbol{x}\right) + \sum_{j=1}^{N-1}\left(\boldsymbol{g}_j\left(\boldsymbol{x}_1\right)\overline{u}_j\right), \tag{2.46}$$

where

$$\boldsymbol{f}\left(\boldsymbol{x}\right) = \begin{bmatrix}\boldsymbol{x}_3\\ \boldsymbol{x}_4\\ \boldsymbol{0}_{(N-1)\times 1}\\ \mathcal{A}\left(\boldsymbol{x}\right)\end{bmatrix}, \quad \boldsymbol{g}_j\left(\boldsymbol{x}_1\right) = \begin{bmatrix}\boldsymbol{0}_{(N-1)\times 1}\\ \boldsymbol{0}_{3\times 1}\\ \boldsymbol{e}_j\\ \mathcal{B}_j\left(\boldsymbol{x}_1\right)\end{bmatrix}, \tag{2.47}$$

and where $j \in \{1, \ldots, N-1\}$, $\boldsymbol{e}_j$ denotes the jth standard basis vector in $\mathbb{R}^{N-1}$ (the jth column of $\boldsymbol{I}_{N-1}$), and $\mathcal{B}_j\left(\boldsymbol{x}_1\right)$ denotes the jth column of $\mathcal{B}\left(\boldsymbol{x}_1\right)$. In literature that considers control-affine systems, the vector $\boldsymbol{f}\left(\boldsymbol{x}\right)$ is often called the drift vector field, while the vectors $\boldsymbol{g}_j\left(\boldsymbol{x}_1\right)$ are called the control vector fields.

Remark 2.4 *We have used $\boldsymbol{x}$ to denote the state vector of the model (2.46) and also the state vector of the model (2.34) even though these two state vectors are not identical (the difference being the use of joint angles in (2.46) and link angles in (2.34)). Several models of snake robot locomotion are presented in this thesis, the presentation of which would be obscured by the introduction of a new symbol to represent the state vector of each model. Since the various models are treated separately, which means that it will always be clear from the context which elements that are contained in the state $\boldsymbol{x}$, we choose to denote the state vector in all models by $\boldsymbol{x}$.*

Remark 2.5 *The input transformation in (2.43) is nonsingular, which means that results from analysis and control design based on the partially linearized model in (2.46) are also applicable to the non-linearized model in (2.34). This is obvious since the behaviour of the model in (2.46) with a control law for $\overline{\boldsymbol{u}}$ is identical to the behaviour of the model in (2.34) with the control law for $\overline{\boldsymbol{u}}$ transformed to $\boldsymbol{u}$ according to (2.43).*

2.8 Chapter Summary

This chapter is summarized as follows:

- We have presented a mathematical model of a planar snake robot with N rigid links interconnected by $N-1$ motorized joints. The surface underneath the robot was assumed to be flat and horizontal.
- We have developed two different ground friction models, i.e. a Coulomb friction model given by (2.19) and a viscous friction model given by (2.24).
- The equations of motion of the snake robot in terms of the acceleration of the link angles, $\ddot{\boldsymbol{\theta}}$, and the acceleration of the CM position, $\ddot{\boldsymbol{p}}$, are given by (2.32).
- The equations of motion of the snake robot in terms of the acceleration of the joint angles, $\ddot{\boldsymbol{\phi}}$, the acceleration of the head link angle, $\ddot{\theta}_N$, and the acceleration of the CM position, $\ddot{\boldsymbol{p}}$, are given by (2.40). In this model, the *actuated* degrees of freedom of the snake robot are separated from the *unactuated* degrees of freedom.
- With the input transformation in (2.43), the model of the snake robot is partially feedback linearized to the simpler form given by (2.46). This form is more suitable for control design and analysis purposes.

Chapter 3

Development of a Mechanical Snake Robot for Motion across Planar Surfaces

The work underlying this thesis includes the development of two mechanical snake robots. The first snake robot, named *Wheeko*, was developed to enable experiments related to snake robot locomotion across flat surfaces. The robot is shown in Fig. 3.1 and is described in more detail in this chapter. The second snake robot, which is called *Kulko*, is described in Chapter 11 and was developed for the purpose of experiments related to snake robot locomotion in environments containing obstacles. The robots *Wheeko* and *Kulko* are, in other words, the experimental platforms used in Part I and Part II of this thesis, respectively.

The internal structure of Wheeko and Kulko are identical. The difference between the two robots concern their outer structure. The joint modules of Wheeko are equipped with passive wheels to give the robot anisotropic friction properties during motion across flat surfaces. The joint modules of Kulko, on the other hand, are covered by contact force sensors and spherical shells that give the robot a smooth outer surface, thereby allowing gliding motion in unstructured environments.

Contributions of this Chapter: Since the internal structure of Wheeko and Kulko are identical, the contributions of this chapter are included in the contributions of Chapter 11, where Kulko is described.

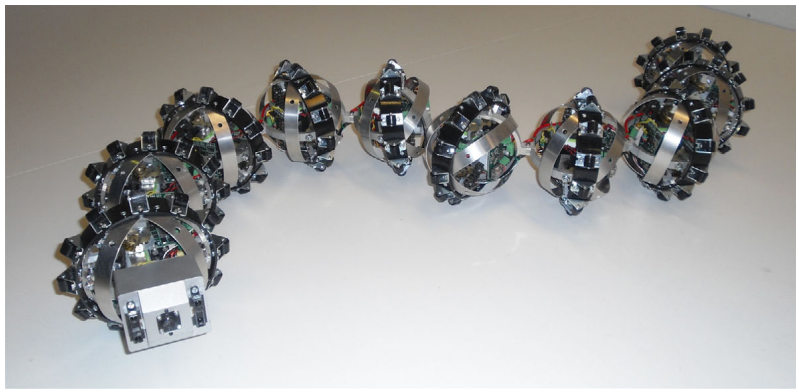


Figure 3.1: The snake robot *Wheeko* developed for locomotion across flat surfaces.

Organization of this Chapter: The joint actuation mechanism, the passive wheels, and the power and control system of Wheeko are presented in Sections 3.1, 3.2, and 3.3, respectively. The setup of the experiments carried out with the robot is presented in Section 3.4. Finally, the chapter is summarized in Section 3.5.

Publications: The material in this chapter is based on the journal papers Liljebäck *et al.* (2011*d*) and Liljebäck *et al.* (2010*k*), and on the conference papers Liljebäck *et al.* (2009*d*) and Liljebäck *et al.* (2010*c*).

3.1 The Joint Actuation Mechanism

Wheeko consists of 10 identical joint modules. The robot was developed with the same joint actuation mechanism as in Kulko to allow the same joint design to be used for both robots. As described in Chapter 11, the joint modules were designed to allow them to be covered by contact force sensors and spherical shells for the purpose of adaptive gliding motion in environments with obstacles.

As illustrated in Fig. 3.2, the articulation mechanism of each joint module has two degrees of freedom (pitch and yaw) and consists of two links supported by bearings in a steel ring. The outer diameter of the steel ring is 130 mm. Each link has a connection point at its centre that allows it to be connected to the next joint module by two screws. The axes of rotation of the two links are orthogonal and intersecting.

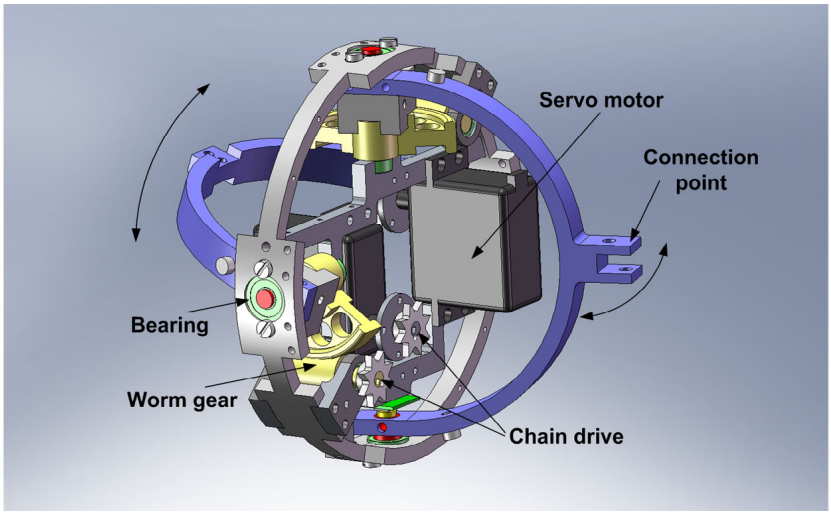


Figure 3.2: Illustration of the articulation mechanism of the joint modules.

The angle of the two moving links in the joint are measured with magnetic rotary encoders (AS5043 from austriamicrosystems). A magnet measuring 6 mm in diameter is attached to each link so that it rotates above the rotary encoder as shown in Fig. 3.3. Each encoder is attached to a custom-designed circuit board shown to the right in Fig. 3.3.

Each link is driven by a *Hitec* servo motor (HS-5955TG) by connecting the output shaft of each motor to a worm gear (gear ratio of 1:5.71) through a steel roller chain. The worm gear and the chain drive are shown in Fig. 3.4 and Fig. 3.5. The servo motors are manufactured to have a limited range of rotation (about $\pm 90°$). However, the gearing between the motors and the links requires the motors to rotate more than this limited range. The motors were therefore manually modified in order to enable them to rotate continuously. The process of modifying the servos is very simple and consists of disconnecting the output shaft of the servo from its internal potmeter and also removing a mechanical pin inside the servo that otherwise would prevent the servo from rotating continuously.

Worm gears have a disadvantage due to a high friction component in the gear system. However, worm gears are advantageous in that they may essentially produce any desired gear ratio in a single gear stage. This facilitates a compact design. In addition, a worm gear is not likely to break in contrast to e.g. spur gears. This makes the joint mechanically robust.

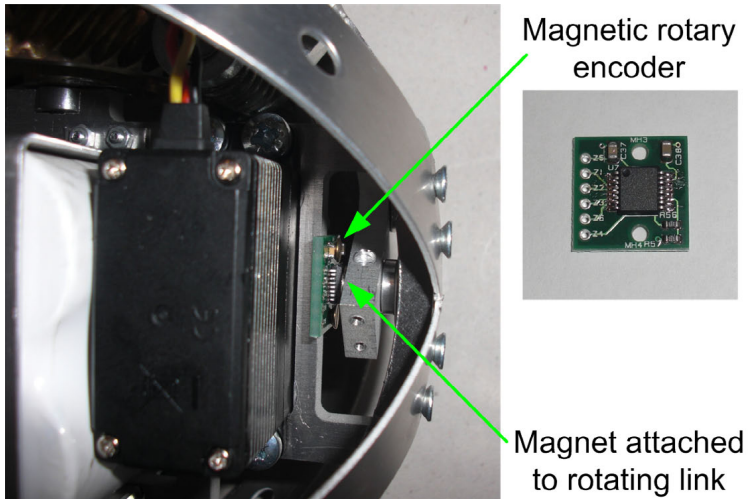


Figure 3.3: Magnetic rotary encoder used for measuring the joint angle.

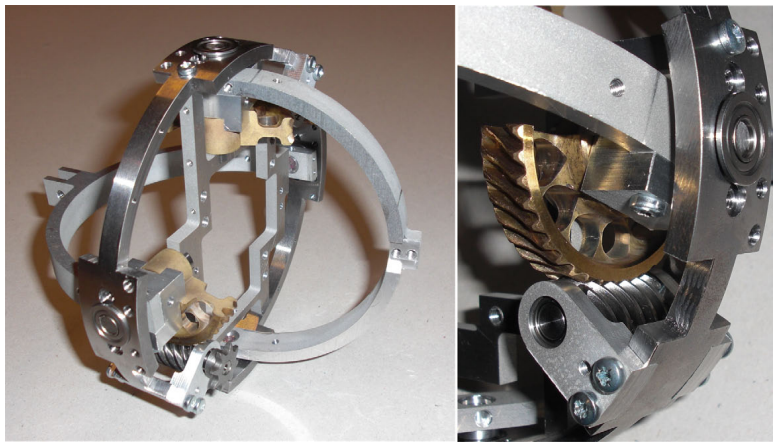


Figure 3.4: The implemented articulation mechanism of the joint modules.

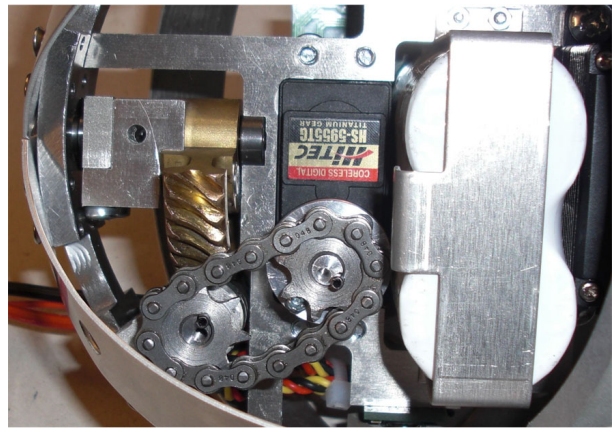


Figure 3.5: Roller chain connecting the servo motor to the worm gear.

Table 3.1: Parameters of a joint module.

Parameter	**Value**
Total weight of a joint module	960 g
Outer diameter	130 mm
Degrees of freedom	2
Max joint travel	$\pm 45°$
Max continuous joint torque	4.5 Nm
Max joint speed (no load)	$70°/_{\text{sec}}$

The steel roller chain between the servo motor and the worm gear is rated to handle forces significantly higher than the forces produced by the servo motor.

Experiments indicate that the servo motors produce a maximum continuous torque of about 1.6 Nm (at 6V supply voltage with a maximum current drain of about 3A). The rated power efficiency of the worm gears is about 75 %. This should theoretically give the joint mechanism a maximum continuous torque of around 7 Nm. However, experiments with the implemented joint mechanism indicate that the maximum continuous torque lies around 4.5 Nm. This is probably due to more friction in the worm gear than expected and also some friction in the chain drive. Table 3.1 lists the parameters characterizing the actuation mechanism.

Figure 3.6: A ring with 12 plastic wheels encloses each joint module in order to give the robot anisotropic ground friction properties.

3.2 The Passive Wheels

As shown in Fig. 3.6, each joint module of Wheeko is enclosed by a plastic ring mounted with 12 plastic wheels. The wheels are passive, i.e. not motorized, and ensure that the ground friction forces acting on the robot are anisotropic (see Section 2.4.1), i.e. that the friction coefficient characterizing the ground friction forces in the normal (sideways) direction of each joint is larger than the friction coefficient characterizing the ground friction forces in the tangential (forward) direction of the joint. Note that the wheels are able to slip sideways, so they do not introduce nonholonomic constraints in the system.

3.3 The Power and Control System

The power and control system of Wheeko is described in Chapter 11 since the system is identical to the power and control system of Kulko.

3.4 The Experimental Setup of the Snake Robot

The experiments carried out with Wheeko are described in Chapter 7 and Chapter 8. The experiments were performed on a white horizontal surface measuring about 240 cm in width and 600 cm in length. The surface is shown in Fig. 3.7. In order to measure the horizontal position of the snake

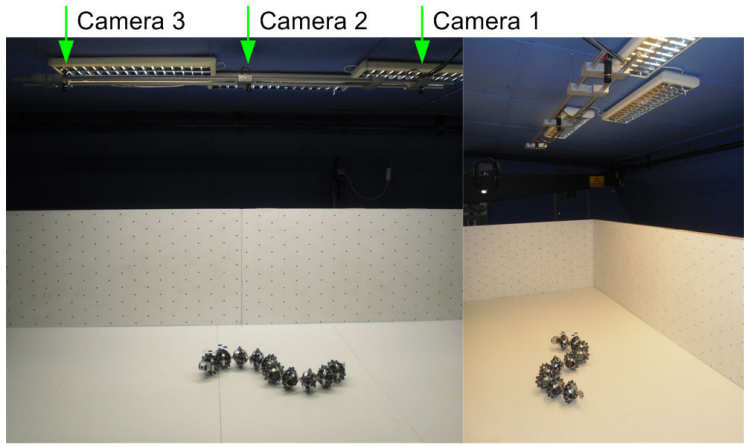


Figure 3.7: The experimental setup. Three cameras mounted in the ceiling measured the position of the snake robot on a horizontal surface measuring about 240 cm in width and 600 cm in length.

robot during the experiments, we employed the open source camera tracking software *SwisTrack* (Lochmatter *et al.*, 2008). Three firewire cameras (Unibrain Fire-i 520c) were mounted in the ceiling above the snake robot as shown in Fig. 3.7 and to the left in Fig. 3.8. The use of multiple cameras allowed for position measurements over a greater distance than the area covered by a single camera. The cameras were mounted facing downwards approximately 218 cm above the floor and 132 cm apart. The distance between the cameras was chosen so that there was a slight overlap between the images from two neighbouring cameras. Each firewire camera was sampled at 15 frames per second.

SwisTrack was configured to track three black circular markers (40 mm in diameter) mounted on the snake robot as shown to the right in Fig. 3.8. The conversion from the pixel position of a marker to the real-world position (in cm) was conducted by SwisTrack based on a specific calibration method available in this software. SwisTrack estimated the maximum position error to be about 1.9 cm and the average position error to be about 0.6 cm. The global frame coordinates of the head link, (x_N, y_N), and the absolute angle of the head link, θ_N, were calculated based on the position of the three individual markers. Knowing the position and orientation of the head of the snake robot, and also the individual joint angles, the kinematic relationships presented in Section 2.3 enabled us to calculate the CM position, $\boldsymbol{p}$, and the

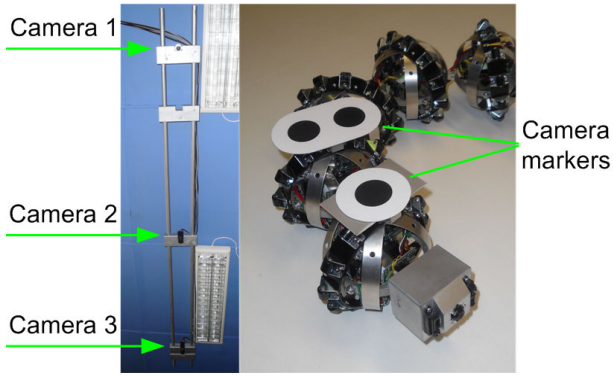


Figure 3.8: Left: The firewire cameras mounted in the ceiling above the snake robot. Right: The black markers mounted on the snake robot to allow the position to be tracked by SwisTrack.

absolute link angles, $\boldsymbol{\theta}$, of the snake robot. We ran three separate instances of SwisTrack in order to process data from all three cameras and developed our own software in order to merge the output from each SwisTrack instance into the final position measurement of the snake robot.

3.5 Chapter Summary

This chapter is summarized as follows:

- We have presented the design of the snake robot *Wheeko*, which was developed for the purpose of experiments related to snake robot locomotion across flat surfaces.
- The robot consists of 10 identical joint modules, and each joint has two degrees of freedom (pitch and yaw).
- The joint modules are equipped with passive wheels to give the robot anisotropic ground friction properties.
- The internal structure of Wheeko is identical to the internal structure of the snake robot *Kulko*, which is described in Chapter 11.

Chapter 4

Analysis and Synthesis of Snake Robot Locomotion

Research on snake robots has been conducted for several decades. However, our understanding of snake locomotion so far is for the most part based on empirical studies of biological snakes and simulation-based synthesis of relationships between parameters of the snake robot. Armed with the mathematical model of the snake robot presented in Chapter 2, we attempt in this chapter to contribute to the understanding of snake robots by employing nonlinear system analysis tools for investigating fundamental properties of their dynamics. We will also derive several interesting properties of snake robot locomotion simply by investigating the equations of motion of the robot, some of which will be instrumental in the development of a *simplified* model in Chapter 6.

In this chapter, we investigate the motion pattern which is most common among biological snakes, namely *lateral undulation* (see Section 1.2.2). This motion pattern is considered in the majority of the snake robot literature, and will also receive much attention throughout this thesis. A well-known mathematical description of the shape of a snake during lateral undulation was presented in Hirose (1993) based on empirical studies of biological snakes. In this chapter, we develop analytical results that support this mathematical description.

Contributions of this Chapter: The first contribution of this chapter is a stabilizability analysis that proves that any asymptotically stabilizing control law for a planar snake robot to an equilibrium point must be *time-varying*, i.e. not of pure-state feedback type (see Theorem 4.3).

The second contribution is a controllability analysis of planar snake robots influenced by viscous ground friction forces. The analysis shows that a snake robot is *not* controllable when the viscous ground friction is *isotropic* (see Theorem 4.4), but that a snake robot becomes *strongly accessible* when the viscous ground friction is *anisotropic* (see Theorem 4.5). The analysis also shows that the snake robot does *not* satisfy sufficient conditions for *small-time local controllability* (see Theorem 4.6). To our best knowledge, no formal controllability analysis has previously been reported for the position and link angles of a wheel-less snake robot influenced by ground friction. The results from the controllability analysis prove that propulsion of a snake robot under viscous friction conditions requires the friction to be *anisotropic*, and also that the joint angles of a snake robot should be *out of phase* during snake locomotion. These claims have been assumed in the snake robot literature, but have never before been formally proven.

The third contribution is the development of a simple relationship between link velocities *normal* to the direction of motion and *propulsive forces* in the direction of motion. This relationship explains how snake robots influenced by anisotropic ground friction are able to locomote forward on a planar surface, and enables us to derive several fundamental properties of snake locomotion (see Properties 4.2, 4.3, and 4.4). In our opinion, previously published research on snake robots has not presented an explicit mathematical description that *easily* explains how a snake robot achieves forward propulsion.

As a fourth and final contribution, we use the derived properties of snake robot locomotion to provide support for the empirically derived mathematical description of lateral undulation proposed in Hirose (1993). We also identify an important property concerning the turning motion of a snake robot (see Property 4.7), and a property related to the relative displacements of the links during lateral undulation (see Property 4.8).

Organization of this Chapter: Section 4.1 introduces some selected tools for analysing controllability of nonlinear systems. Section 4.2 and Section 4.3 study, respectively, the stabilizability and controllability properties of planar snake robots. Section 4.4 explains how a snake robot is able to move forward, while Section 4.5 deduces how the links of the robot can be moved in order to achieve forward propulsion. The deduced link motion is identified as the gait pattern *lateral undulation* in Section 4.6, and two different control laws for the joints

of the snake robot are presented in Section 4.7. The properties related to the turning motion and the relative displacements of the links during lateral undulation are derived in Section 4.8 and Section 4.9, respectively. Finally, the chapter is summarized in Section 4.10.

Publications: The material in this chapter is based on the journal papers Liljebäck *et al.* (2011*b*) and Liljebäck *et al.* (2011*d*), and on the conference papers Liljebäck *et al.* (2009*b*), Liljebäck *et al.* (2009*c*), and Liljebäck *et al.* (2010*j*).

4.1 Introduction to Nonlinear Controllability Analysis

This section presents a brief summary of selected tools for analysing the controllability of nonlinear systems. The summary given below is formulated in an intuitive form that aims to be easily understandable for readers unaccustomed with nonlinear controllability analysis. For a rigorous presentation, the readers are referred to Bianchini and Stefani (1990); Nijmeijer and Schaft (1990); Sussmann (1987).

Analysing the controllability of a *linear* system is straightforward and involves checking if the system matrices satisfy the *Kalman rank condition* (see Nijmeijer and Schaft, 1990). However, studying the controllability of a *nonlinear* system is far more complex and constitutes an active area of research. In the following, we summarize important controllability concepts for control-affine nonlinear systems, i.e. systems of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}\left(\boldsymbol{x}\right) + \sum_{j=1}^{m} \boldsymbol{g}_j\left(\boldsymbol{x}\right) \overline{u}_j, \tag{4.1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector, $\overline{\boldsymbol{u}} \in \mathbb{R}^m$ is the control input, $\boldsymbol{f}\left(\boldsymbol{x}\right) \in \mathbb{R}^n$ is the drift vector field, and $\boldsymbol{g}_j\left(\boldsymbol{x}\right) \in \mathbb{R}^n$, $j \in \{1, \ldots, m\}$, are the control vector fields of the system. Note that the model of the snake robot was written in this form in Section 2.7.

A nonlinear system is said to be *controllable* if there exist admissible control inputs that will move the system between two arbitrary states in finite time. However, general conditions for this kind of controllability that are both necessary and sufficient do not exist. Nonlinear controllability is instead typically analysed by investigating the local behaviour of the system near equilibrium points.

The simplest approach to studying the controllability of the system (4.1) is to linearize the system about an equilibrium point $\boldsymbol{x}^e$ according to

$$\dot{\boldsymbol{z}} = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{B}\overline{\boldsymbol{u}}, \tag{4.2}$$

where $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{x}^e$, $\boldsymbol{A} = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}^e}$, and $\boldsymbol{B} = [\boldsymbol{g}_1, \ldots, \boldsymbol{g}_m]$. If the linearized system satisfies the Kalman rank condition at $\boldsymbol{x}^e$, which requires that the controllability matrix given by $[\boldsymbol{B},\ \boldsymbol{A}\boldsymbol{B},\ \boldsymbol{A}^2\boldsymbol{B},\ \ldots,\ \boldsymbol{A}^{n-1}\boldsymbol{B}]$ has full rank, then the nonlinear system (4.1) is controllable in the sense that the set of states that can be reached from $\boldsymbol{x}^e$ contains a neighborhood of $\boldsymbol{x}^e$. Unfortunately, many underactuated systems do not have a controllable linearization. Moreover, a nonlinear system can be controllable even though its linearization is not.

A necessary (but not sufficient) condition for controllability from a state $\boldsymbol{x}_0$ (not necessarily an equilibrium) is that the nonlinear system satisfies the *Lie algebra rank condition* (LARC), also called the *accessibility rank condition* (see Nijmeijer and Schaft, 1990). If this is the case, the system is said to be *locally accessible* from $\boldsymbol{x}_0$. This property means that the space that the system can reach within any time $T > 0$ is fully n-dimensional, i.e. the reachable space from $\boldsymbol{x}_0$ has a dimension equal to the dimension of the state space. A slightly stronger property is *strong accessibility*, which means that the space that the system can reach in *exactly* time T for any $T > 0$ is fully n-dimensional.

Accessibility of a nonlinear system is investigated by computing the *accessibility algebra*, here denoted $\boldsymbol{\Delta}$, of the system. Computation of $\boldsymbol{\Delta}$ requires knowledge of the *Lie bracket* (see Nijmeijer and Schaft, 1990), which is now briefly explained. The drift and control vector fields of the nonlinear system (4.1) indicate directions in which the state $\boldsymbol{x}$ can move. These directions will generally only span a subset of the complete state space. However, through combined motion along two or more of these vector fields, it is possible for the system to move in directions not spanned by the original system vector fields. The Lie bracket between two vector fields $\boldsymbol{Y}$ and $\boldsymbol{Z}$ produces a new vector field defined as $[\boldsymbol{Y}, \boldsymbol{Z}] = \frac{\partial \boldsymbol{Z}}{\partial \boldsymbol{x}}\boldsymbol{Y} - \frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{x}}\boldsymbol{Z}$. When $\boldsymbol{Y}$ and $\boldsymbol{Z}$ are any of the system vector fields, the Lie bracket $[\boldsymbol{Y}, \boldsymbol{Z}]$ *approximates* the net motion produced when the system follows these two vector fields in an alternating fashion. The classical example is parallel parking with a car, where sideways motion of the car may be achieved through an alternating turning and forward/backward motion. Note that Lie brackets can be computed from other Lie brackets, thereby producing nested Lie brackets. The *accessibility algebra*, $\boldsymbol{\Delta}$, is a set of vector fields

composed of the system vector fields, $\boldsymbol{f}$ and $\boldsymbol{g}_j$, the Lie brackets between the system vector fields, and also higher order Lie brackets generated by nested Lie brackets. The LARC is satisfied at $\boldsymbol{x}_0$ if the vector fields in $\boldsymbol{\Delta}(\boldsymbol{x}_0)$ span the entire n-dimensional state space ($\dim(\text{span}(\boldsymbol{\Delta})) = n$). The following result is proved in Nijmeijer and Schaft (1990):

Theorem 4.1 *The system (4.1) is* locally accessible *from $\boldsymbol{x}_0$ if and only if the LARC is satisfied at $\boldsymbol{x}_0$. The system is* locally strongly accessible *if the drift field $\boldsymbol{f}$ by itself (i.e. unbracketed) is not included in the* accessibility algebra.

Accessibility does *not* imply controllability since it only infers conclusions on the dimension of the reachable space from $\boldsymbol{x}_0$. Accessibility is, however, a necessary (but not sufficient) condition for *small-time local controllability* (STLC). STLC is desirable since it is in fact a stronger property than controllability. If a system is STLC, then the control input can steer the system in any direction in an arbitrarily small amount of time. For second-order systems, STLC is only considered from equilibrium states since it is generally not possible for a second-order system to instantly move in one direction if it already has a velocity in the opposite direction. For example, an airplane in flight is not STLC since it cannot instantly move opposite to its direction of motion.

Sufficient conditions for STLC were presented by Sussmann (1987), and later extended by Bianchini and Stefani (1990). We now summarize these conditions. For any Lie bracket term $\boldsymbol{b} \in \mathbb{R}^n$ generated from the system vector fields, define the θ-degree of $\boldsymbol{b}$, denoted $\delta_\theta(\boldsymbol{b})$, and the l-degree of $\boldsymbol{b}$, denoted $\delta_l(\boldsymbol{b})$, as

$$\delta_\theta(\boldsymbol{b}) = \frac{1}{\theta}\delta^0(\boldsymbol{b}) + \sum_{j=1}^{m}\delta^j(\boldsymbol{b}), \tag{4.3}$$

$$\delta_l(\boldsymbol{b}) = \sum_{j=0}^{m} l_j \delta^j(\boldsymbol{b}), \tag{4.4}$$

respectively, where $\delta^0(\boldsymbol{b})$ is the number of times the drift vector field $\boldsymbol{f}$ appears in the bracket $\boldsymbol{b}$, $\delta^j(\boldsymbol{b})$ is the number of times the control vector field $\boldsymbol{g}_j$ appears in the bracket $\boldsymbol{b}$, θ is an arbitrary number satisfying $\theta \in [1, \infty)$, and l_j is an arbitrary number satisfying $l_j \geq l_0 \geq 0$, $\forall\, j \in \{0, \ldots, m\}$. The bracket $\boldsymbol{b}$ is said to be *bad* if $\delta^0(\boldsymbol{b})$ is odd and $\delta^1(\boldsymbol{b}), \ldots, \delta^m(\boldsymbol{b})$ are all even. A bracket is *good* if it is not bad. As an example, we have that the bracket

$\left[\boldsymbol{g}_j, [\boldsymbol{f}, \boldsymbol{g}_k]\right]$ is *bad* for $j = k$ and *good* for $j \neq k$. This classification is motivated by the fact that a bad bracket *may* have directional constraints. E.g. the drift vector $\boldsymbol{f}$ is *bad* because it only allows motion in its positive direction and not in its negative direction, $-\boldsymbol{f}$. A bad bracket is said to be θ-*neutralized* (resp. l-*neutralized*) if it can be written as a linear combination of good brackets of lower θ-degree (resp. l-degree). The *Sussmann condition* and the *Bianchini and Stefani condition* for STLC are now combined in the following theorem:

Theorem 4.2 *The system (4.1) is* small-time locally controllable *(STLC) from an equilibrium point* $\boldsymbol{x}^e$ *(* $\boldsymbol{f}\left(\boldsymbol{x}^e\right) = 0$*) if the LARC is satisfied at* $\boldsymbol{x}^e$ *and either all* bad *brackets are* θ-neutralized *(Sussmann, 1987) or all* bad *brackets are* l-neutralized *(Bianchini and Stefani, 1990).*

4.2 Stabilizability Properties of Planar Snake Robots

In this section, we present a fundamental theorem concerning the properties of an asymptotically stabilizing control law for snake robots to any equilibrium point. The model of the snake robot is given by (2.46) and we make no assumptions regarding the ground friction forces other than that they are given on the form of (2.16).

The equation (2.46) maps the state $\boldsymbol{x}$ and the control input $\overline{\boldsymbol{u}}$ of the robot into the resulting derivative of the state vector, $\dot{\boldsymbol{x}}$. For any equilibrium point $(\boldsymbol{x}_1 = \boldsymbol{x}_1^e, \boldsymbol{x}_2 = \boldsymbol{x}_2^e, \boldsymbol{x}_3 = \boldsymbol{0}, \boldsymbol{x}_4 = \boldsymbol{0})$, where $(\boldsymbol{x}_1^e, \boldsymbol{x}_2^e)$ is the configuration of the system at the equilibrium point, we have that $\dot{\boldsymbol{x}} = \boldsymbol{0}$. A well-known result presented in Brockett (1983) states that a necessary condition for the existence of a *time-invariant* (i.e. not explicitly dependent on time) *continuous* state feedback law, $\overline{\boldsymbol{u}} = \overline{\boldsymbol{u}}(\boldsymbol{x})$, that makes $(\boldsymbol{x}_1^e, \boldsymbol{x}_2^e, \boldsymbol{0}, \boldsymbol{0})$ asymptotically stable, is that the image of the mapping $(\boldsymbol{x}, \overline{\boldsymbol{u}}) \mapsto \dot{\boldsymbol{x}}$ contains some neighbourhood of $\dot{\boldsymbol{x}} = \boldsymbol{0}$ (this requirement is explained below). A result presented in Coron and Rosier (1994) states that a control system that can be asymptotically stabilized (in the Filippov sense) by a *time-invariant discontinuous* state feedback law can be asymptotically stabilized by a *time-varying continuous* state feedback law. If, moreover, the control system is *affine* (i.e. linear with respect to the control input), then it can be asymptotically stabilized by a *time-invariant continuous* state feedback law. We now employ these results to prove the following fundamental theorem for planar snake robots:

Theorem 4.3 *An asymptotically stabilizing control law for a planar snake robot described by (2.46) to any equilibrium point must be time-varying, i.e. of the form* $\overline{\boldsymbol{u}} = \overline{\boldsymbol{u}}(\boldsymbol{x}, t)$.

Proof. The result in Brockett (1983) states that the mapping $(\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \boldsymbol{x}_3,\ \boldsymbol{x}_4,\ \overline{\boldsymbol{u}}) \mapsto (\boldsymbol{x}_3,\ \boldsymbol{x}_4,\ \overline{\boldsymbol{u}},\ \mathcal{A}(\boldsymbol{x}) + \mathcal{B}(\boldsymbol{x}_1)\,\overline{\boldsymbol{u}})$ must map an arbitrary neighbourhood of $(\boldsymbol{x}_1 = \boldsymbol{x}_1^e,\ \boldsymbol{x}_2 = \boldsymbol{x}_2^e,\ \boldsymbol{x}_3 = \boldsymbol{0},\ \boldsymbol{x}_4 = \boldsymbol{0},\ \overline{\boldsymbol{u}} = \boldsymbol{0})$ onto a neighbourhood of $\dot{\boldsymbol{x}} = \boldsymbol{0}$, i.e. a neighbourhood of $(\boldsymbol{x}_3 = \boldsymbol{0},\ \boldsymbol{x}_4 = \boldsymbol{0},\ \overline{\boldsymbol{u}} = \boldsymbol{0},\ \mathcal{A}(\boldsymbol{x}) + \mathcal{B}(\boldsymbol{x}_1)\,\overline{\boldsymbol{u}} = \boldsymbol{0})$. For this to be true, points of the form $(\boldsymbol{x}_3 = \boldsymbol{0},\ \boldsymbol{x}_4 = \boldsymbol{0},\ \overline{\boldsymbol{u}} = \boldsymbol{0},\ \mathcal{A}(\boldsymbol{x}) + \mathcal{B}(\boldsymbol{x}_1)\,\overline{\boldsymbol{u}} = \boldsymbol{\epsilon})$ must be contained in this mapping for some arbitrary $\boldsymbol{\epsilon} \neq \boldsymbol{0}$ because points of this form are contained in every neighbourhood of $\dot{\boldsymbol{x}} = \boldsymbol{0}$. In other words, there must exist some state and control input $(\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \boldsymbol{x}_3,\ \boldsymbol{x}_4,\ \overline{\boldsymbol{u}})$ that can be mapped to points of the form $(\boldsymbol{x}_3 = \boldsymbol{0},\ \boldsymbol{x}_4 = \boldsymbol{0},\ \overline{\boldsymbol{u}} = \boldsymbol{0},\ \mathcal{A}(\boldsymbol{x}) + \mathcal{B}(\boldsymbol{x}_1)\,\overline{\boldsymbol{u}} = \boldsymbol{\epsilon} \neq \boldsymbol{0})$. However, these points do not exist for the system in (2.46) because $\boldsymbol{x}_3 = \boldsymbol{0}$, $\boldsymbol{x}_4 = \boldsymbol{0}$, and $\overline{\boldsymbol{u}} = \boldsymbol{0}$ means that $\mathcal{A}(\boldsymbol{x}) + \mathcal{B}(\boldsymbol{x}_1)\,\overline{\boldsymbol{u}} = \boldsymbol{0} \neq \boldsymbol{\epsilon}$. Hence, the snake robot *cannot* be asymptotically stabilized to $(\boldsymbol{x}_1 = \boldsymbol{x}_1^e,\ \boldsymbol{x}_2 = \boldsymbol{x}_2^e,\ \boldsymbol{x}_3 = \boldsymbol{0},\ \boldsymbol{x}_4 = \boldsymbol{0})$ by a *time-invariant continuous* state feedback law. Moreover, since the system in (2.46) is affine and *cannot* be asymptotically stabilized by a *time-invariant continuous* state feedback law, the result in Coron and Rosier (1994) proves that the system can neither be asymptotically stabilized by a *time-invariant discontinuous* state feedback law. We can therefore conclude that an asymptotically stabilizing control law for a planar snake robot to any equilibrium point must be time-varying, i.e. of the form $\overline{\boldsymbol{u}} = \overline{\boldsymbol{u}}(\boldsymbol{x}, t)$. ■

Remark 4.1 *Theorem 4.3 is independent of the choice of friction model and applies to any planar snake robot described by a friction model with the property that* $\mathcal{A}(\boldsymbol{x}^e) = \boldsymbol{0}$ *for any equilibrium point* $\boldsymbol{x}^e$.

4.3 Controllability Analysis of Planar Snake Robots

This section studies the controllability of planar snake robots described by the model (2.46). As described in Section 2.4.1, we conjecture that the motion of a snake robot is *qualitatively* (although not *quantitatively*) similar with anisotropic viscous friction as with anisotropic Coulomb friction. In the following, we therefore assume that the ground friction is viscous since the simplicity of the viscous model compared to the Coulomb model is more suitable for controllability analysis purposes.

4.3.1 Controllability with Isotropic Viscous Friction

We begin the controllability analysis of the snake robot by first assuming that the viscous ground friction is isotropic. In this case, it turns out that the equations of motion take on a particularly simple form that enables us to study controllability through pure inspection of the equations of motion. From (2.32b), the acceleration of the CM of the snake robot is given as

$$\begin{bmatrix} \ddot{p}_x \\ \ddot{p}_y \end{bmatrix} = \frac{1}{Nm} \begin{bmatrix} \boldsymbol{e}^T \boldsymbol{f}_{R,x} \\ \boldsymbol{e}^T \boldsymbol{f}_{R,y} \end{bmatrix} = \frac{1}{Nm} \begin{bmatrix} \sum_{i=1}^{N} \boldsymbol{f}_{R,x,i} \\ \sum_{i=1}^{N} \boldsymbol{f}_{R,y,i} \end{bmatrix}. \tag{4.5}$$

By inserting (2.20) into (4.5), the CM acceleration of the robot is given as

$$\begin{bmatrix} \ddot{p}_x \\ \ddot{p}_y \end{bmatrix} = \frac{c}{Nm} \begin{bmatrix} -N\dot{p}_x + \left(\sum_{i=1}^{N} \sigma_i \right) \boldsymbol{S}_\theta \dot{\boldsymbol{\theta}} \\ -N\dot{p}_y - \left(\sum_{i=1}^{N} \sigma_i \right) \boldsymbol{C}_\theta \dot{\boldsymbol{\theta}} \end{bmatrix} = -\frac{c}{m} \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \end{bmatrix} \tag{4.6}$$

because it may be shown that $\sum_{i=1}^{N} \sigma_i = 0$. This enables us to state the following theorem:

Theorem 4.4 *A planar snake robot influenced by* isotropic *viscous ground friction is* not controllable*.*

Proof. In order to control the position, the snake robot must accelerate its CM. From (4.6), it is clear that the CM acceleration is proportional to the CM velocity. If the robot starts from rest ($\dot{\boldsymbol{p}} = \boldsymbol{0}$), it is therefore impossible to achieve acceleration of the CM. The position of the robot is in other words completely uncontrollable in this case, which renders the robot uncontrollable. ■

4.3.2 Controllability with Anisotropic Viscous Friction

The equations of motion of the snake robot in (2.46) become far more complex under anisotropic friction conditions. We therefore employ the controllability concepts presented in Section 4.1 and begin by computing the Lie brackets of the system vector fields. The drift vector field $\boldsymbol{f}(\boldsymbol{x})$ and the control vector fields $\boldsymbol{g}_j(\boldsymbol{x}_1)$, $j \in \{1, \ldots, m\}$, of the snake robot are defined in (2.46). As explained in Section 4.1, Lie bracket computation

involves partial derivatives of the components of the vector fields. These computations can be carried out without dealing with the complex expressions contained in $\mathcal{A}(\boldsymbol{x})$ and $\mathcal{B}(\boldsymbol{x}_1)$ given by (2.45) since we only need to know which variables each vector field depends on. As an example, consider column j of $\mathcal{B}(\boldsymbol{x}_1)$. Since we know that it only depends on $\boldsymbol{x}_1$, we may immediately write $\frac{\partial \mathcal{B}_j(\boldsymbol{x}_1)}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial \mathcal{B}_j(\boldsymbol{x}_1)}{\partial \boldsymbol{x}_1} & \mathbf{0}_{3\times(N+5)} \end{bmatrix}$. This methodology enables us to compute the following Lie brackets of the system vector fields (evaluated at an equilibrium point):

$$\big[\boldsymbol{f}, \boldsymbol{g}_j\big]^{\dot{\boldsymbol{q}}_\phi = \mathbf{0}} = \begin{bmatrix} -\boldsymbol{e}_j \\ -\mathcal{B}_j \\ \mathbf{0}_{(N-1)\times 1} \\ -\mathcal{C}_j \end{bmatrix}, \tag{4.7a}$$

$$\big[\boldsymbol{f}, \big[\boldsymbol{f}, \boldsymbol{g}_j\big]\big]^{\dot{\boldsymbol{q}}_\phi = \mathbf{0}} = \begin{bmatrix} \mathbf{0}_{(N-1)\times 1} \\ \mathcal{C}_j \\ \mathbf{0}_{(N-1)\times 1} \\ \frac{\partial \mathcal{A}}{\partial \boldsymbol{x}_4}\mathcal{C}_j \end{bmatrix}, \tag{4.7b}$$

$$\big[[\boldsymbol{f}, \boldsymbol{g}_j], [\boldsymbol{f}, \boldsymbol{g}_k]\big]^{\dot{\boldsymbol{q}}_\phi = \mathbf{0}} = \begin{bmatrix} \mathbf{0}_{(N-1)\times 1} \\ \mathcal{D}_{jk} \\ \mathbf{0}_{(N-1)\times 1} \\ \mathcal{E}_{jk} \end{bmatrix}, \tag{4.7c}$$

where $j, k \in \{1, \ldots, N-1\}$ and

$$\mathcal{C}_j = \frac{\partial \mathcal{A}}{\partial \boldsymbol{x}_3}\boldsymbol{e}_j + \frac{\partial \mathcal{A}}{\partial \boldsymbol{x}_4}\mathcal{B}_j, \tag{4.8}$$

$$\mathcal{D}_{jk} = \frac{\partial \mathcal{B}_k}{\partial \boldsymbol{x}_1}\boldsymbol{e}_j - \frac{\partial \mathcal{B}_j}{\partial \boldsymbol{x}_1}\boldsymbol{e}_k, \tag{4.9}$$

$$\mathcal{E}_{jk} = \frac{\partial \mathcal{C}_k}{\partial \boldsymbol{x}_1}\boldsymbol{e}_j - \frac{\partial \mathcal{C}_j}{\partial \boldsymbol{x}_1}\boldsymbol{e}_k + \frac{\partial \mathcal{C}_k}{\partial \boldsymbol{x}_2}\mathcal{B}_j - \frac{\partial \mathcal{C}_j}{\partial \boldsymbol{x}_2}\mathcal{B}_k + \frac{\partial \mathcal{C}_k}{\partial \boldsymbol{x}_4}\mathcal{C}_j - \frac{\partial \mathcal{C}_j}{\partial \boldsymbol{x}_4}\mathcal{C}_k \tag{4.10}$$

The Lie brackets have been evaluated at zero velocity ($\dot{\boldsymbol{q}}_\phi = \mathbf{0}$) since we are interested in controllability from an equilibrium point. The above vector fields represent our choice of vector fields to be contained in the *accessibility algebra*, $\boldsymbol{\Delta}$, of the system. To construct $\boldsymbol{\Delta}$ of full rank, we need to find $(2N+4)$ independent vector fields since the snake robot has a $(2N+4)$-dimensional state space. Each of the four types of vector fields above represent $(N-1)$ vector fields. Solving $4(N-1) \geq 2N+4$ gives that our analysis is only valid if the snake robot has $N \geq 4$ links. This is a mild requirement, however, since a snake robot generally has more than four links. In the remainder of this controllability analysis, we assume that the robot consists

of exactly $N = 4$ links (and thereby $N-1 = 3$ active joints) and argue that the following controllability results will also be valid for snake robots with more links. In particular, a robot with $N > 4$ links can behave as a robot with $N = 4$ links by fixing $(N-4)$ joint angles at zero degrees and allowing the remaining three joint angles to move. This means that controllability of the robot with $N = 4$ is a sufficient although not necessary condition for controllability with $N > 4$.

With $N = 4$ links, the system has a $(2N+4) = 12$-dimensional state space. The system satisfies the *Lie algebra rank condition* (LARC) if the above vector fields span a 12-dimensional space. We therefore assemble the 12 vector fields into the following matrix, which represents the *accessibility algebra* of the system evaluated at an equilibrium point $\boldsymbol{x}^e$:

$$\begin{aligned}\boldsymbol{\Delta}(\boldsymbol{x}^e) &= [\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3, [\boldsymbol{f}, \boldsymbol{g}_1], [\boldsymbol{f}, \boldsymbol{g}_2], [\boldsymbol{f}, \boldsymbol{g}_3], \\ &\quad [\boldsymbol{f}, [\boldsymbol{f}, \boldsymbol{g}_1]], [\boldsymbol{f}, [\boldsymbol{f}, \boldsymbol{g}_2]], [\boldsymbol{f}, [\boldsymbol{f}, \boldsymbol{g}_3]], \\ &\quad [[\boldsymbol{f}, \boldsymbol{g}_1], [\boldsymbol{f}, \boldsymbol{g}_2]], [[\boldsymbol{f}, \boldsymbol{g}_1], [\boldsymbol{f}, \boldsymbol{g}_3]], [[\boldsymbol{f}, \boldsymbol{g}_2], [\boldsymbol{f}, \boldsymbol{g}_3]]] \\ &= \begin{bmatrix} \boldsymbol{0}_{3\times3} & -\boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & -\mathcal{B} & \mathcal{C} & \mathcal{D} \\ \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \mathcal{B} & -\mathcal{C} & \frac{\partial \mathcal{A}}{\partial \boldsymbol{x}_4}\mathcal{C} & \mathcal{E} \end{bmatrix} \in \mathbb{R}^{12\times12}, \end{aligned} \tag{4.11}$$

where

$$\mathcal{C} = \frac{\partial \mathcal{A}}{\partial \boldsymbol{x}_3} + \frac{\partial \mathcal{A}}{\partial \boldsymbol{x}_4}\mathcal{B} \in \mathbb{R}^{3\times3}, \tag{4.12}$$

$$\mathcal{D} = \begin{bmatrix} \mathcal{D}_{12} & \mathcal{D}_{13} & \mathcal{D}_{23} \end{bmatrix} \in \mathbb{R}^{3\times3}, \tag{4.13}$$

$$\mathcal{E} = \begin{bmatrix} \mathcal{E}_{12} & \mathcal{E}_{13} & \mathcal{E}_{23} \end{bmatrix} \in \mathbb{R}^{3\times3}. \tag{4.14}$$

We now state the following theorem regarding the accessibility of the snake robot:

Theorem 4.5 *A planar snake robot with $N \geq 4$ links influenced by* anisotropic *viscous ground friction ($c_t \neq c_n$) is* locally strongly accessible *from any equilibrium point $\boldsymbol{x}^e$ ($\dot{\boldsymbol{q}}_\phi = \boldsymbol{0}$) satisfying* $\det(\mathcal{C}) \neq 0$ *and* $\det\left(\mathcal{E} - \frac{\partial \mathcal{A}}{\partial \boldsymbol{x}_4}\mathcal{D}\right) \neq 0$, *where* $\det(*)$ *denotes the determinant evaluated at $\boldsymbol{x}^e$.*

Proof. By Theorem 4.1, the system is *locally strongly accessible* from $\boldsymbol{x}^e$ if $\boldsymbol{\Delta}(\boldsymbol{x}^e)$, given by (4.11), has full rank, i.e. spans a 12-dimensional space. The proof is complete if we can show that this is the case as long as $\det(\mathcal{C}) \neq 0$ and $\det\left(\mathcal{E} - \frac{\partial \mathcal{A}}{\partial \boldsymbol{x}_4}\mathcal{D}\right) \neq 0$ at $\boldsymbol{x}^e$. The matrix $\boldsymbol{\Delta}(\boldsymbol{x}^e)$ has full rank when all its

columns are linearly independent. By investigating the particular structure of $\boldsymbol{\Delta}(\boldsymbol{x}^e)$, we see that the first and third row contains an identity matrix and then zeros in the remaining elements of these rows. It is therefore impossible to write the columns containing the two identity matrices as linear combinations of other columns. We can therefore conclude that any linear dependence between the columns of $\boldsymbol{\Delta}(\boldsymbol{x}^e)$ must be caused by linear dependence between the columns of the following submatrix of $\boldsymbol{\Delta}(\boldsymbol{x}^e)$:

$$\widetilde{\boldsymbol{\Delta}}(\boldsymbol{x}^e) = \begin{bmatrix} \mathcal{C} & \mathcal{D} \\ \frac{\partial \mathcal{A}}{\partial \boldsymbol{x}_4}\mathcal{C} & \mathcal{E} \end{bmatrix} \in \mathbb{R}^{6\times 6}. \tag{4.15}$$

Linear dependence between columns of a square matrix causes its determinant to become zero. We therefore calculate the determinant of $\widetilde{\boldsymbol{\Delta}}(\boldsymbol{x}^e)$ by employing the following well-known mathematical relationship concerning the determinant of a block matrix (see e.g. Harville, 2000):

$$\det\left(\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}\right) = \det(\boldsymbol{A}) \det\left(\boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B}\right), \tag{4.16}$$

where $\boldsymbol{A}$ and $\boldsymbol{D}$ are any square matrices and $\boldsymbol{A}$ is non-singular. The determinant of $\widetilde{\boldsymbol{\Delta}}(\boldsymbol{x}^e)$ can now be calculated as

$$\det\left(\widetilde{\boldsymbol{\Delta}}(\boldsymbol{x}^e)\right) = \det(\mathcal{C}) \det\left(\mathcal{E} - \frac{\partial \mathcal{A}}{\partial \boldsymbol{x}_4}\mathcal{D}\right), \tag{4.17}$$

which is zero when $\det(\mathcal{C}) = 0$ or when $\det\left(\mathcal{E} - \frac{\partial \mathcal{A}}{\partial \boldsymbol{x}_4}\mathcal{D}\right) = 0$. This means that $\widetilde{\boldsymbol{\Delta}}(\boldsymbol{x}^e)$, and thereby also $\boldsymbol{\Delta}(\boldsymbol{x}^e)$, has full rank whenever $\det(\mathcal{C}) \neq 0$ and $\det\left(\mathcal{E} - \frac{\partial \mathcal{A}}{\partial \boldsymbol{x}_4}\mathcal{D}\right) \neq 0$. This completes the proof. ■

The requirement regarding the two determinants in Theorem 4.5 is not very restrictive, but it implies that the snake robot can attain configurations that are singular, i.e. certain shapes of the snake robot are obstructive from a control perspective since the dimension of the reachable space from these configurations is not full-dimensional. These singular configurations are revealed by the following property:

Property 4.1 *The accessibility algebra $\boldsymbol{\Delta}(\boldsymbol{x}^e)$ drops rank at equilibrium points where all relative joint angles are equal ($\phi_1 = \ldots = \phi_{N-1}$).*

This property can be shown to hold with a mathematical software tool such as *Matlab Symbolic Math Toolbox* since it can be verified that $\det(\mathcal{C})\big|_{\phi_1=\ldots=\phi_{N-1}} = 0$, thereby violating the full rank conditions stated

in Theorem 4.5. Property 4.1 is interesting since it implies that the joint angles of a snake robot should be *out of phase* during snake locomotion. This claim has been stated in several previous works, such as Gray (1946); Hicks (2003); Hirose (1993); Kane and Lecison (2000), but no formal mathematical proof was given.

We now show that the snake robot does *not* satisfy sufficient conditions for *small-time local controllability* (STLC).

Theorem 4.6 *At any equilibrium point* $\boldsymbol{x}^e$ ($\dot{\boldsymbol{q}}_\phi = \boldsymbol{0}$)*, a planar snake robot with* $N \geq 4$ *links influenced by viscous ground friction does* not *satisfy the sufficient conditions for* small-time local controllability *(STLC) stated in Theorem 4.2.*

Proof. The proof is complete if we can show that there are *bad* brackets of the system vector fields that cannot be neither θ*-neutralized* nor l*-neutralized* (see Theorem 4.2). The *bad* brackets with the lowest θ-degree and the lowest l-degree (except for $\boldsymbol{f}$, which vanishes at any equilibrium point) are $\big[\boldsymbol{g}_j, \big[\boldsymbol{f}, \boldsymbol{g}_j\big]\big]$, $j \in \{1, 2, 3\}$. Theorem 4.2 requires these vectors to be written as linear combinations of *good* brackets with either lower θ-degree or lower l-degree. The only such *good* brackets are $\boldsymbol{g}_j$, $\big[\boldsymbol{f}, \boldsymbol{g}_j\big]$, $\big[\boldsymbol{f}, \big[\boldsymbol{f}, \boldsymbol{g}_j\big]\big]$, …, $\big[\boldsymbol{f}, [\cdots \big[\boldsymbol{f}, \boldsymbol{g}_j\big]] \cdots]$, $j \in \{1, 2, 3\}$. Brackets of the form $\big[\boldsymbol{g}_k, \boldsymbol{g}_j\big]$ are not considered because $\big[\boldsymbol{g}_k, \boldsymbol{g}_j\big] = \boldsymbol{0}$, $j, k \in \{1, 2, 3\}$. For a proper choice of θ and l_j, $j \in \{0, 1, 2, 3\}$, these brackets have both lower θ-degree and lower l-degree. It is straightforward to verify that $\big[\boldsymbol{g}_j, \big[\boldsymbol{f}, \boldsymbol{g}_j\big]\big] \in \mathbb{R}^{2N+4=12}$ is a vector of all zeros except for element number $2N + 2 = 10$. The only way to write this vector as a linear combination of the *good* brackets listed above is if these *good* brackets span the entire 12-dimensional state space. This is not the case, however, because the vectors $\big[\boldsymbol{f}, \big[\boldsymbol{f}, \boldsymbol{g}_j\big]\big]$, …, $\big[\boldsymbol{f}, [\cdots \big[\boldsymbol{f}, \boldsymbol{g}_j\big]] \cdots]$, $j \in \{1, 2, 3\}$, are linearly dependent, as can be seen by assembling the matrix

$$\begin{aligned}
&\big[\big[\boldsymbol{f}, \big[\boldsymbol{f}, \boldsymbol{g}_j\big]\big], \big[\boldsymbol{f}, \big[\boldsymbol{f}, \big[\boldsymbol{f}, \boldsymbol{g}_j\big]\big]\big], \big[\boldsymbol{f}, \big[\boldsymbol{f}, \big[\boldsymbol{f}, \big[\boldsymbol{f}, \boldsymbol{g}_j\big]\big]\big]\big], \cdots\big] \\
&= \begin{bmatrix}
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \cdots \\
\mathcal{C} & -\frac{\partial \mathcal{A}}{\partial \boldsymbol{x}_4}\mathcal{C} & \left(\frac{\partial \mathcal{A}}{\partial \boldsymbol{x}_4}\right)^2 \mathcal{C} & \cdots \\
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \cdots \\
\frac{\partial \mathcal{A}}{\partial \boldsymbol{x}_4}\mathcal{C} & -\left(\frac{\partial \mathcal{A}}{\partial \boldsymbol{x}_4}\right)^2 \mathcal{C} & \left(\frac{\partial \mathcal{A}}{\partial \boldsymbol{x}_4}\right)^3 \mathcal{C} & \cdots
\end{bmatrix},
\end{aligned} \tag{4.18}$$

and noting that the fourth row is a multiple of the second row. It is therefore not possible to either θ*-neutralize* nor l*-neutralize* the *bad* brackets of the system in (2.46). The linear dependence in (4.18) is also present for $N > 4$

links since the six non-zero rows of (4.18) concern the position and head angle of the snake while the $2N-2$ remaining rows will be zero regardless of N. This completes the proof. ■

Note that Theorem 4.6 does not claim that the snake robot is not STLC. In other words, the snake robot may be STLC even though the sufficient conditions of Theorem 4.2 are violated. Note also that STLC is not a requirement for controllability since, as described in Section 4.1, it is in fact a stronger property than controllability. In summary, the above results do not enable us to conclude that a wheel-less snake robot influenced by anisotropic viscous ground friction is controllable. However, the above results are hopefully an important step towards formally proving that such mechanisms are controllable, which we consider highly likely to be the case.

We end this section with a note on Theorem 4.5. This theorem clearly shows that *anisotropic* friction is an important property for a snake robot. In the snake robot literature, it is common for snake robots to exhibit the property $c_n \gg c_t$. The extreme case of this property is realized by installing passive wheels (that cannot slip sideways) along the snake body since this ideally means that $c_t = 0$ and $c_n = \infty$. However, from Theorem 4.5 it is clear that the only requirement for *strong accessibility* is that the friction coefficients are *not equal*. The property $c_t > c_n$ is therefore also valid. This means that the passive wheels commonly mounted tangential to the snake body may equally well be mounted transversal to the snake body. The resulting motion will of course be different, but the *strong accessibility* property is still preserved.

4.4 Analysis of Propulsive Forces during Snake Locomotion

Having derived some fundamental properties of snake robot locomotion in Section 4.2 and Section 4.3, we now turn our attention to the following question: How and why does anisotropic viscous ground friction enable snake robots to locomote forward on a planar surface? We will answer this question simply by investigating the equations of motion of the snake robot.

We begin by deriving an expression for the total force propelling the CM of the snake robot forward as a function of the linear link velocities. We call this the *propulsive force* on the robot and denote it by $F_{\text{prop}} \in \mathbb{R}$. The forward direction of motion is assumed to be along the global positive x axis, which means that the propulsive force is simply the sum of all external forces on the snake robot in the global x direction. From (2.32b) and (2.24),

we can calculate F_{prop} as

$$F_{\text{prop}} = Nm\ddot{p}_x = \boldsymbol{e}^T \boldsymbol{f}_{R,x} = -\boldsymbol{e}^T \Big(\big(c_t (\boldsymbol{C}_\theta)^2 + c_n (\boldsymbol{S}_\theta)^2 \big) \dot{\boldsymbol{X}} + (c_t - c_n) \boldsymbol{S}_\theta \boldsymbol{C}_\theta \dot{\boldsymbol{Y}} \Big). \tag{4.19}$$

The result of multiplying the vector $\boldsymbol{e}^T = [1, \ldots, 1] \in \mathbb{R}^N$ with $\boldsymbol{f}_{R,x}$ in (4.19) is an addition of all elements in $\boldsymbol{f}_{R,x}$, which means that (4.19) can be written as

$$F_{\text{prop}} = -\sum_{i=1}^{N} \big(\big(c_t \cos^2 \theta_i + c_n \sin^2 \theta_i \big) \dot{x}_i + (c_t - c_n) \sin \theta_i \cos \theta_i \dot{y}_i \big). \tag{4.20}$$

The propulsive force contribution from a single link, $F_{\text{prop},i} \in \mathbb{R}$, is in other words given by

$$F_{\text{prop},i} = -F_x(\theta_i) \dot{x}_i - F_y(\theta_i) \dot{y}_i, \tag{4.21}$$

where

$$F_x(\theta_i) = c_t \cos^2 \theta_i + c_n \sin^2 \theta_i, \tag{4.22}$$

$$F_y(\theta_i) = (c_t - c_n) \sin \theta_i \cos \theta_i, \tag{4.23}$$

and where we recall from Section 2.3 that the angle θ_i of link i is expressed with respect to the global x axis with counterclockwise positive direction. We see from (4.21) that $F_{\text{prop},i}$ consists of two components, i.e. one involving the linear velocity of the link in the *forward* direction of motion, $F_x(\theta_i) \dot{x}_i$, and one involving the linear velocity *normal* to the direction of motion, $F_y(\theta_i) \dot{y}_i$. Due to the minus signs in (4.21), the products $F_x(\theta_i) \dot{x}_i$ and $F_y(\theta_i) \dot{y}_i$ provide a positive contribution to the propulsive force only if they are *negative*. Since the viscous friction coefficients, c_t and c_n, are always positive, the expression $F_x(\theta_i)$ is obviously always positive. We assume that the snake robot is not generating waves that involve x direction velocities of any of the links opposite to the direction of motion. When the snake robot is moving in the forward direction ($\dot{p}_x > 0$), we therefore have that $\dot{x}_i > 0$, which means that the product $F_x(\theta_i) \dot{x}_i$ of the propulsive force is always positive. This product is therefore *not* contributing to the forward propulsion of the robot, but rather opposing it. This is also expected since the snake robot must naturally be subjected to a friction force opposite to the direction of motion.

Any maintained propulsive force in the forward direction of motion must therefore be produced by the *sideways* motion of the links, i.e. the product $F_y(\theta_i) \dot{y}_i$. A plot of $F_y(\theta_i)$ for different values of the normal friction coefficient c_n, while keeping the tangential friction coefficient c_t fixed, is shown in

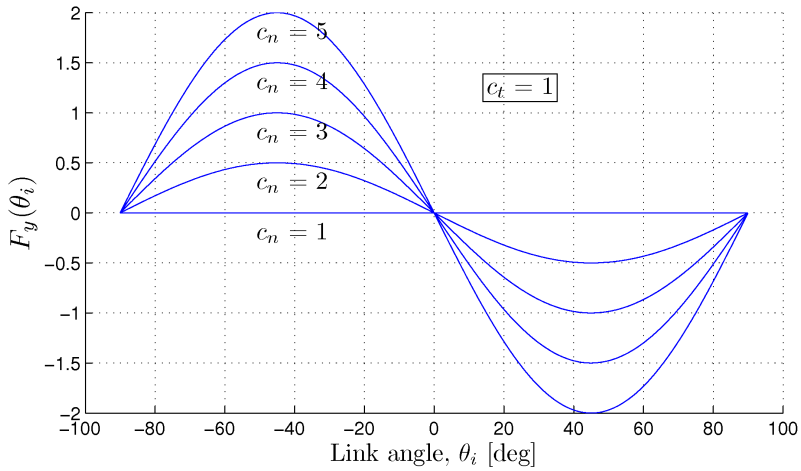


Figure 4.1: The mapping from sideways link motion to forward propulsion for different viscous friction coefficients.

Fig. 4.1. For each plot, the angle between the link and the forward direction, θ_i, is varied from $-90°$ to $90°$. The sideways motion of a link has no effect on the propulsive force on the snake robot when the friction coefficients are equal since this gives $F_y(\theta_i) = 0$. However, when $c_n > c_t$, Fig. (4.1) reveals that $F_y(\theta_i)$ is negative as long as θ_i is positive, and vice versa. This means that the product $F_y(\theta_i)\,\dot{y}_i$ is negative (the sideways motion of link i contributes to the propulsion) as long as $\text{sgn}(\theta_i) = \text{sgn}(\dot{y}_i)$. The sideways motion of link i is in other words contributing to the propulsion of the snake robot as long as θ_i is *positive during leftward motion of the link* (left with respect to the direction of motion) and *negative during rightward motion of the link* (right with respect to the direction of motion). This fundamental relationship may be written $\text{sgn}(F_{\text{prop},i}) = \text{sgn}(\text{sgn}(\theta_i)\,\text{sgn}(\dot{y}_i))$.

It is straightforward to calculate that the extrema of $F_y(\theta_i)$ occur at $\theta_i = \pm 45°$. This is also seen from Fig. 4.1. This means that, for a given $\dot{y}_i$, a link produces its highest propulsive force when it forms an angle of $\pm 45°$ with the forward direction of motion. It is also evident from (4.21) that the magnitude of $F_y(\theta_i)\,\dot{y}_i$, and thereby the magnitude of the propulsive force, $|F_{\text{prop},i}|$, is increased by increasing c_n with respect to c_t, or by increasing the magnitude of the sideways link velocity, $|\dot{y}_i|$.

It should now be clear that the function $F_y(\theta_i)$ maps the link velocities normal to the direction of motion into force components in the direction

of motion. The following simple analogy may help understand this result. Imagine a small, hand-held, wheeled wagon of some sort. The direction of the wheels corresponds to the tangential direction of a snake robot link. Obviously, the friction coefficient of the wagon in the direction of the wheels is smaller than the friction coefficient normal to the wheels. Now assume that you push the wagon across a table in the direction of the wheels. While maintaining constant direction of motion, assume that you slowly rotate the wagon about the vertical axis, thereby forcing the wheels to slip. The hand that push and rotate the wagon will now feel a tendency of the wagon to move sideways in the same direction towards which the wagon was rotated. This is in accordance with the results presented above.

The above analysis is summarized by the following properties of planar snake robot locomotion under viscous friction conditions:

Property 4.2 *For a snake robot described by (2.34) with $c_n > c_t$, forward propulsion is produced by the link velocity components that are transversal to the forward direction. The function $F_y(\theta_i)$ maps the link velocities transversal to the direction of motion into force components in the direction of motion.*

Property 4.3 *For a snake robot described by (2.34) with $c_n > c_t$, the propulsive force generated by the transversal motion of link i is positive as long as $\operatorname{sgn}(\theta_i) = \operatorname{sgn}(\dot{y}_i)$.*

Property 4.4 *For a snake robot described by (2.34) with $c_n > c_t$, the magnitude of the propulsive force produced by link i, $|F_{\text{prop},i}|$, is increased by increasing c_n with respect to c_t, or by increasing the magnitude of the sideways link velocity, $|\dot{y}_i|$, or by increasing $|\theta_i|$ as long as $|\theta_i| < 45°$.*

Property 4.5 *For a given $\dot{y}_i$, a link produces its highest propulsive force when it forms an angle of $\theta_i = \pm 45°$ with the forward direction of motion.*

4.5 Synthesis of Propulsive Motion for the Snake Robot

The results presented so far in this chapter are general in the sense that no assumptions have yet been made regarding the actual motion pattern displayed by the snake robot. In this chapter, we use these results as a basis for deducing how the links of a snake robot can be moved in order to propel the robot forward along the global x axis. In order to deduce the

propulsive link motion, we focus on manipulating the magnitude, $|F_{\text{prop},i}|$, and direction, $\text{sgn}\,(F_{\text{prop},i})$, of the propulsive force from each link.

Theorem 4.3 in Section 4.2 suggests that the angle of each link should be time-varying. Furthermore, Property 4.2 shows that propulsive forces are generated by moving the links *transversally* to the desired direction of motion. We therefore conclude that the links must have a *periodic* velocity component normal to the direction of motion. This suggests that each link should be moved alternatingly to the left and right with respect to the direction of motion, which can be achieved by letting the trajectory of each link angle have the form

$$\theta_i = \alpha \sin(\omega t), \tag{4.24}$$

where $i \in \{1, \ldots, N\}$, $\alpha > 0$ is the amplitude of the link angles during the locomotion, $\omega > 0$ is the angular frequency of the periodic motion, and t denotes time. For simplicity, we assume that α and ω are constant and identical for all links.

In accordance with Property 4.1 in Section 4.3.2, the joint angles should be *out of phase* during snake locomotion since this improves the controllability properties of the robot. This suggests that (4.24) should be modified to

$$\theta_i = \alpha \sin(\omega t + (i-1)\,\delta), \tag{4.25}$$

where δ is the phase shift between adjacent links, which, for simplicity, is assumed to be constant.

We now investigate how α, ω, and δ affect $|F_{\text{prop},i}|$ and $\text{sgn}\,(F_{\text{prop},i})$ as the snake robot moves along the global x axis. To simplify the analysis, we assume that the snake robot consists of only $N = 3$ links. This is the minimum number of links required to achieve propulsion since phase shift between joints requires at least two joints. The below analysis for $N = 3$ links also apply to robots with $N > 3$ links since a snake robot can be regarded as a connection of multiple three-linked segments. The link angle trajectories are given from (4.25) as

$$\theta_1 = \alpha \sin(\omega t), \tag{4.26a}$$
$$\theta_2 = \alpha \sin(\omega t + \delta), \tag{4.26b}$$
$$\theta_3 = \alpha \sin(\omega t + 2\delta), \tag{4.26c}$$

which, when differentiated with respect to time, gives the angular link

velocities

$$\dot{\theta}_1 = \alpha\omega\cos(\omega t), \tag{4.27a}$$
$$\dot{\theta}_2 = \alpha\omega\cos(\omega t + \delta), \tag{4.27b}$$
$$\dot{\theta}_3 = \alpha\omega\cos(\omega t + 2\delta). \tag{4.27c}$$

The normal direction velocity of each link is given by (2.13b). We disregard the normal direction velocity of the snake robot by setting $\dot{p}_y \approx 0$. This approximation is a fairly accurate during motion along the global x axis, which is the case for this analysis. Inserting (4.26) and (4.27) into (2.13b) gives

$$\begin{aligned}\dot{y}_1 = &-\tfrac{\alpha\omega l}{3}\left(2\cos(\omega t)\cos(\alpha\sin(\omega t))\right)\\ &-\tfrac{\alpha\omega l}{3}\left(3\cos(\omega t+\delta)\cos(\alpha\sin(\omega t+\delta))\right)\\ &-\tfrac{\alpha\omega l}{3}\left(\cos(\omega t+2\delta)\cos(\alpha\sin(\omega t+2\delta))\right),\end{aligned} \tag{4.28a}$$

$$\begin{aligned}\dot{y}_2 = &\tfrac{\alpha\omega l}{3}\left(\cos(\omega t)\cos(\alpha\sin(\omega t))\right)\\ &-\tfrac{\alpha\omega l}{3}\left(\cos(\omega t+2\delta)\cos(\alpha\sin(\omega t+2\delta))\right),\end{aligned} \tag{4.28b}$$

$$\begin{aligned}\dot{y}_3 = &\tfrac{\alpha\omega l}{3}\left(\cos(\omega t)\cos(\alpha\sin(\omega t))\right)\\ &+\tfrac{\alpha\omega l}{3}\left(3\cos(\omega t+\delta)\cos(\alpha\sin(\omega t+\delta))\right)\\ &+\tfrac{\alpha\omega l}{3}\left(2\cos(\omega t+2\delta)\cos(\alpha\sin(\omega t+2\delta))\right).\end{aligned} \tag{4.28c}$$

Property 4.4 tells us that $|F_{\text{prop},i}|$ is increased by increasing $|\dot{y}_i|$. From (4.28a)-(4.28c), it is therefore clear that $|F_{\text{prop},i}|$ is increased by increasing α and/or ω. We now determine if δ should be positive or negative in order to achieve $\text{sgn}(F_{\text{prop},i}) = 1$, which is necessary to propel the snake robot forward along the global x axis. From Property 4.3, we know that $\text{sgn}(F_{\text{prop},i}) = 1$ requires $\text{sgn}(\theta_i) = \text{sgn}(\dot{y}_i)$. Considering $\dot{y}_2$ in (4.28b) (since this expression is easy to analyse), it is seen through pure inspection that $\dot{y}_2 = 0$ when $\omega t = -\delta$. When $\omega t = -\delta$, we see from (4.26b) and (4.27b) that $\theta_2 = 0$ and $\dot{\theta}_2 = \alpha\omega > 0$. θ_2 is in other words about to become positive, which means that we also require $\dot{y}_2$ to become positive. This is the case if $\ddot{y}_2 > 0$ when $\omega t = -\delta$. Differentiating (4.28b) with respect to time gives

$$\ddot{y}_2\Big|_{\omega t = -\delta} = \frac{2\alpha\omega^2 l}{3}\left(\alpha\cos^2(\delta)\sin(\alpha\sin(\delta)) + \sin(\delta)\cos(\alpha\sin(\delta))\right), \tag{4.29}$$

from which it is easily seen that $\ddot{y}_2 > 0$ when $\delta > 0$, i.e. $\text{sgn}(F_{\text{prop},i}) = 1$ when $\delta > 0$. This indicates that the links generate positive propulsive forces if $\delta > 0$.

In order to verify that forward propulsion requires $\delta > 0$, we have plotted (4.26) and (4.28) together in Fig. 4.2 for $\alpha = 70°$, $\omega = 70°$, and for different positive values of δ over a period of ωt from 0 to 2π. The figures show that $\text{sgn}\left(F_{\text{prop},i}\right) = \text{sgn}\left(\text{sgn}\left(\theta_i\right)\text{sgn}\left(\dot{y}_i\right)\right) = 1$ is always satisfied for link 2, but only satisfied over about half the period for link 1 and 3 when δ is small. As δ is increased, $\text{sgn}\left(F_{\text{prop},i}\right) = 1$ is satisfied over a larger portion of the period. The optimal choice of δ will be derived in Chapter 7. For now, however, we simply conclude that positive propulsive forces requires $\delta > 0$.

The following property summarizes the above analysis.

Property 4.6 *A snake robot described by (2.34) with anisotropic viscous ground friction properties ($c_n > c_t$) achieves forward propulsion by moving its links according to $\theta_i = \alpha \sin\left(\omega t + (i-1)\,\delta\right)$ where $i \in \{1, \ldots, N\}$, $\alpha > 0$, $\omega > 0$, and $\delta > 0$. Increasing α and/or ω increases the magnitude of the propulsive forces generated by the links.*

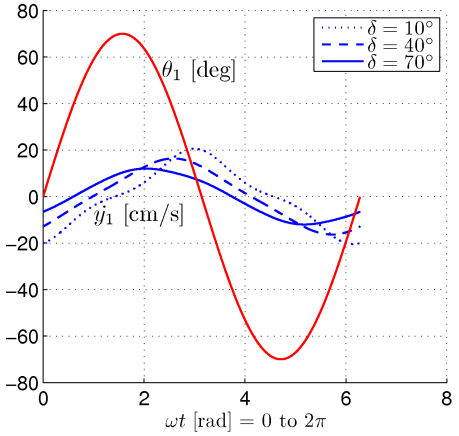
4.6 The Gait Pattern Lateral Undulation

During locomotion according to the gait pattern presented in Property 4.6, the snake robot produces continuous body waves that are propagated backwards from the head to the tail. This form of motion is called *lateral undulation* and is the most common form of locomotion displayed by biological snakes (see Section 1.2.2). As described in Section 1.4, motion by lateral undulation has a central role in the control design efforts described in this thesis since we consider this motion pattern to be most relevant and most efficient for planar snake robot locomotion.

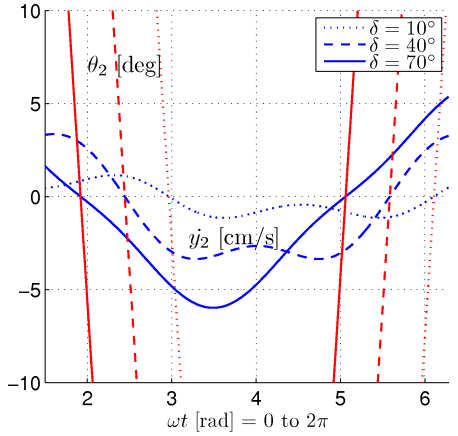
A well-known mathematical description of lateral undulation was presented in Hirose (1993) based on empirical studies of biological snakes. Hirose discovered that a close approximation to the shape of a biological snake during lateral undulation is given by a planar curve whose curvature varies sinusoidally. Hirose named this curve the *serpenoid curve* and described it by

$$x(s) = \int_0^s \cos\left(a\cos\left(b\sigma\right) + c\sigma\right) d\sigma, \quad y(s) = \int_0^s \sin\left(a\cos\left(b\sigma\right) + c\sigma\right) d\sigma, \tag{4.30}$$

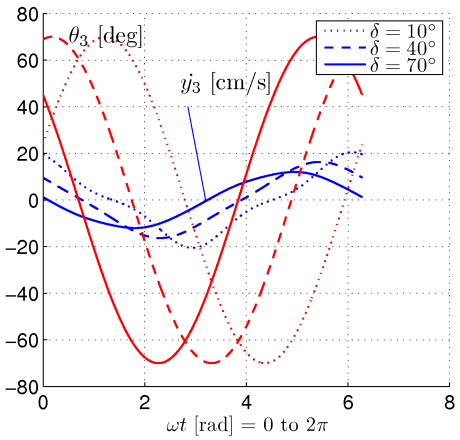
where $(x(s), y(s))$ are the coordinates of the point along the curve at arc length s from the origin, and where a, b, and c are positive scalars. The



(a) Plot of θ_1 and $\dot{y}_1$.



(b) Plot of θ_2 and $\dot{y}_2$.



(c) Plot of θ_3 and $\dot{y}_3$.

Figure 4.2: The relation between θ_i and $\dot{y}_i$, $i \in \{1, 2, 3\}$, for $\alpha = 70°$, $\omega = 70°$, and $\delta = 10°$ (dotted) , $40°$ (dashed) , $70°$ (solid).

curvature κ of the serpenoid curve varies sinusoidally according to $\kappa(s) = |ab\sin(bs) - c|$. Note that we have used the notation from Saito *et al.* (2002), which considers motion control of snake robots based on the serpenoid curve. It is shown in Saito *et al.* (2002) that a serpenoid curve of arc length 1 can be approximated by N identical discrete segments by calculating the angle θ_i of segment $i \in \{1, \ldots, N\}$ with respect to the x axis according to

$$\theta_i = a\cos\left(\frac{ib}{N}\right). \tag{4.31}$$

This implies that a snake robot with N identical discrete links attains a discrete approximation to the serpenoid curve by moving its link angles sinusoidally with a constant phase shift between the links, which means that the motion pattern that we derived in Property 4.6 is a discrete approximation to the serpenoid curve. The analysis leading up to Property 4.6 is therefore a support of the serpenoid curve motion proposed by Hirose. However, while Hirose derived the serpenoid curve based on empirical studies of biological snakes, we have in this chapter based our arguments on the mathematical properties derived from the equations of motion of the snake robot. The choices made in the analysis leading up to Property 4.6 are obviously inspired by the serpenoid curve. However, it is still interesting to see how logical arguments that support an empirically derived result can be developed through a mathematical analysis.

Since there are N link angles, but only $N-1$ control inputs (i.e. the snake robot is underactuated), it is more suitable to specify the serpenoid curve motion by the $N-1$ joint angles. Since the angle of joint $i \in \{1, \ldots, N-1\}$ is given from (2.1) as $\phi_i = \theta_i - \theta_{i+1}$, it is easy to verify that the reference motion of the joint angles will have the same form as the reference motion of the link angles specified in Property 4.6. In accordance with Hirose (1993), we also introduce a joint angle offset ϕ_{offset} in the reference motion of the joints. It will be shown in Section 4.8 that this offset can be used to control the direction of the locomotion since the offset makes the link motion asymmetrical with respect to the current heading of the robot. We now summarize the above discussion.

Definition 4.1 ***Lateral undulation.***
The gait pattern lateral undulation *is achieved by moving the joints of a planar snake robot according to*

$$\phi_{i,\text{ref}} = \alpha\sin(\omega t + (i-1)\delta) + \phi_o, \tag{4.32}$$

where $i \in \{1, \ldots, N-1\}$, α and ω are the amplitude and frequency, respectively, of the sinusoidal joint motion, δ determines the phase shift between the joints, and ϕ_o is a joint offset, which we assume to be identical for all joints.

4.7 The Control System of the Joints

In this thesis, we consider two different control laws for making the joint angles $\boldsymbol{\phi} = \begin{bmatrix}\phi_1, \ldots, \phi_{N-1}\end{bmatrix}^T \in \mathbb{R}^{N-1}$ track the joint reference angles given by $\boldsymbol{\phi}_{\text{ref}} = \begin{bmatrix}\phi_{1,\text{ref}}, \ldots, \phi_{N-1,\text{ref}}\end{bmatrix}^T \in \mathbb{R}^{N-1}$. We assume that the control input $\boldsymbol{u} \in \mathbb{R}^{N-1}$ of the model (2.34) is set according to the linearizing control law (2.43) so that the joint dynamics of the snake robot is given by $\ddot{\boldsymbol{\phi}} = \overline{\boldsymbol{u}}$. Both control laws are therefore defined in terms of the linearized control input $\overline{\boldsymbol{u}} \in \mathbb{R}^{N-1}$.

4.7.1 A Simple Joint Controller

The first control law, which we refer to as the *simple joint controller*, combines proportional action with velocity damping as follows:

$$\overline{\boldsymbol{u}} = k_p \left(\boldsymbol{\phi}_{\text{ref}} - \boldsymbol{\phi}\right) - k_d \dot{\boldsymbol{\phi}}, \tag{4.33}$$

where $k_p > 0$ and $k_d > 0$ are controller gains, and where $\dot{\boldsymbol{\phi}}_{\text{ref}} = \boldsymbol{0}$ since the purpose of the derivative part is simply to damp the joint motion if the joint velocities become large. The advantage of this controller is that it does not require calculation of the derivative of $\boldsymbol{\phi}_{\text{ref}}$ with respect to time, which e.g. allows $\boldsymbol{\phi}_{\text{ref}}$ to be discontinuous. The disadvantage of the controller, however, is that its stability properties are not as easy to establish as the stability properties of the second control law.

4.7.2 An Exponentially Stable Joint Controller

The second control law, which we refer to as the *exponentially stable joint controller*, is defined as

$$\overline{\boldsymbol{u}} = \ddot{\boldsymbol{\phi}}_{\text{ref}} + k_d \left(\dot{\boldsymbol{\phi}}_{\text{ref}} - \dot{\boldsymbol{\phi}}\right) + k_p \left(\boldsymbol{\phi}_{\text{ref}} - \boldsymbol{\phi}\right), \tag{4.34}$$

where $k_p > 0$ and $k_d > 0$ are scalar controller gains. Since the joint dynamics is given by $\ddot{\boldsymbol{\phi}} = \overline{\boldsymbol{u}}$, the resulting error dynamics of the joints is given by

$$\left(\ddot{\boldsymbol{\phi}}_{\text{ref}} - \ddot{\boldsymbol{\phi}}\right) + k_d \left(\dot{\boldsymbol{\phi}}_{\text{ref}} - \dot{\boldsymbol{\phi}}\right) + k_p \left(\boldsymbol{\phi}_{\text{ref}} - \boldsymbol{\phi}\right) = \boldsymbol{0}, \tag{4.35}$$

which is clearly *exponentially stable* (see Khalil, 2002). During lateral undulation according to (4.32), we can easily calculate $\dot{\boldsymbol{\phi}}_{\text{ref}}$ and $\ddot{\boldsymbol{\phi}}_{\text{ref}}$ if ϕ_o is assumed to be a constant offset since this gives

$$\dot{\phi}_{i,\text{ref}} = \alpha\omega \cos\left(\omega t + (i-1)\,\delta\right), \tag{4.36a}$$

$$\ddot{\phi}_{i,\text{ref}} = -\alpha\omega^2 \sin\left(\omega t + (i-1)\,\delta\right). \tag{4.36b}$$

If, on the other hand, ϕ_o is a complex function of time so that $\dot{\boldsymbol{\phi}}_{\text{ref}}$ and $\ddot{\boldsymbol{\phi}}_{\text{ref}}$ cannot easily be calculated analytically, then $\dot{\boldsymbol{\phi}}_{\text{ref}}$ and $\ddot{\boldsymbol{\phi}}_{\text{ref}}$ can be obtained by passing the commanded reference angles, $\boldsymbol{\phi}_{\text{ref}}$, through a 3rd order low-pass filtering reference model, as described in Appendix C.2.

4.8 Analysis of Turning Motion during Lateral Undulation

In this section, we investigate how the joint offset ϕ_o in (4.32) affects the heading of the snake robot during lateral undulation. To this end, the model of the snake robot in (2.46) was implemented in *Matlab R2008b*, where the dynamics of the model was calculated by use of the *ode45* solver with a relative and absolute error tolerance of 10^{-6}.

We considered a snake robot with $N = 10$ links of length $2l = 0.14$ m, mass $m = 1$ kg, and moment of inertia $J = 0.0016$ kgm^2. The ground friction forces were assumed to be of the viscous type defined in (2.24) with friction coefficients $c_t = 1$ and $c_n = 10$. The simple joint controller in (4.33) was used to control the joints of the robot according to (4.32) with gait pattern parameters $\alpha = 30°$, $\omega = 70°/_s$, and $\delta = 40°$. In order to study the effect of ϕ_o on the motion, the offset angle was set to $\phi_o = 5°$ in the time interval $t \in [20, 30]$ and $\phi_o = -10°$ in the time interval $t \in [50, 60]$. The offset angle was set to $\phi_o = 0°$ outside these two time intervals.

The simulation result is shown in Fig. 4.3. The top of Fig. 4.3 shows the trace of the head during the motion, while the bottom of the figure shows the average joint angle, which we denote by $\overline{\phi} = \frac{1}{N-1}\sum_{i=1}^{N-1}\phi_i$. The triangles pointing up and down in the top of Fig. 4.3 indicate, respectively, the beginning and end of the two time intervals where ϕ_o is non-zero. Fig. 4.3 shows that the snake robot crawls forward without turning as long as the average joint angle, $\overline{\phi}$, is zero. However, when the average joint angle is non-zero, the direction of the motion changes. We see from the figure that a positive (resp. negative) average joint angle produces a counterclockwise (resp. clockwise) rotation of the snake robot. We also see that the speed

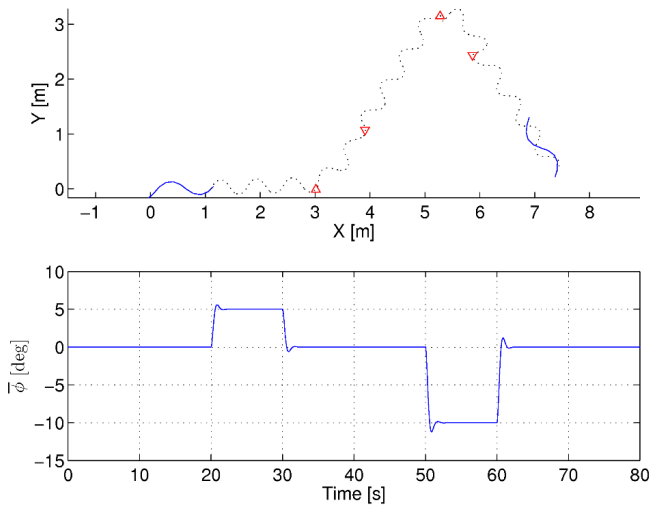


Figure 4.3: Simulated motion of a snake robot with $N = 10$ links. A joint offset angle of $\phi_o = 5°$ and $\phi_o = -10°$ is introduced at $t = 20$ s and $t = 50$ s, respectively. Top: Trace of the head of the snake robot. Bottom: The average joint angle.

of the directional change is correlated with the amplitude of the average joint angle. This result is supported by the directional controllers for snake locomotion considered in e.g. Saito *et al.* (2002); Sfakiotakis and Tsakiris (2007). The following property summarizes this analysis:

Property 4.7 *During* lateral undulation *with a snake robot described by (2.46) with $c_n > c_t$, the overall direction of the locomotion will remain constant as long as the average joint angle is zero, but will change in the counterclockwise (resp. clockwise) direction when the average joint angle is positive (resp. negative). The rate of directional change increases when the amplitude of the average joint angle increases.*

Remark 4.2 *A formal proof that ϕ_o affects the motion direction of the snake robot still remains. One possible approach is to prove this property by investigating Lie brackets of the system vector fields (see Section 4.1). Such an approach is employed in e.g. Morgansen* et al. *(2001) to study the motion of robotic fish. A challenge in the present work, however, is that a snake robot with revolute joints has no explicitly defined orientation since there is an independent link angle associated with each link. An estimate of*

the heading of the robot was given in Definition 2.3 as the average of the link angles. The summed dynamics of all the links is, unfortunately, extremely complex, which makes it difficult to analyse the Lie bracket motion of the heading.

4.9 Analysis of Relative Motion between Consecutive Links during Lateral Undulation

From the results presented so far in this chapter, it should be clear that planar snake robot locomotion consists of periodic body shape changes that generate external forces that propel the robot forward. Since body shape changes is equivalent to displacing the links relative to each other, we can always partition the body shape changes into relative link displacements that are, respectively, *tangential* and *transversal* to the forward direction of the robot. We know from Property 4.2 that the transversal link displacements are what propel the snake robot forward.

With respect to the gait pattern *lateral undulation* presented in Definition 4.1, it seems natural to ask how large part of the body shape changes during lateral undulation that constitute tangential and transversal link displacements, respectively. To answer this question, we employed the simulation setup presented in Section 4.8 to simulate a snake robot with $N = 10$ links during lateral undulation along the global x axis with $\alpha = 30°$, $\omega = 30°/s$, $\delta = 40°$, and $\phi_o = 0°$. The trace of the head of the snake robot is plotted in the top of Fig. 4.4, while the two bottom plots in Fig. 4.4 show the relative displacements between the CM of two arbitrarily chosen links (link 4 and link 5) in the global x and y direction, respectively. The plots indicate that, during lateral undulation, the relative displacements between the CM of two adjacent links along the forward direction of motion are approximately constant, while the relative displacements normal to the direction of motion oscillate around zero. This observation is formalized as follows:

Property 4.8 *The change in body shape during lateral undulation consists mainly of relative displacements of the CM of the links* normal *to the forward direction of motion. The relative displacements of the CM of the links along the forward direction are approximately constant.*

Remark 4.3 *Property 4.8 and Property 4.2 constitute the basis for the simplified model of the snake robot presented in Chapter 6. The properties*

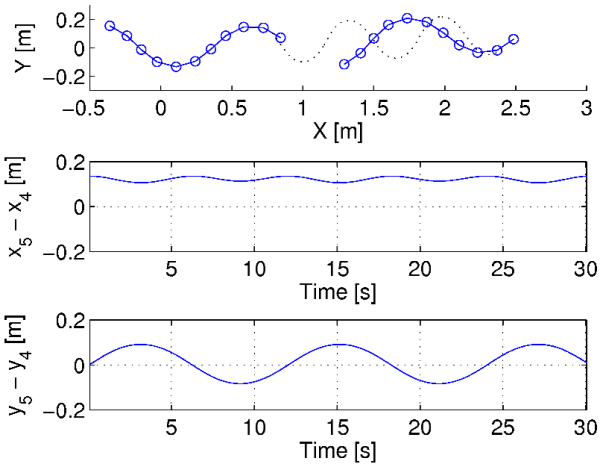


Figure 4.4: Top: Simulated motion of a snake robot with $N = 10$ links. Middle: Relative displacements between the CM of link 4 and link 5 in the global x direction. Bottom: Relative displacements between the CM of link 4 and link 5 in the global y direction.

tell us two things, namely 1) that lateral undulation mainly consists of link displacements that are transversal to the direction of motion, and 2) that the transversal link displacements are what propel the robot forward. These results are appealing since they suggest that snake robot locomotion can be described in terms of the transversal displacements of the links instead of the more complex rotational link motion. We elaborate this approach in Chapter 6.

4.10 Chapter Summary

This chapter is summarized as follows:

- We have presented a stabilizability analysis that proves that any asymptotically stabilizing control law for a planar snake robot to an equilibrium point must be *time-varying*, i.e. not of pure-state feedback type (see Theorem 4.3).

- We have presented a controllability analysis of planar snake robots influenced by viscous ground friction forces that proves that:

  - A snake robot is *not* controllable when the viscous ground friction is *isotropic* (see Theorem 4.4).
  - A snake robot is *strongly accessible* from any equilibrium point (except for certain singular configurations) when the viscous ground friction is *anisotropic* (see Theorem 4.5).
  - A snake robot does *not* satisfy sufficient conditions for *small-time local controllability* (see Theorem 4.6).
  - The joint angles of a snake robot should be *out of phase* during snake locomotion (implied by Property 4.1).

- We have identified a simple relationship between link velocities *normal* to the direction of motion and resulting *propulsive forces* in the direction of motion that:
  - revealed several fundamental properties of snake robot locomotion (see Properties 4.2, 4.3, and 4.4).
  - explains how anisotropic ground friction enables snake robots to locomote forward on a planar surface.

- We have employed the derived properties to synthesise how the links of a snake robot can be moved in order to propel the robot forward (see Property 4.6).

- We have identified the synthesised motion as the gait pattern *lateral undulation* (see Definition 4.1), which is the most common form of biological snake locomotion.

- We have explained how offsetting the joint angles during lateral undulation enables directional control of the motion (see Property 4.7).

- We have shown that the body shape changes during lateral undulation mainly consist of relative displacements of the links *normal* to the forward direction of motion, while the relative displacements of the links along the forward direction are approximately constant (see Property 4.8).

Chapter 5

Path Following Control and Analysis of Snake Robots based on the Poincaré Map

In Chapter 4, we derived the gait pattern *lateral undulation*, which enables planar snake robots with anisotropic ground friction properties to locomote forward on a planar surface. We now turn to the problem of controlling the heading and position of the snake robot, and in particular, we consider the problem of enabling the robot to track a straight path. Straight line path following capabilities are important since they enable a snake robot to follow a desired path given by waypoints interconnected by straight lines. Straight line path following is therefore relevant for many future applications of snake robots, such as automated inspection rounds in inaccessible areas of industrial process facilities or mapping of confined spaces by moving along prescribed paths.

Control design for snake robots is challenging since these mechanisms are underactuated. In particular, the model of the snake robot in (2.46) contains $N+2$ degrees of freedom, but only $N-1$ control inputs. The underactuated degrees of freedom, i.e. the heading and position of the robot, make it impossible to independently control all degrees of freedom of the robot. During path following control of snake robots, there is additionally the challenge that the CM position $\boldsymbol{p}$ does not trace out a straight path during forward locomotion, but rather oscillates periodically about the straight line pointing in the forward direction of the robot. Moreover, we also expect the heading $\overline{\theta}$ of the robot, which was defined in (2.2), to oscillate periodically during forward locomotion since we cannot generally assume

that the average of the link angles is always zero. As long as the heading and position of the robot display such oscillating behaviour, it makes no sense to attempt to control these states to stationary values with respect to the desired straight path (which we normally would during path following control of more conventional propulsion mechanisms, such as marine vessels and wheeled robots).

With these challenges in mind, it becomes clear that we need a mathematical tool which allows us to study the periodically oscillating behaviour of the system states. We find such a tool in the theory of *Poincaré maps*. The Poincaré map represents a widely used tool for analysing the existence and stability of periodic orbits of dynamical systems. In this chapter, we first propose a path following controller for planar snake robots, and subsequently we analyse the stability of the locomotion along the path by use of a Poincaré map. In particular, we show that all state variables of the snake robot, except for the position along the path, trace out an *exponentially stable* periodic orbit during path following with the proposed controller. We also present simulation results that illustrate the performance of the controller. Note that the path following controller considered in this chapter is extended in Chapter 8, where we employ cascaded systems theory to investigate the convergence of the snake robot to the desired path based on the simplified model presented in Chapter 6. The path following controller in Chapter 8 circumvents the oscillating behaviour of the snake robot through a coordinate transformation combined with feedback cancellation of the oscillating dynamics.

Contributions of this Chapter: The contribution of this chapter is a control law that enables snake robots to track a straight path. To our best knowledge, a *Poincaré map* has never before been used to study the stability properties of snake robot locomotion. We therefore also consider the methodology of this chapter to be a contribution within the snake robot literature.

Organization of this Chapter: Section 5.1 gives a brief presentation of the Poincaré map. The path following controller is presented in Section 5.2, and the stability of this controller is investigated in Section 5.3 by use of a Poincaré map. Simulation results are presented in Section 5.4. Finally, the chapter is summarized in Section 5.5.

Publications: The material in this chapter is based on the journal paper Liljebäck *et al.* (2011*b*) and the conference paper Liljebäck *et al.* (2009*c*).

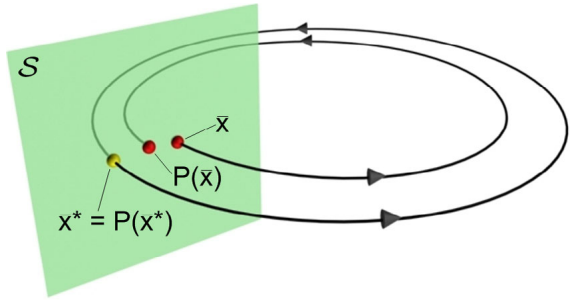


Figure 5.1: Illustration of the Poincaré map corresponding to a Poincaré section $\mathcal{S}$.

5.1 Introduction to Poincaré Maps

In this section, we give an introduction to the Poincaré map since this is used as a stability analysis tool in Section 5.3. For further details on the topic, the reader is referred to Parker and Chua (1989) and Westervelt *et al.* (2007).

5.1.1 General Description of Poincaré Maps

The Poincaré map represents a widely used tool for analysing the existence and stability of periodic orbits of dynamical systems. Consider an autonomous (not explicitly dependent on time) n-dimensional dynamical system of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^n, \tag{5.1}$$

where $\boldsymbol{f}(\boldsymbol{x})$ is assumed to be *continuously differentiable*. Assume that the solution of this differential equation for a particular initial condition is a *limit cycle*. Then the flow of $\boldsymbol{x}$ in the n-dimensional state space will return to the initial condition after a time T, corresponding to the period of the limit cycle.

We now define an $(n-1)$-dimensional hyperplane $\mathcal{S}$ (called a *Poincaré section*) such that the limit cycle intersects and passes through $\mathcal{S}$ at some instant in time. We denote by $\overline{\boldsymbol{x}} \in \mathbb{R}^{n-1}$ the $(n-1)$-dimensional state vector when $\boldsymbol{x}$ is constrained to $\mathcal{S}$. The point on $\mathcal{S}$ where the limit cycle intersects $\mathcal{S}$ is denoted $\overline{\boldsymbol{x}}^* \in \mathbb{R}^{n-1}$. Assume now that we initialize (5.1) on the hyperplane $\mathcal{S}$ somewhere close to $\overline{\boldsymbol{x}}^*$. Due to the continuity of the solutions of (5.1) with respect to the initial condition, the flow of $\boldsymbol{x}$ will,

in approximately T seconds, return to and intersect S somewhere close to $\overline{\boldsymbol{x}}^*$. This is illustrated in Fig. 5.1. The mapping from an initial point $\overline{\boldsymbol{x}}$ on $\mathcal{S}$ to the next point where the flow of $\boldsymbol{x}$ intersects $\mathcal{S}$ is called the *Poincaré map* and is denoted by $\boldsymbol{P}(\overline{\boldsymbol{x}}) \in \mathbb{R}^{n-1}$. The Poincaré map is in other words a function that accepts an initial point on a Poincaré section as input and outputs where the Poincaré section will be intersected next by the flow of $\boldsymbol{x}$. This is written more formally as $\boldsymbol{P} : \mathcal{S} \to \mathcal{S}$. The point $\overline{\boldsymbol{x}}^*$ is called a *fixed point* of the Poincaré map since the Poincaré map maps $\overline{\boldsymbol{x}}^*$ back to itself. This is also illustrated in Fig. 5.1. We only consider *one-sided* Poincaré maps, i.e. we only consider crossings of $\mathcal{S}$ in directions corresponding to the direction of $\dot{\boldsymbol{x}}$ when $\boldsymbol{x}$ initially left $\mathcal{S}$.

The Poincaré map can be interpreted as a *discrete-time* system with an $(n-1)$-dimensional state space that evolves on the Poincaré section. This is seen by denoting by $\overline{\boldsymbol{x}}[k] \in \mathcal{S}$ the point of the k-th intersection with $\mathcal{S}$ by the flow of $\boldsymbol{x}$. The Poincaré map may then be written as

$$\overline{\boldsymbol{x}}[k+1] = \boldsymbol{P}(\overline{\boldsymbol{x}}[k]), \quad \overline{\boldsymbol{x}}[0] \in \mathcal{S}. \tag{5.2}$$

The usefulness of the Poincaré map for stability analysis lies in the fact that *local exponential stability* of the fixed point $\overline{\boldsymbol{x}}^*$ on the Poincaré section is equivalent to *local exponential stability* of the underlying periodic orbit (see e.g. Westervelt *et al.*, 2007), i.e. nearby orbits converge exponentially to the periodic orbit. Note that the stability is only *asymptotic* (i.e. not *exponential*) if $\boldsymbol{f}(\boldsymbol{x})$ in (5.1) is *continuous* but not *continuously differentiable* (Westervelt *et al.*, 2007). The problem of determining if a periodic orbit of the system (5.1) is exponentially stable is, in other words, reduced to determining if $\overline{\boldsymbol{x}}^*$ is an exponentially stable equilibrium point of the *discrete-time* system in (5.2), which is a much simpler problem to solve. A significant drawback of Poincaré maps is that they provide little insight into properties of the system dynamics.

Note that the method of Poincaré maps may also be applied to *non-autonomous* periodic systems, i.e. systems of the form $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, t)$, by incapsulating the time t in an augmented periodic state variable $\beta = 2\pi t/T$. This is performed for the snake robot in Section 5.3.1.

5.1.2 Practical Application of Poincaré Maps

This section provides an informal description of the practical use of Poincaré maps. The aim is to show how this method can be employed in practice in order to investigate the stability properties of a time-periodic dynamical system.

Calculating the Poincaré Map

It is difficult to determine the Poincaré map analytically since it requires the solution of the differential equation (5.1). However, the Poincaré map of (5.1) is simply the forward integration of this differential equation. It is therefore possible to compute the Poincaré map $\boldsymbol{P}\left(\overline{\boldsymbol{x}}_0\right)$ numerically by initializing (5.1) on $\mathcal{S}$ at $\overline{\boldsymbol{x}}_0$ and simulating (5.1) until $\mathcal{S}$ is intersected. The state corresponding to this intersection is the Poincaré map $\boldsymbol{P}\left(\overline{\boldsymbol{x}}_0\right)$.

Locating Fixed Points of the Poincaré Map

The easiest way of locating a fixed point $\overline{\boldsymbol{x}}^*$ of the Poincaré map is to simply let the simulation of (5.1) run until it reaches the steady state. This is called the *brute-force approach* and has three serious disadvantages. First of all, convergence to the fixed point can be exceedingly slow. Secondly, the method can only locate stable fixed points. Thirdly, it may be difficult to tell when the steady state has been reached.

A more sophisticated method is to exploit the fact that locating $\overline{\boldsymbol{x}}^*$ is equivalent to locating zeros of the error function

$$\boldsymbol{E}\left(\overline{\boldsymbol{x}}\right) = \boldsymbol{P}\left(\overline{\boldsymbol{x}}\right) - \overline{\boldsymbol{x}}, \quad \boldsymbol{E}\left(\overline{\boldsymbol{x}}\right) \in \mathbb{R}^{n-1}, \tag{5.3}$$

since we have that $\overline{\boldsymbol{x}}^* = \boldsymbol{P}\left(\overline{\boldsymbol{x}}^*\right)$. The *Newton-Raphson algorithm* (Parker and Chua, 1989) is a general algorithm for locating zeros of a differentiable function, and it may therefore be employed for locating $\overline{\boldsymbol{x}}^*$. By starting from an inital guess, $\overline{\boldsymbol{x}}^k$, of the fixed point, the Newton-Raphson algorithm calculates a more accurate estimate of $\overline{\boldsymbol{x}}^*$ through the formula

$$\overline{\boldsymbol{x}}^{k+1} = \overline{\boldsymbol{x}}^k - \boldsymbol{J}_E\left(\overline{\boldsymbol{x}}^k\right)^{-1} \boldsymbol{E}\left(\overline{\boldsymbol{x}}^k\right), \tag{5.4}$$

where $\boldsymbol{J}_E\left(\overline{\boldsymbol{x}}^k\right)$, which is the Jacobian of the error function $\boldsymbol{E}\left(\overline{\boldsymbol{x}}\right)$, is defined by

$$\boldsymbol{J}_E = \frac{\partial \boldsymbol{E}}{\partial \overline{\boldsymbol{x}}} = \begin{bmatrix} \frac{\partial E_1}{\partial \overline{x}_1} & \cdots & \frac{\partial E_1}{\partial \overline{x}_{n-1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial E_{n-1}}{\partial \overline{x}_1} & \cdots & \frac{\partial E_{n-1}}{\partial \overline{x}_{n-1}} \end{bmatrix} \in \mathbb{R}^{(n-1)\times(n-1)}. \tag{5.5}$$

The Jacobian $\boldsymbol{J}_E\left(\overline{\boldsymbol{x}}^k\right)$ can be calculated numerically by defining

$$d\overline{\boldsymbol{x}}_i = [0, \ldots, 0, \Delta_i, 0, \ldots, 0]^T \in \mathbb{R}^{n-1}, \tag{5.6}$$

where the i-th element is non-zero, and Δ_i is a small perturbation of $\overline{x}_i$ along $\mathcal{S}$. Column i of $\boldsymbol{J}_E\left(\overline{\boldsymbol{x}}^k\right)$ may then be approximated numerically as

$$\frac{\partial \boldsymbol{E}}{\partial \overline{x}_i}\left(\overline{\boldsymbol{x}}^k\right) \approx \frac{\boldsymbol{E}\left(\overline{\boldsymbol{x}}^k + \boldsymbol{d}\overline{\boldsymbol{x}}_i\right) - \boldsymbol{E}\left(\overline{\boldsymbol{x}}^k\right)}{\Delta_i}. \tag{5.7}$$

This enables a column-wise construction of $\boldsymbol{J}_E\left(\overline{\boldsymbol{x}}^k\right)$. If the initial condition is within the basin of attraction of a periodic orbit, the Newton-Raphson algorithm will converge rapidly towards the fixed point $\overline{\boldsymbol{x}}^*$.

Analysing Stability of a Periodic Orbit

As explained in Section 5.1.1, a fixed point $\overline{\boldsymbol{x}}^*$ of the Poincaré map corresponds to a periodic orbit of the underlying dynamical system. Once the fixed point has been found using e.g. the Newton-Raphson algorithm, the stability of the periodic orbit may be tested by investigating if the fixed point is a stable equilibrium point of the Poincaré map. This is done by calculating the Jacobian linearization of the Poincaré map about the fixed point, i.e. by calculating the Jacobian $\boldsymbol{J}_P\left(\overline{\boldsymbol{x}}^*\right) = \frac{\partial \boldsymbol{P}}{\partial \overline{\boldsymbol{x}}}\big|_{\overline{\boldsymbol{x}}=\overline{\boldsymbol{x}}^*} \in \mathbb{R}^{(n-1)\times(n-1)}$. $\boldsymbol{J}_P\left(\overline{\boldsymbol{x}}^*\right)$ is calculated by following the same procedure as for calculating $\boldsymbol{J}_E\left(\overline{\boldsymbol{x}}^k\right)$ in (5.5). The Poincaré map linearized about the fixed point is thereby given as $\overline{\boldsymbol{x}}\left[k+1\right] = \boldsymbol{J}_P\left(\overline{\boldsymbol{x}}^*\right)\overline{\boldsymbol{x}}\left[k\right]$. This is a linear discrete-time system which is *exponentially stable* if the magnitude of all the eigenvalues of $\boldsymbol{J}_P\left(\overline{\boldsymbol{x}}^*\right)$ are strictly less than one. The fixed point $\overline{\boldsymbol{x}}^*$ of the Poincaré map, and thereby also the periodic orbit of the underlying dynamical system, is therefore *locally exponentially stable* if the magnitude of all the eigenvalues of $\boldsymbol{J}_P\left(\overline{\boldsymbol{x}}^*\right)$ are strictly less than one.

5.2 Straight Line Path Following Control of Snake Robots

In this section, we propose a control law that enables a snake robot to track a straight path. We assume that the snake robot is described by the model in (2.34) and that the robot is influenced by anisotropic viscous ground friction forces described by (2.24).

5.2.1 Control Objective

In order to track the desired straight path, we define the global coordinate system so that the global x axis is aligned with the desired path. The

position of the snake robot along the global y axis, p_y, is then the shortest distance from the robot to the desired path (i.e. the cross-track error) and the heading $\overline{\theta}$ of the robot, which was defined in (2.2), is the angle that the robot forms with the desired path. The control objective is thereby to regulate p_y and $\overline{\theta}$ so that they oscillate about zero, i.e. so that their trajectories trace out a limit cycle containing $\left(p_y = 0, \overline{\theta} = 0\right)$ in its interior. As explained in the introduction of this chapter, we do not attempt to regulate p_y and $\overline{\theta}$ to zero since we expect the heading and position of the robot to display oscillating behaviour during locomotion along the desired path.

Since snake robot locomotion is a slow form of robotic mobility, which is generally employed for traversability purposes, we consider it less important to accurately control the forward velocity of the robot. During path following with a snake robot, it therefore makes sense to focus all the control efforts on converging to the path and subsequently progressing along the path at some nonzero forward velocity $\overline{v}_t(t) > 0$, where $\overline{v}_t(t)$ is the forward velocity of the robot defined in (2.5).

From the above discussion, the control problem is to design a feedback control law such that for all $t > t_c \geq 0$, there exists a $\tau \in [t, t + T]$ satisfying

$$p_y(\tau) = 0, \tag{5.8}$$

$$\overline{\theta}(\tau) = 0, \tag{5.9}$$

$$\overline{v}_t(t) > 0, \tag{5.10}$$

where t_c is some (unknown) finite time duration corresponding to the time it takes the snake robot to converge to the desired straight path, and $T > 0$ is some constant that characterize the time period of the cyclic gait pattern of the snake robot. In other words, we require that p_y and $\overline{\theta}$ are zero at least once *within* each cycle of the locomotion since this means that p_y and $\overline{\theta}$ oscillate about zero. Note that we require $\overline{v}_t(t) > 0$ for all $t > t_c$.

5.2.2 The Straight Line Path Following Controller

We choose to propel the snake robot forward according to the gait pattern lateral undulation defined in (4.32), which is achieved by controlling joint $i \in \{1, \ldots, N-1\}$ according to

$$\phi_{i,\text{ref}} = \alpha \sin\left(\omega t + (i-1)\,\delta\right) + \phi_o, \tag{5.11}$$

where α and ω are the amplitude and frequency, respectively, of the sinusoidal joint motion, δ determines the phase shift between the joints, and

ϕ_o is a joint offset, which we assume to be identical for all joints. Note that with the gait pattern in (5.11), the period of the cyclic locomotion considered in control objectives (5.8) and (5.9) will be $T = 2\pi/\omega$.

The heading $\overline{\theta}$ of the robot was defined in (2.2) as

$$\overline{\theta} = \frac{1}{N}\sum_{i=1}^{N}\theta_i. \tag{5.12}$$

In order to steer the snake robot towards the desired straight path (i.e. the global x axis), we define the heading reference angle according to the Line-of-Sight (LOS) guidance law

$$\overline{\theta}_{\text{ref}} = -\arctan\left(\frac{p_y}{\Delta}\right), \tag{5.13}$$

where p_y is the cross-track error, and $\Delta > 0$ is a design parameter referred to as the *look-ahead distance* that influences the rate of convergence to the desired path. This LOS guidance law is commonly used during e.g. path following control of marine surface vessels (see e.g. Fossen, 2002; Fredriksen and Pettersen, 2006). As illustrated in Fig. 5.2, the LOS angle $\overline{\theta}_{\text{ref}}$ corresponds to the orientation of the snake robot when it is headed towards the point located a distance Δ ahead of itself along the desired path.

We know from Property 4.7 that ϕ_o can be used to control the direction of the locomotion, and we therefore conjecture that we can steer the heading $\overline{\theta}$ according to the LOS angle in (5.13) by defining this joint angle offset according to

$$\phi_o = k_\theta\left(\overline{\theta} - \overline{\theta}_{\text{ref}}\right), \tag{5.14}$$

where $k_\theta > 0$ is a controller gain. To make the joints track the reference angles given by (5.11), we set the control input $\boldsymbol{u} \in \mathbb{R}^{N-1}$ of the model (2.34) according to the linearizing control law (2.43), and we set the new control input $\overline{\boldsymbol{u}} \in \mathbb{R}^{N-1}$ according to the simple joint controller defined in (4.33) as

$$\overline{u}_i = k_p\left(\phi_{i,\text{ref}} - \phi_i\right) - k_d\dot{\phi}_i, \tag{5.15}$$

where $k_p > 0$ and $k_d > 0$ are controller gains.

5.3 Stability Analysis of the Path Following Controller based on the Poincaré Map

In this section, we employ the theory of *Poincaré maps* (see Section 5.1) to prove that the path following controller proposed in Section 5.2 generates

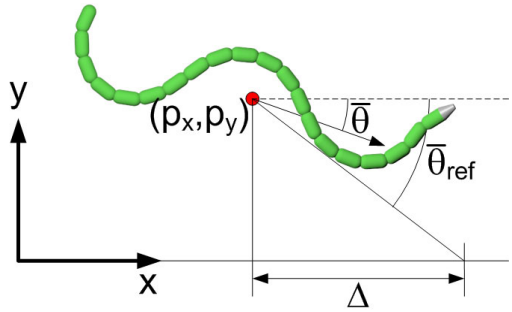


Figure 5.2: The Line-of-Sight (LOS) guidance law.

a *locally exponentially stable periodic orbit* in the state space of the snake robot as the robot locomotes along the desired straight path, i.e. the global x axis.

5.3.1 Converting the Snake Robot Model to a Time-periodic Autonomous System

Stability analysis of the time-periodic state variables of the snake robot by use of Poincaré maps requires that the model of the snake robot represents an *autonomous* system, i.e. a system not explicitly dependent on time. More specificly, the stability conclusions described in Section 5.1.2 are not valid if the Jacobian matrix $\boldsymbol{J}_P(\overline{\boldsymbol{x}}^*)$ is a function of time t. However, since the path following controller proposed in Section 5.2 depends explicitly on time, i.e. $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x}, t)$, the model of the snake robot (2.34) with the path following controller can be written as

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{\theta}} \\ \dot{\boldsymbol{p}} \\ \ddot{\boldsymbol{\theta}} \\ \ddot{\boldsymbol{p}} \end{bmatrix} = \boldsymbol{F}(\boldsymbol{x}, t), \tag{5.16}$$

which is a *non-autonomous* system since time t is explicitly present in the system equations. We therefore follow the approach described in Parker and Chua (1989) in order to convert the snake robot model to an autonomous system by simply augmenting the state vector $\boldsymbol{x}$ with an extra state $\beta = 2\pi t/T$, where $T = 2\pi/\omega$ is the period of the cyclic locomotion generated by the gait pattern in (5.11). We make β periodic by enforcing that $0 \leq \beta <$

2π, i.e. we set β to zero each time $\beta = 2\pi$. The model (2.34) with the path following controller can thereby be written as the following *autonomous* system:

$$\begin{array}{ll} \dot{\boldsymbol{x}} = \boldsymbol{F}\left(\boldsymbol{x}, \frac{T}{2\pi}\beta\right), & \boldsymbol{x}\left(t_0\right) = \boldsymbol{x}_0, \\ \dot{\beta} = \frac{2\pi}{T}, & \beta\left(t_0\right) = \frac{2\pi t_0}{T}. \end{array} \tag{5.17}$$

We have, in other words, encapsulated time t in the new state variable β, which is periodic since $0 \leq \beta < 2\pi$.

5.3.2 Specification of the Poincaré Section for the Snake Robot

During locomotion along the global positive x axis, our goal is that the x axis position of the snake robot, p_x, increases, while all other states of the snake robot in (2.34) trace out a stable limit cycle in the state space. We therefore exclude p_x from the Poincaré map of the snake robot, which produces a *partial Poincaré map* (Westervelt *et al.*, 2007). Exclusion of p_x has no effect on the other state variables since p_x is not present in any of their derivatives in (2.34). The synthesis and analysis of lateral undulation in Chapter 4, which is the basis of the path following controller proposed in Section 5.2, enable us to argue that forward motion along the x axis (increase of p_x) is achieved as long as the remaining state variables trace out a stable periodic orbit.

We choose the global x axis as the Poincaré section $\mathcal{S}$ of the system in (5.17). Since p_x is not included in the Poincaré map, we write $\mathcal{S} = \left\{ \left(\boldsymbol{\theta}, p_y, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{p}}, \beta \right) | p_y = 0 \right\}$. Following the notation in Section 5.1, the vector of *independent* time-periodic states constrained to $\mathcal{S}$ is given by

$$\overline{x} = \begin{bmatrix} \boldsymbol{\theta} \\ \dot{\boldsymbol{\theta}} \\ \dot{\boldsymbol{p}} \\ \beta \end{bmatrix} \in \mathbb{R}^{2N+3}. \tag{5.18}$$

Note that the considered Poincaré map is *one-sided*, which means that the Poincaré section is crossed when p_y switches from a positive to a negative value, i.e. when the CM position of the robot crosses the x axis from above.

5.3.3 Stability Analysis of the Poincaré Map

In order to investigate the stability of the path following controller, we considered a three-linked snake robot where $N = 3$, $l = 0.07$ m, $m = 1$ kg,

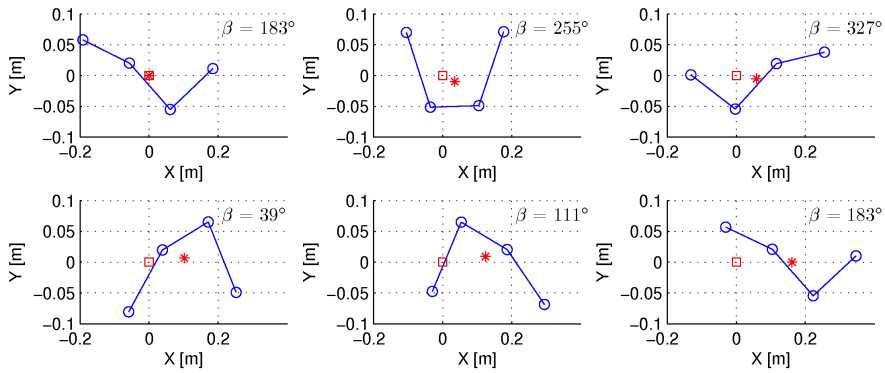


Figure 5.3: The motion of the snake robot over one period of the cyclic locomotion.

and $J = 0.0016$ kgm^2. The ground friction coefficients were chosen as $c_t = 1$ and $c_n = 10$, and the parameters of the path following controller in Section 5.2 were $\alpha = 70°$, $\omega = 70°/s$, $\delta = 70°$, $k_p = 20$, $k_d = 5$, $k_\theta = 0.3$, and $\Delta = 0.42$ m.

The Poincaré map of the snake robot model in (5.17) was calculated as described in Section 5.1.2 using *Matlab R2008b* on a laptop running *Windows XP*. The *ode45* solver in Matlab was used with a relative and absolute error tolerance of 10^{-6}. The Newton-Raphson algorithm described in Section 5.1.2 calculated the fixed point, $\overline{\boldsymbol{x}}^* \in \mathbb{R}^9$, of the Poincaré map as

$$\begin{aligned}\overline{\boldsymbol{x}}^* = [&-15.0°, -32.6°, 27.6°, -72.4°/\text{s}, 13.7°/\text{s}, 66.7°/\text{s},\\ &4.6 \text{ cm/s}, -1.2 \text{ cm/s}, 182.5°]^T .\end{aligned} \tag{5.19}$$

A plot of the cyclic locomotion of the snake robot over one period is shown in Fig. 5.3. The initial state of the snake robot was given by $\overline{\boldsymbol{x}}^*$ and the initial position was $\boldsymbol{p} = \boldsymbol{0}$. After one period of the motion, the state variables returned to their initial value, $\overline{\boldsymbol{x}}^*$. At this point, however, the position of the snake robot along the x axis had increased, which was also our goal. To clearly illustrate the limit cycle behaviour of the periodic state variables in (5.18), a 3D plot of the three absolute link angles over one period is given to the left in Fig. 5.4.

The Jacobian linearization of the Poincaré map about the fixed point (5.19) was calculated as described in Section 5.1.2. The magnitude of the

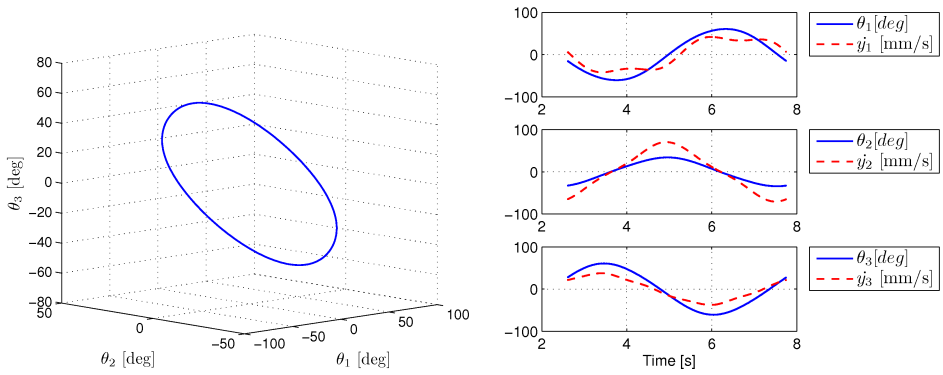


Figure 5.4: Left: The limit cycle traced out by the link angles of the snake robot. Right: Plot of θ_i and $\dot{y}_i$ $(i = 1, 2, 3)$ over one period of the cyclic locomotion.

eigenvalues of $\boldsymbol{J}_P(\overline{\boldsymbol{x}}^*) \in \mathbb{R}^{9\times 9}$ was

$$|\text{eig}(\boldsymbol{J}_P(\overline{\boldsymbol{x}}^*))| = \big[0.78, 0.78, 0.0022, 2.1\times 10^{-4}, 4.9\times 10^{-5}, \\ 4.1\times 10^{-5}, 9.6\times 10^{-6}, 2.9\times 10^{-6}, 1.6\times 10^{-6}\big]^T . \tag{5.20}$$

The magnitude of all the eigenvalues are strictly less than one, which means that the periodic orbit traced out by the variables in (5.18) is *locally exponentially stable* for the given choice of controller parameters. All initial states inside the basin of attraction of this periodic orbit will converge *exponentially* to this periodic orbit.

We now summarize the above analysis with regards to the control objectives in (5.8), (5.9), and (5.10). Since the path following controller generates an exponentially stable periodic orbit, and since $p_y = 0$ is the Poincaré section of this periodic orbit (i.e. the system returns to $p_y = 0$ with time period T), we can conclude that control objective (5.8) is achieved from all initial states inside the basin of attraction of this periodic orbit. Since the snake robot is conducting lateral undulation under anisotropic ground friction conditions, the synthesis and analysis of this gait pattern in Chapter 4 implies that the robot is locomoting forward, i.e. control objective (5.10) is satisfied. In particular, Proposition 4.3 in Section 4.4 states that the propulsive force generated by the transversal motion of link $i \in \{1, 2, 3\}$ is positive as long as $\text{sgn}(\theta_i) = \text{sgn}(\dot{y}_i)$. A plot of θ_i and $\dot{y}_i$ over one period is given to the right in Fig. 5.4, which clearly shows that $\text{sgn}(\theta_i) = \text{sgn}(\dot{y}_i)$ over the majority of the period. This means that the net propulsive force on the robot is positive. Finally, since control objectives (5.8) and (5.10)

are both satisfied, control objective (5.9) must also be satisfied. In particular, if the heading did not oscillate about zero, then the snake robot would drift away from the x axis, which contradicts the achievement of objective (5.8).

Remark 5.1 *Note that since the Poincaré map and its Jacobian linearization are found basically by simulating the model of the snake robot, we have only proved that the periodic orbit of the state variables is locally exponentially stable for the given choice of numerical parameters described in the beginning of this section. This is a drawback of stability analysis based on the Poincaré map.*

5.4 Simulation Study: The Performance of the Path Following Controller

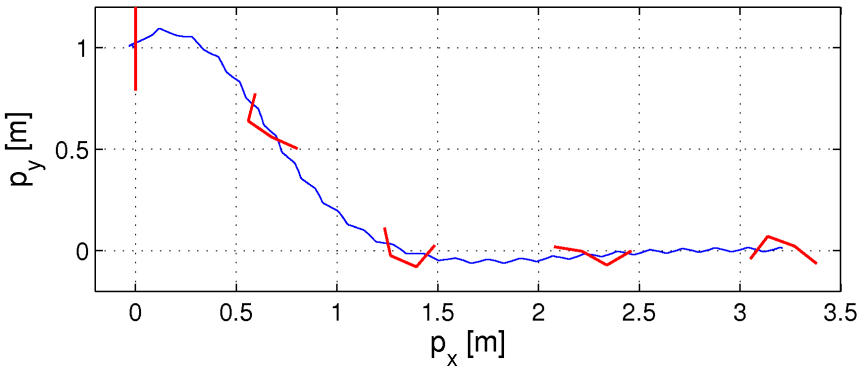
In order to illustrate the performance of the path following controller, we provide in this section a simulation result where the three-linked snake robot starts from rest with initial configuration given by $\boldsymbol{\theta} = \mathbf{0}^\circ$, $p_x = 0$ m, and $p_y = 1$ m, i.e. the snake robot is initially headed *away* from the desired path (the x axis) and the initial distance from the CM to the desired path is 1 m. We employ the model and controller parameters described in the beginning of the previous subsection.

The path traced out by the CM of the snake robot is shown in Fig. 5.5(a), where the shape and position of the robot are shown in red at $t = 0$ s, $t = 30$ s, $t = 60$ s, $t = 90$ s, and $t = 120$ s, respectively. We see that the position of the snake robot converges nicely to the desired path, i.e. the x axis. From Fig. 5.5(b), which shows the heading of the robot (solid blue) and the reference angle of the heading (dashed red), we see that the heading also converges nicely to zero, i.e. to the direction of the desired path. During the motion along the x axis near the end of the simulation, Fig. 5.5 clearly shows the oscillating behaviour of the heading and the position, which was predicted in the introduction of this chapter.

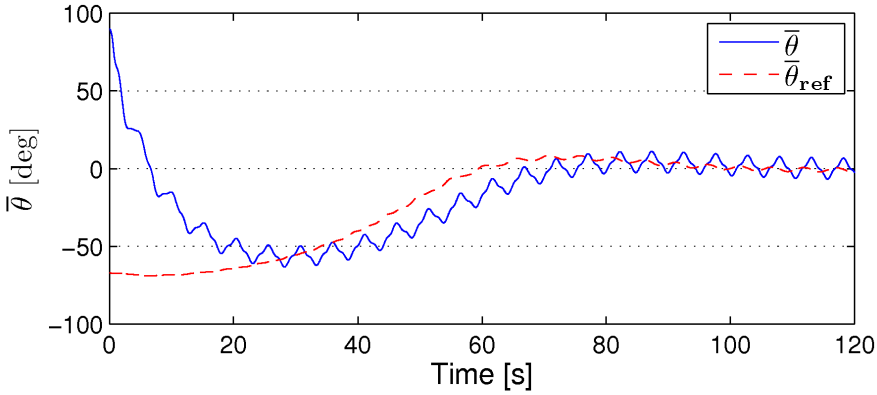
5.5 Chapter Summary

This chapter is summarized as follows:

- We have proposed a control law that enables snake robots to track straight paths.



(a) The path of the CM of the snake robot.



(b) The heading angle, $\overline{\theta}$.

Figure 5.5: Simulation of the straight line path following controller.

- We have analysed the stability of the path following controller by use of a Poincaré map, and in particular, we have shown that all state variables of the snake robot, except for the position along the path, trace out an *exponentially stable* periodic orbit during path following with the proposed controller.
- We have presented simulation results that illustrate the performance of the path following controller.

Chapter 6

A Simplified Model of Snake Robot Locomotion on Planar Surfaces

Faced with the problem of *proving* that a control strategy for the snake robot satisfies some control objective, an attractive idea is to base the controller analysis on a *simplified* model of the snake robot that avoids the complex expressions contained in the model given by (2.46). The hypothesis behind this idea is of course that the complex model in (2.46) contains nonlinear dynamics that is not essential to the overall locomotion of the snake robot. We have already seen support of this claim in the simulation results of the path following controller proposed in the previous chapter. In particular, Fig. 5.5 clearly shows that the heading and position of a snake robot display an oscillating behaviour during locomotion. However, for control design and analysis purposes, we are not particularly interested in this oscillatory dynamics as we are primarily concerned with the *overall* motion of the heading and position of the robot.

In this chapter, we therefore propose a simplified model of a planar snake robot aimed at simplifying *analytical* investigations of the equations of motion. The basic idea behind the model is to capture only the essential properties of snake robot dynamics that we derived in Chapter 4, i.e. the features that determine the overall behaviour of the snake. In order to verify that the essential features of snake robot locomotion are contained in the simplified model, we will also in this chapter repeat the stabilizability and controllability analysis presented in Chapter 4, but this time based on the simplified model. It will be shown that the stabilizability and controllability

properties of the simplified model are indeed similar to the properties of the complex model. Simulation results that compare the complex and the simplified model are presented to provide further support of this claim.

Contributions of this Chapter: The main contribution of this chapter is the *simplified model* of planar snake robot locomotion. The stabilizability and controllability analysis of the simplified model is similar to the analysis presented in Chapter 4, but is still a contribution since the analysis in this chapter is based on a different model.

Organization of this Chapter: An overview of the simplified modelling approach is given in Section 6.1. The models of the snake robot kinematics, the ground friction forces, and the snake robot dynamics are presented in Sections 6.2, 6.3, and 6.4, respectively. Subsequently, the complete simplified model is summarized in Section 6.5. The stabilizability and controllability properties of the model are studied in Section 6.6 and Section 6.7, respectively, and simulation results that compare the simplified and the complex model are presented in Section 6.8. Finally, the chapter is summarized in Section 6.9.

Publications: The material in this chapter is based on the journal paper Liljebäck *et al.* (2011*d*) and the conference papers Liljebäck *et al.* (2010*j*) and Liljebäck *et al.* (2010*l*).

6.1 Overview of the Modelling Approach

Property 4.8 from Section 4.9 tells us that lateral undulation mainly consists of link displacements that are transversal to the direction of motion. At the same time, Property 4.2 from Section 4.4 tells us that the transversal link displacements are what propel the robot forward. From these results sprung the idea of describing the mapping from body shape changes to propulsion in terms of the translational displacements of the links instead of the rotational joint motion. The motivation behind this idea is that translational motion is generally less complex to model than rotational motion. In particular, the model given by (2.46), which describes the mapping from the rotational joint motion to the propulsion of a snake robot, is quite complex.

The simplified modelling approach is illustrated in Fig. 6.1, where we see that the body shape changes of the snake robot are described, not as rotational link displacements, but as linear link displacements transversal to

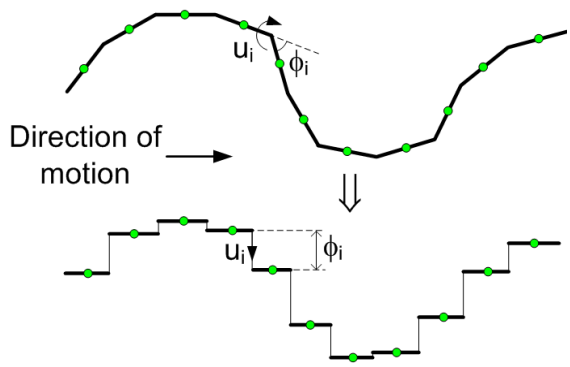


Figure 6.1: The snake robot is modelled as a series of prismatic joints that displace the CM of each link transversal to the direction of motion.

the forward direction of motion. This essentially means that we will model the revolute joints of a snake robot as prismatic (translational) joints. The rotational motion of the links during body shape changes will in other words be disregarded. However, the model will still capture the *effect* of the rotational link motion during body shape changes, which we know from Property 4.8 to be primarily a linear displacement of the CM of the links normal to the forward direction of motion.

The kinematics and dynamics of the snake robot will be detailed in the following subsections in terms of the symbols illustrated in Fig. 6.2 and Fig. 6.3. The following vectors and matrices are used in the development of the model:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & & \\ & \ddots & \ddots & \\ & & 1 & 1 \end{bmatrix} \in \mathbb{R}^{(N-1)\times N}, \boldsymbol{D} = \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \end{bmatrix} \in \mathbb{R}^{(N-1)\times N},$$

$$\boldsymbol{e} = [1, \ldots, 1]^T \in \mathbb{R}^N, \quad \overline{\boldsymbol{e}} = [1, \ldots, 1]^T \in \mathbb{R}^{N-1},$$

$$\overline{\boldsymbol{D}} = \boldsymbol{D}^T \left(\boldsymbol{D}\boldsymbol{D}^T\right)^{-1} \in \mathbb{R}^{N\times(N-1)}.$$

Remark 6.1 *Property 4.2 makes no assumption regarding the gait pattern of the snake robot. Property 4.8, however, is only valid for gait patterns where the relative link displacements transversal to the direction of motion dominate over the relative link displacements tangential to the direction of motion. The simplified model should therefore only be used to study gait patterns that comply with this condition, i.e. gait patterns with limited link*

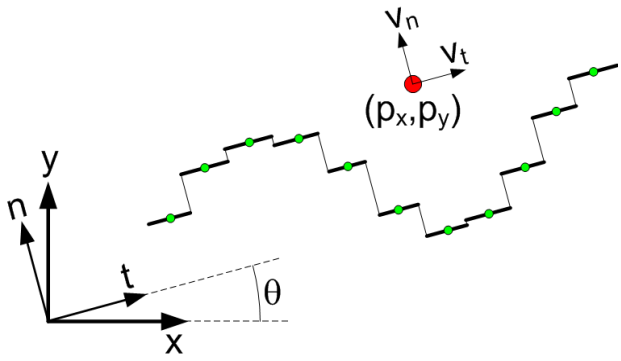


Figure 6.2: Illustration of the two coordinate frames employed in the simplified model. The global x-y frame is fixed. The t-n frame is always aligned with the snake robot.

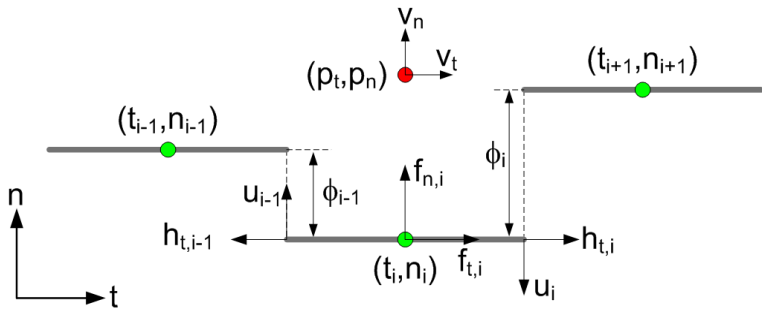


Figure 6.3: Parameters characterizing the kinematics and dynamics of the snake robot.

angles with respect to the forward direction.

6.2 The Kinematics of the Snake Robot

We consider a planar snake robot with N links of length l interconnected by $N-1$ motorized *prismatic* (translational) joints. All N links have the same mass m, and the total mass of the snake robot is thus Nm. Note that we denote the total link length in the simplified model by l, whereas the total link length in the complex model is $2l$ for notational convenience.

The snake robot moves on a horizontal and flat surface, and has $N+2$ degrees of freedom. We define the motion of the robot with respect to the two coordinate frames illustrated in Fig. 6.2. The x-y frame is the fixed global frame. The t-n frame is always aligned with the snake robot, i.e. the t and n axis always point in the *tangential* and *normal* direction of the robot, respectively. The origin of both frames are fixed and coincide. We will denote the direction of the t axis as the *tangential* or *forward* direction of the robot, and the direction of the n axis as the *normal* direction. Note that we do not refer to the t-n frame as the *body* frame of the snake robot since the t-n frame is not fixed to the robot. However, if a body frame fixed to the robot had been defined, the orientation of this frame would be identical to the orientation of the t-n frame.

The position of the snake robot is described through the coordinates of its CM (centre of mass). As seen in Fig. 6.2 and Fig. 6.3, the global frame position of the robot is denoted by $(p_x, p_y) \in \mathbb{R}^2$, while the t-n frame position is denoted by $(p_t, p_n) \in \mathbb{R}^2$. The global frame orientation of the robot is denoted by $\theta \in \mathbb{R}$ and is expressed with respect to the global x axis with counterclockwise positive direction. The angle between the global x axis and the t axis is also θ since the t-n frame is always aligned with the robot. Describing the position in a frame which is always aligned with the snake robot is inspired by and similar to a coordinate transformation proposed in Pettersen and Egeland (1996).

Remark 6.2 *A snake robot with revolute joints has no explicitly defined orientation. We therefore estimated the orientation of the snake robot in the complex model (2.46) as the mean of the absolute link angles (see Definition 2.3). With the simplified modelling approach, however, we avoid this issue since the scalar variable θ provides an explicit representation of the orientation of the snake robot.*

The relationship between the t-n frame position and the global frame

position is given by

$$p_t = p_x \cos\theta + p_y \sin\theta, \tag{6.1a}$$
$$p_n = -p_x \sin\theta + p_y \cos\theta. \tag{6.1b}$$

As illustrated in Fig. 6.2, the forward and normal direction velocity of the CM of the snake robot are denoted by $v_t \in \mathbb{R}$ and $v_n \in \mathbb{R}$, respectively. Using (2.3), the relationship between the global frame velocity of the robot and the t-n frame velocity is given by

$$\dot{p}_x = v_t \cos\theta - v_n \sin\theta, \tag{6.2a}$$
$$\dot{p}_y = v_t \sin\theta + v_n \cos\theta, \tag{6.2b}$$

and the inverse relationship is given by

$$v_t = \dot{p}_x \cos\theta + \dot{p}_y \sin\theta, \tag{6.3a}$$
$$v_n = -\dot{p}_x \sin\theta + \dot{p}_y \cos\theta. \tag{6.3b}$$

Differentiating (6.1) with respect to time and inserting (6.3) gives

$$\dot{p}_t = v_t + p_n \dot{\theta}, \tag{6.4a}$$
$$\dot{p}_n = v_n - p_t \dot{\theta}. \tag{6.4b}$$

We denote the t-n frame position of the CM of link i by $(t_i, n_i) \in \mathbb{R}^2$. The $N-1$ prismatic joints of the snake robot control the normal direction distance between the links. As seen in Fig. 6.3, the normal direction distance between link i and link $i+1$ is given by

$$\phi_i = n_{i+1} - n_i, \tag{6.5}$$

and represents the coordinate of joint i. The controlled distance ϕ_i replaces the controlled joint angle in the original model given by (2.46).

Remark 6.3 *The state ϕ_i of joint i in the simplified model is a translational* distance*, while the state ϕ_i of joint i in the complex model is a joint* angle*. In the simplified model, we therefore refer to ϕ_i as a joint coordinate instead of a joint angle.*

The link positions are constrained by the prismatic joints according to

$$t_i - t_{i+1} + l = 0, \tag{6.6a}$$
$$n_i - n_{i+1} + \phi_i = 0. \tag{6.6b}$$

These holonomic constraints may be expressed in matrix form for all links as

$$\boldsymbol{D}\boldsymbol{t} + l\overline{\boldsymbol{e}} = \boldsymbol{0}, \tag{6.7a}$$
$$\boldsymbol{D}\boldsymbol{n} + \boldsymbol{\phi} = \boldsymbol{0}, \tag{6.7b}$$

where $\boldsymbol{D}$ and $\overline{\boldsymbol{e}}$ are defined in Section 6.1, $\boldsymbol{t} = [t_1, \ldots, t_N]^T \in \mathbb{R}^N$, $\boldsymbol{n} = [n_1, \ldots, n_N]^T \in \mathbb{R}^N$, and $\boldsymbol{\phi} = \begin{bmatrix}\phi_1, \ldots, \phi_{N-1}\end{bmatrix}^T \in \mathbb{R}^{N-1}$. The t-n frame position of the CM of the snake robot can be written in terms of the link positions as

$$p_t = \frac{1}{N}\boldsymbol{e}^T\boldsymbol{t}, \tag{6.8a}$$
$$p_n = \frac{1}{N}\boldsymbol{e}^T\boldsymbol{n}, \tag{6.8b}$$

where $\boldsymbol{e}$ is defined in Section 6.1. Combining (6.7) and (6.8) gives

$$\begin{bmatrix} \boldsymbol{D} \\ \frac{1}{N}\boldsymbol{e}^T \end{bmatrix} \boldsymbol{t} = \begin{bmatrix} -l\overline{\boldsymbol{e}} \\ p_t \end{bmatrix}, \quad \begin{bmatrix} \boldsymbol{D} \\ \frac{1}{N}\boldsymbol{e}^T \end{bmatrix} \boldsymbol{n} = \begin{bmatrix} -\boldsymbol{\phi} \\ p_n \end{bmatrix}. \tag{6.9}$$

Similar to the approach in the complex model, we can use (2.10) to solve (6.9) for the link positions as

$$\boldsymbol{t} = p_t\boldsymbol{e} - l\overline{\boldsymbol{D}}\overline{\boldsymbol{e}}, \tag{6.10a}$$
$$\boldsymbol{n} = p_n\boldsymbol{e} - \overline{\boldsymbol{D}}\boldsymbol{\phi}, \tag{6.10b}$$

where $\overline{\boldsymbol{D}} = \boldsymbol{D}^T\left(\boldsymbol{D}\boldsymbol{D}^T\right)^{-1} \in \mathbb{R}^{N\times(N-1)}$. By differentiating (6.10) with respect to time and inserting (6.4), the individual link velocities are given as

$$\dot{\boldsymbol{t}} = \left(v_t + p_n\dot{\theta}\right)\boldsymbol{e}, \tag{6.11a}$$
$$\dot{\boldsymbol{n}} = \left(v_n - p_t\dot{\theta}\right)\boldsymbol{e} - \overline{\boldsymbol{D}}\dot{\boldsymbol{\phi}}. \tag{6.11b}$$

6.3 The Ground Friction Model

Following the argumentation in Section 2.4.1, we assume that the snake robot is influenced by anisotropic viscous ground friction forces, and we assume that these forces act on the CM of each link. The ground friction forces must be defined so that Property 4.2, Property 4.3, and Property 4.4 from Section 4.4 also apply to the simplifed model of the snake robot.

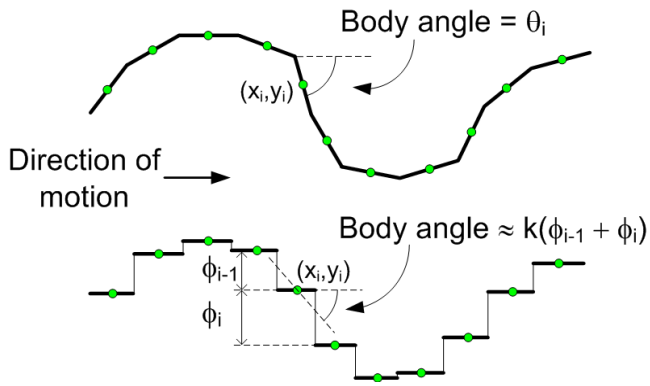


Figure 6.4: The body angle of link i is θ_i for a snake robot with revolute joints. For a snake robot with prismatic joints, we can regard $\phi_{i-1} + \phi_i$ to be approximately proportional to the body angle with respect to the forward direction.

Property 4.2 requires that the normal direction velocity of link i, which is given by $\dot{n}_i$, produces a friction force component in the tangential direction. Furthermore, Property 4.4 states that the magnitude of this force component is increased by increasing the link angle. In order to preserve these properties, we assume that the magnitude of this tangential friction force component is proportional to $\phi_{i-1} + \phi_i$, i.e. the relative distance between link $i-1$ and link $i+1$. This assumption is illustrated in Fig. 6.4, which shows that we can regard $\phi_{i-1} + \phi_i$ to be approximately proportional to the body angle with respect to the forward direction. Property 4.3 is preserved if the tangential friction force component produced by $\dot{n}_i$ is positive when $\text{sgn}\left(\phi_{i-1} + \phi_i\right) = \text{sgn}\left(\dot{n}_i\right)$ and negative otherwise.

We denote the tangential and normal direction friction force on link i by $f_{t,i}$ and $f_{n,i}$, respectively. The following friction model complies with the above requirements:

$$\begin{bmatrix} f_{t,i} \\ f_{n,i} \end{bmatrix} = \begin{bmatrix} -c_1 & c_2\left(\phi_{i-1} + \phi_i\right) \\ c_2\left(\phi_{i-1} + \phi_i\right) & -c_1 \end{bmatrix} \begin{bmatrix} \dot{t}_i \\ \dot{n}_i \end{bmatrix}_{\dot{\theta}=0}. \tag{6.12}$$

The viscous friction coefficient c_1 determines the magnitude of the friction force components resisting the tangential and normal link motion, while c_2 determines the magnitude of the tangential and normal friction force components induced by the normal and tangential link velocities, respectively. The subscript $\dot{\theta} = 0$ after the link velocity means that the friction model

disregards the link velocity components due to the angular velocity of the snake robot, $\dot{\theta}$. This is a reasonable assumption since the dynamics of the angular rotation of the snake robot will generally be much slower than the body shape dynamics. This assumption also simplifies the friction model significantly. The friction forces on all N links can now be expressed as

$$\begin{bmatrix} \boldsymbol{f}_t \\ \boldsymbol{f}_n \end{bmatrix} = \begin{bmatrix} -c_1 \boldsymbol{I}_N & c_2 \operatorname{diag}\left(\boldsymbol{A}^T \boldsymbol{\phi}\right) \\ c_2 \operatorname{diag}\left(\boldsymbol{A}^T \boldsymbol{\phi}\right) & -c_1 \boldsymbol{I}_N \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{t}} \\ \dot{\boldsymbol{n}} \end{bmatrix}_{\dot{\theta}=0}, \tag{6.13}$$

where $\boldsymbol{f}_t \in \mathbb{R}^N$ and $\boldsymbol{f}_n \in \mathbb{R}^N$ contain, respectively, the tangential and normal direction friction forces on the links, $\boldsymbol{I}_N$ is the $N \times N$ identity matrix, $\boldsymbol{A}$ is defined in Section 6.1, and the operator $\operatorname{diag}(\cdot)$ produces a diagonal matrix with the elements of its argument along its diagonal. Inserting (6.11) into (6.13) with $\dot{\theta} = 0$ gives

$$\boldsymbol{f}_t = -c_1 v_t \boldsymbol{e} + c_2 \operatorname{diag}\left(\boldsymbol{A}^T \boldsymbol{\phi}\right) \left(v_n \boldsymbol{e} - \overline{\boldsymbol{D}} \dot{\boldsymbol{\phi}}\right), \tag{6.14a}$$

$$\boldsymbol{f}_n = -c_1 v_n \boldsymbol{e} + c_1 \overline{\boldsymbol{D}} \dot{\boldsymbol{\phi}} + c_2 v_t \operatorname{diag}\left(\boldsymbol{A}^T \boldsymbol{\phi}\right) \boldsymbol{e}. \tag{6.14b}$$

6.4 The Dynamics of the Snake Robot

We now develop the equations that describe the accelerations of the snake robot. From Fig. 6.3, it can be seen that the force balance for link i is given by

$$m\ddot{t}_i = f_{t,i} + h_{t,i} - h_{t,i-1}, \tag{6.15a}$$

$$m\ddot{n}_i = f_{n,i} - u_i + u_{i-1}, \tag{6.15b}$$

where $f_{t,i}$ and $f_{n,i}$ are the ground friction forces defined in (6.12), $h_{t,i}$ and $h_{t,i-1}$ are the joint constraint forces on link i from link $i+1$ and link $i-1$, respectively, and u_i and u_{i-1} are the actuator forces at joint i and joint $i-1$, respectively. The joint constraint forces, $h_{t,i}$ and $h_{t,i-1}$, *prevent* relative motion between the links in the tangential direction, while the actuator forces, u_i and u_{i-1}, *produce* relative motion between the links in the normal direction. The force balance for all links can be written in matrix form as

$$m\ddot{\boldsymbol{t}} = \boldsymbol{f}_t + \boldsymbol{D}^T \boldsymbol{h}_t, \tag{6.16a}$$

$$m\ddot{\boldsymbol{n}} = \boldsymbol{f}_n - \boldsymbol{D}^T \boldsymbol{u}, \tag{6.16b}$$

where $\boldsymbol{D}$ is defined in Section 6.1, $\boldsymbol{h}_t = [h_{t,1}, \ldots, h_{t,N-1}]^T \in \mathbb{R}^{N-1}$, and $\boldsymbol{u} = [u_1, \ldots, u_{N-1}]^T \in \mathbb{R}^{N-1}$. Premultiplying (6.16b) by $\frac{1}{m}\boldsymbol{D}$ gives

$$\boldsymbol{D}\ddot{\boldsymbol{n}} = \frac{1}{m}\boldsymbol{D}\boldsymbol{f}_n - \frac{1}{m}\boldsymbol{D}\boldsymbol{D}^T\boldsymbol{u}. \tag{6.17}$$

By differentiating (6.7b) twice with respect to time, it is easily seen that $\boldsymbol{D}\ddot{\boldsymbol{n}} = -\ddot{\boldsymbol{\phi}}$. We can therefore write the body shape dynamics of the snake robot as

$$\ddot{\boldsymbol{\phi}} = -\frac{1}{m}\boldsymbol{D}\boldsymbol{f}_n + \frac{1}{m}\boldsymbol{D}\boldsymbol{D}^T\boldsymbol{u}. \tag{6.18}$$

Inserting (6.14b) into (6.18) and using the easily verifiable relations $\boldsymbol{D}\boldsymbol{e} = \boldsymbol{0}$, $\boldsymbol{D}\overline{\boldsymbol{D}} = \boldsymbol{I}_{N-1}$, and $\boldsymbol{D}\,\mathrm{diag}\left(\boldsymbol{A}^T\boldsymbol{\phi}\right)\boldsymbol{e} = -\boldsymbol{A}\boldsymbol{D}^T\boldsymbol{\phi}$, we get

$$\ddot{\boldsymbol{\phi}} = -\frac{c_1}{m}\dot{\boldsymbol{\phi}} + \frac{c_2}{m}v_t\boldsymbol{A}\boldsymbol{D}^T\boldsymbol{\phi} + \frac{1}{m}\boldsymbol{D}\boldsymbol{D}^T\boldsymbol{u}. \tag{6.19}$$

The tangential and normal direction acceleration of the CM of the snake robot, denoted by $\dot{v}_t$ and $\dot{v}_n$, respectively, are given as the sum of all tangential and normal direction forces on the links divided by the mass of the snake robot, Nm. This is written

$$\dot{v}_t = \frac{1}{Nm}\left(\boldsymbol{e}^T m\ddot{\boldsymbol{t}}\right) = \frac{1}{Nm}\boldsymbol{e}^T\boldsymbol{f}_t, \tag{6.20a}$$

$$\dot{v}_n = \frac{1}{Nm}\left(\boldsymbol{e}^T m\ddot{\boldsymbol{n}}\right) = \frac{1}{Nm}\boldsymbol{e}^T\boldsymbol{f}_n, \tag{6.20b}$$

where we note that the joint constraint forces, $\boldsymbol{h}_t$, and the actuator forces, $\boldsymbol{u}$, are eliminated when the link accelerations are summed (i.e. $\boldsymbol{e}^T\boldsymbol{D}^T = \boldsymbol{0}$). Inserting (6.14) into (6.20) and using the easily verifiable relations $\boldsymbol{e}^T\,\mathrm{diag}\left(\boldsymbol{A}^T\boldsymbol{\phi}\right)\boldsymbol{e} = 2\overline{\boldsymbol{e}}^T\boldsymbol{\phi}$, $\boldsymbol{e}^T\overline{\boldsymbol{D}} = \boldsymbol{0}$, and $\boldsymbol{e}^T\,\mathrm{diag}\left(\boldsymbol{A}^T\boldsymbol{\phi}\right)\overline{\boldsymbol{D}} = \boldsymbol{\phi}^T\boldsymbol{A}\overline{\boldsymbol{D}}$, we get

$$\dot{v}_t = -\frac{c_1}{m}v_t + \frac{2c_2}{Nm}v_n\overline{\boldsymbol{e}}^T\boldsymbol{\phi} - \frac{c_2}{Nm}\boldsymbol{\phi}^T\boldsymbol{A}\overline{\boldsymbol{D}}\dot{\boldsymbol{\phi}}, \tag{6.21a}$$

$$\dot{v}_n = -\frac{c_1}{m}v_n + \frac{2c_2}{Nm}v_t\overline{\boldsymbol{e}}^T\boldsymbol{\phi}. \tag{6.21b}$$

As noted in Remark 6.2, a significant difference between the snake robot with revolute joints in (2.46) and the snake robot with prismatic joints in the simplified model concerns the absolute orientation of the robot. The snake robot with revolute joints has no explicitly defined orientation since there is an independent link angle associated with each link. The orientation of the robot with prismatic joints, however, is explicitly defined in terms of

the scalar angle θ, which is also the angle of all the links. This difference must be taken into account when we model the angular acceleration, $\ddot{\theta}$, of the snake robot with prismatic joints. The model must comply with Property 4.7 from Section 4.8, which basically requires that the direction of the forward motion (i.e. the orientation θ) changes when the average of the joint coordinates, $\frac{1}{N-1}\overline{\boldsymbol{e}}^T\boldsymbol{\phi}$, is nonzero. A model that complies with this property is

$$\ddot{\theta} = -c_3\dot{\theta} + \frac{c_4}{N-1}v_t\overline{\boldsymbol{e}}^T\boldsymbol{\phi}. \tag{6.22}$$

The rotation of the snake robot is opposed by a viscous friction torque determined by the friction coefficient c_3. In addition, the average of the joint coordinates induces a torque on the robot which is scaled through the coefficient c_4 and also through the forward velocity v_t. The induced torque must be multiplied by v_t since the snake robot otherwise would experience a constant angular velocity when it is lying still with a nonzero average joint coordinate. Even though the model of $\ddot{\theta}$ is not based on first principles, Property 4.7 suggests that the behaviour of this model will closely resemble the behaviour of a snake robot with revolute joints when the coefficients c_3 and c_4 are properly chosen.

6.5 The Complete Simplified Model of the Snake Robot

We now summarize the complete simplified model of the snake robot. Since the robot has $N+2$ degrees of freedom, a state vector containing the generalized coordinates and velocities of the robot will have dimension $2N+4$. We choose the state vector of the system as

$$\boldsymbol{x} = \left[\boldsymbol{\phi}^T, \theta, p_x, p_y, \boldsymbol{v}_\phi^T, v_\theta, v_t, v_n\right]^T \in \mathbb{R}^{2N+4}, \tag{6.23}$$

where $\boldsymbol{\phi} \in \mathbb{R}^{N-1}$ are the joint coordinates, $\theta \in \mathbb{R}$ is the absolute orientation, $(p_x, p_y) \in \mathbb{R}^2$ is the global frame position of the CM, $\boldsymbol{v}_\phi = \dot{\boldsymbol{\phi}} \in \mathbb{R}^{N-1}$ are the joint velocities, $v_\theta = \dot{\theta} \in \mathbb{R}$ is the angular velocity, and $(v_t, v_n) \in \mathbb{R}^2$ is the tangential and normal direction velocity of the snake robot. From (6.2), (6.19), (6.21), and (6.22), we can write the complete model of the

snake robot as

$$\dot{\boldsymbol{\phi}} = \boldsymbol{v}_\phi, \tag{6.24a}$$

$$\dot{\theta} = v_\theta, \tag{6.24b}$$

$$\dot{p}_x = v_t \cos\theta - v_n \sin\theta, \tag{6.24c}$$

$$\dot{p}_y = v_t \sin\theta + v_n \cos\theta, \tag{6.24d}$$

$$\dot{\boldsymbol{v}}_\phi = -\frac{c_1}{m}\boldsymbol{v}_\phi + \frac{c_2}{m} v_t \boldsymbol{A}\boldsymbol{D}^T\boldsymbol{\phi} + \frac{1}{m}\boldsymbol{D}\boldsymbol{D}^T\boldsymbol{u}, \tag{6.24e}$$

$$\dot{v}_\theta = -c_3 v_\theta + \frac{c_4}{N-1} v_t \overline{\boldsymbol{e}}^T\boldsymbol{\phi}, \tag{6.24f}$$

$$\dot{v}_t = -\frac{c_1}{m} v_t + \frac{2c_2}{Nm} v_n \overline{\boldsymbol{e}}^T\boldsymbol{\phi} - \frac{c_2}{Nm}\boldsymbol{\phi}^T\boldsymbol{A}\overline{\boldsymbol{D}}\boldsymbol{v}_\phi, \tag{6.24g}$$

$$\dot{v}_n = -\frac{c_1}{m} v_n + \frac{2c_2}{Nm} v_t \overline{\boldsymbol{e}}^T\boldsymbol{\phi}, \tag{6.24h}$$

where $\boldsymbol{u} \in \mathbb{R}^{N-1}$ are the actuator forces at the joints, $\boldsymbol{A}$, $\boldsymbol{D}$, $\overline{\boldsymbol{D}}$, and $\overline{\boldsymbol{e}}$ are defined in Section 6.1, and c_1, c_2, c_3, and c_4 are positive scalar constants characterizing the external forces acting on the snake robot.

Similar to the partial feedback linearization performed for the complex model in Section 2.7, we will usually assume that the actuator forces of the simplified model are set according to the linearizing control law

$$\boldsymbol{u} = m\left(\boldsymbol{D}\boldsymbol{D}^T\right)^{-1}\left(\overline{\boldsymbol{u}} + \frac{c_1}{m}\dot{\boldsymbol{\phi}} - \frac{c_2}{m} v_t \boldsymbol{A}\boldsymbol{D}^T\boldsymbol{\phi}\right), \tag{6.25}$$

where $\overline{\boldsymbol{u}} = [\overline{u}_1, \ldots, \overline{u}_{N-1}]^T \in \mathbb{R}^{N-1}$ is a new set of control inputs. This control law transforms the joint dynamics (6.24e) into

$$\dot{\boldsymbol{v}}_\phi = \overline{\boldsymbol{u}}. \tag{6.26}$$

6.6 Stabilizability Analysis of the Simplified Model

In this section, we show that the simplified model maintains the stabilizability properties of snake robot locomotion that were derived in Section 4.2. In particular, we investigate the properties of an asymptotically stabilizing control law for the simplified model to any equilibrium point $\boldsymbol{x}^e = \left[(\boldsymbol{\phi}^e)^T, \theta^e, p_x^e, p_y^e, \boldsymbol{v}_\phi^T = \boldsymbol{0}, v_\theta = 0, v_t = 0, v_n = 0\right]^T$. As explained in Section 4.2, a well-known result presented in Brockett (1983) states that a necessary condition for the existence of a *time-invariant* (i.e. not explicitly dependent on time) *continuous* state feedback law, $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x})$, that makes

$\boldsymbol{x}^e$ asymptotically stable, is that the image of the mapping $(\boldsymbol{x}, \boldsymbol{u}) \mapsto \dot{\boldsymbol{x}}$ contains some neighbourhood of $\dot{\boldsymbol{x}} = \boldsymbol{0}$. A result presented in Coron and Rosier (1994) states that a control system that can be asymptotically stabilized (in the Filippov sense) by a *time-invariant discontinuous* state feedback law can be asymptotically stabilized by a *time-varying continuous* state feedback law. If, moreover, the control system is *affine* (i.e. linear with respect to the control input), then it can be asymptotically stabilized by a *time-invariant continuous* state feedback law. We now employ these results to prove the following fundamental result:

Theorem 6.1 *An asymptotically stabilizing feedback control law for a planar snake robot described by (6.24) to any equilibrium point must be time-varying, i.e. of the form* $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x}, t)$.

Proof. The result in Brockett (1983) states that the mapping $(\boldsymbol{x}, \boldsymbol{u}) \mapsto \dot{\boldsymbol{x}}$ must map an arbitrary neighbourhood of $\boldsymbol{x}^e$ onto a neighbourhood of $\dot{\boldsymbol{x}} = \boldsymbol{0}$. For this to be true, points of the form

$$\dot{\boldsymbol{x}} = \left[\dot{\boldsymbol{\phi}}^T = \boldsymbol{0}, \dot{\theta} = 0, \dot{p}_x = 0, \dot{p}_y = 0, \dot{\boldsymbol{v}}_\phi^T = \boldsymbol{0}, \dot{v}_\theta = 0, \dot{v}_t = \epsilon \neq 0, \dot{v}_n = 0\right]^T \tag{6.27}$$

must be contained in this mapping for some arbitrary $\epsilon \neq 0$ because points of this form are contained in every neighbourhood of $\dot{\boldsymbol{x}} = \boldsymbol{0}$. However, these points do not exist for the model (6.24) because $\dot{v}_t = 0 \neq \epsilon$ when all the other derivatives of the state vector are zero. Hence, the snake robot *cannot* be asymptotically stabilized to $\boldsymbol{x}^e$ by a *time-invariant continuous* state feedback law. Moreover, since the model is affine and *cannot* be asymptotically stabilized by a *time-invariant continuous* state feedback law, the result in Coron and Rosier (1994) proves that the system can neither be asymptotically stabilized by a *time-invariant discontinuous* state feedback law. We can therefore conclude that an asymptotically stabilizing control law for the snake robot to any equilibrium point must be time-varying, i.e. of the form $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x}, t)$. ■

The stabilizability properties of the simplified model stated in Theorem 6.1 are similar to the stabilizability properties of the complex model stated in Theorem 4.3, which supports the conjecture that the simplified model captures the essential part of the dynamics of planar snake robot locomotion.

6.7 Controllability Analysis of the Simplified Model

In this section, we show that the simplified model maintains the controllability properties of snake robot locomotion that were derived in Section 4.3.2. We assume that the joint dynamics has been linearized by the control law (6.25) so that $\dot{\boldsymbol{v}}_\phi = \overline{\boldsymbol{u}}$. This enables us to rewrite the model of the snake robot (6.24) in the standard form of a control affine system as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \sum_{j=1}^{N-1} \boldsymbol{g}_j \overline{\boldsymbol{u}}_j, \tag{6.28}$$

where $\boldsymbol{f}(\boldsymbol{x})$ contains all the terms from (6.24) with $\overline{\boldsymbol{u}} = \boldsymbol{0}_{(N-1)\times 1}$, $\overline{\boldsymbol{u}}_j$ is the jth element of the control input vector $\overline{\boldsymbol{u}} \in \mathbb{R}^{N-1}$, and

$$\boldsymbol{g}_j = \begin{bmatrix} \boldsymbol{0}_{(N+2)\times 1} \\ \boldsymbol{e}_j \\ \boldsymbol{0}_{3\times 1} \end{bmatrix}, \tag{6.29}$$

where $\boldsymbol{e}_j$ denotes the jth standard basis vector in $\mathbb{R}^{N-1}$ (the jth column of $\boldsymbol{I}_{N-1}$). It can easily be shown that the linearization of the model (6.28) about an equilibrium point $\boldsymbol{x}^e$ is not controllable since the *Kalman rank condition* is not satisfied (see Section 4.1). To study the controllability of the model in (6.28), we must therefore consider nonlinear controllability concepts.

In the following, we investigate the controllability of the simplified model (6.28) in terms of *strong accessibility* and *small-time local controllability* (STLC) (see Section 4.1) by following the same approach as for the complex model in Section 4.3.2. We assume that the snake robot consists of $N = 4$ links interconnected by $N - 1 = 3$ joints. The model of this robot has $2N + 4 = 12$ states. We argue that the following controllability results will also be valid for a snake robot with more links. In particular, a snake robot with $N > 4$ links can behave as a snake robot with $N = 4$ links by fixing $(N - 4)$ joint coordinates at zero and allowing the remaining joints to move. By calculating Lie brackets of the system vector fields in (6.28), we can construct the following *accessibility algebra* of the system evaluated at an equilibrium point $\boldsymbol{x}^e$:

$$\boldsymbol{\Delta}(\boldsymbol{x}^e) = [\boldsymbol{\Delta}_1, \ldots, \boldsymbol{\Delta}_{15}]_{\boldsymbol{x}^e} \in \mathbb{R}^{12\times 15}, \tag{6.30}$$

where

$\boldsymbol{\Delta}_1 = \boldsymbol{g}_1, \boldsymbol{\Delta}_2 = \boldsymbol{g}_2, \boldsymbol{\Delta}_3 = \boldsymbol{g}_3,$

$$\boldsymbol{\Delta}_4 = [\boldsymbol{f}, \boldsymbol{g}_1], \boldsymbol{\Delta}_5 = [\boldsymbol{f}, \boldsymbol{g}_2], \boldsymbol{\Delta}_6 = [\boldsymbol{f}, \boldsymbol{g}_3],$$
$$\boldsymbol{\Delta}_7 = [\boldsymbol{f}, [\boldsymbol{f}, \boldsymbol{g}_1]], \boldsymbol{\Delta}_8 = [\boldsymbol{f}, [\boldsymbol{f}, [\boldsymbol{f}, \boldsymbol{g}_1]]],$$
$$\boldsymbol{\Delta}_9 = [\boldsymbol{f}, [\boldsymbol{f}, [\boldsymbol{f}, [\boldsymbol{f}, \boldsymbol{g}_1]]]],$$
$$\boldsymbol{\Delta}_{10} = [\boldsymbol{g}_1, [\boldsymbol{f}, [\boldsymbol{f}, \boldsymbol{g}_2]]],$$
$$\boldsymbol{\Delta}_{11} = [\boldsymbol{g}_1, [\boldsymbol{f}, [\boldsymbol{f}, [\boldsymbol{f}, \boldsymbol{g}_2]]]],$$
$$\boldsymbol{\Delta}_{12} = [\boldsymbol{g}_1, [\boldsymbol{f}, [\boldsymbol{f}, [\boldsymbol{f}, [\boldsymbol{f}, \boldsymbol{g}_2]]]]],$$
$$\boldsymbol{\Delta}_{13} = [\boldsymbol{g}_1, [\boldsymbol{f}, [\boldsymbol{f}, [\boldsymbol{f}, [\boldsymbol{f}, [\boldsymbol{f}, \boldsymbol{g}_2]]]]]],$$
$$\boldsymbol{\Delta}_{14} = [\boldsymbol{g}_1, [\boldsymbol{f}, [\boldsymbol{f}, [\boldsymbol{f}, [\boldsymbol{f}, [\boldsymbol{f}, \boldsymbol{g}_3]]]]]],$$
$$\boldsymbol{\Delta}_{15} = [\boldsymbol{g}_2, [\boldsymbol{f}, [\boldsymbol{f}, [\boldsymbol{f}, [\boldsymbol{f}, [\boldsymbol{f}, \boldsymbol{g}_3]]]]]].$$

The accessibility algebra satisfies the following property:

Property 6.1 *The accessibility algebra* $\boldsymbol{\Delta}(\boldsymbol{x}^e)$ *in (6.30) has full rank (*$\operatorname{rank}(\boldsymbol{\Delta}(\boldsymbol{x}^e)) = 12$*) as long as the sum of the joint coordinates is nonzero, i.e. as long as* $\overline{\boldsymbol{e}}^T\boldsymbol{\phi} \neq 0$.

We do not present the expressions contained in each column of $\boldsymbol{\Delta}(\boldsymbol{x}^e)$ since the expressions are rather excessive. However, Property 6.1 can be shown to hold by employing a computer software for symbolic mathematics, such as *Matlab Symbolic Toolbox*. Note that we have included three more columns than rows in $\boldsymbol{\Delta}(\boldsymbol{x}^e)$ because different pairs of columns become linearly independent at certain configurations. Including three redundant columns ensures that $\boldsymbol{\Delta}(\boldsymbol{x}^e)$ does not drop rank at these configurations. We are now ready to state the following result:

Theorem 6.2 *A planar snake robot described by (6.24) with* $N = 4$ *links is* locally strongly accessible *from any equilibrium point* $\boldsymbol{x}^e$ *satisfying* $\overline{\boldsymbol{e}}^T\boldsymbol{\phi} \neq 0$.

Proof. By Theorem 4.1, the system is locally strongly accessible from $\boldsymbol{x}^e$ if $\boldsymbol{\Delta}(\boldsymbol{x}^e)$ in (6.30) has full rank (i.e. spans a 12-dimensional space) and does not contain the drift vector field $\boldsymbol{f}$ by itself (i.e. unbracketed). By Property 6.1, the snake robot satisfies these conditions as long as $\overline{\boldsymbol{e}}^T\boldsymbol{\phi} \neq 0$. This completes the proof. ■

We now show that the snake robot does *not* satisfy sufficient conditions for small-time local controllability (STLC). As described in Section 4.1, STLC requires that we classify the Lie brackets of the system vector fields in terms of *good* and *bad* brackets. A Lie bracket is said to be *bad* if it contains the drift vector field $\boldsymbol{f}$ an odd number of times and each control vector field $\boldsymbol{g}_j$ an even number of times (0 is even). This classification is motivated by the fact that a bad bracket *may* have directional constraints. E.g. the drift vector $\boldsymbol{f}$ is *bad* because it only allows motion in its positive

direction. According to Theorem 4.2, the snake robot is STLC from an equilibrium point $\boldsymbol{x}^e$ if it is accessible from $\boldsymbol{x}^e$ and all *bad* brackets of the system can be neutralized, i.e. written as linear combinations of *good* brackets of lower θ-*degree* or lower l-*degree*. The model of the snake robot satisfies the following property:

Property 6.2 *The brackets* $\boldsymbol{g}_j$, $\left[\boldsymbol{f}, \boldsymbol{g}_j\right]$, $\left[\boldsymbol{g}_j, \left[\boldsymbol{f}, \boldsymbol{g}_k\right]\right]$, $\left[\boldsymbol{g}_j, \left[\boldsymbol{f}, \left[\boldsymbol{f}, \boldsymbol{g}_k\right]\right]\right]$, $\left[\left[\boldsymbol{f}, \boldsymbol{g}_j\right], \left[\boldsymbol{f}, \boldsymbol{g}_k\right]\right]$, $\left[\boldsymbol{f}, \left[\boldsymbol{f}, \boldsymbol{g}_j\right]\right]$, $\left[\boldsymbol{f}, \left[\boldsymbol{f}, \left[\boldsymbol{f}, \boldsymbol{g}_j\right]\right]\right]$, $\cdots$, $\left[\boldsymbol{f}, \left[\cdots \left[\boldsymbol{f}, \boldsymbol{g}_j\right]\right] \cdots\right]$, *where* $j, k \in \{1, 2, 3\}$ *and* $j \neq k$, *are all* good *brackets, but does* not *span the entire* 12*-dimensional state space.*

We do not present the expressions contained in the brackets in Property 6.2 due to their excessive nature. However, the property can be shown to hold by employing a computer software for symbolic mathematics, such as *Matlab Symbolic Toolbox*. Property 6.2 enables us to state the following result:

Theorem 6.3 *At any equilibrium point* $\boldsymbol{x}^e$, *a planar snake robot described by (6.24) with* $N = 4$ *links does* not *satisfy the sufficient conditions for* small-time local controllability *(STLC) stated in Theorem 4.2.*

Proof. The bracket $\left[\boldsymbol{g}_j, \left[\boldsymbol{f}, \left[\boldsymbol{f}, \left[\boldsymbol{f}, \boldsymbol{g}_j\right]\right]\right]\right]$ of the system, where $j \in \{1, 2, 3\}$, is a *bad* bracket. The only *good* brackets of lower θ-degree or lower l-degree that can neutralize this bad bracket are of the form $\boldsymbol{g}_j$, $\left[\boldsymbol{f}, \boldsymbol{g}_j\right]$, $\left[\boldsymbol{g}_j, \left[\boldsymbol{f}, \boldsymbol{g}_k\right]\right]$, $\left[\boldsymbol{g}_j, \left[\boldsymbol{f}, \left[\boldsymbol{f}, \boldsymbol{g}_k\right]\right]\right]$, $\left[\left[\boldsymbol{f}, \boldsymbol{g}_j\right], \left[\boldsymbol{f}, \boldsymbol{g}_k\right]\right]$, $\left[\boldsymbol{f}, \left[\boldsymbol{f}, \boldsymbol{g}_j\right]\right]$, $\left[\boldsymbol{f}, \left[\boldsymbol{f}, \left[\boldsymbol{f}, \boldsymbol{g}_j\right]\right]\right]$, $\cdots$, $\left[\boldsymbol{f}, \left[\cdots \left[\boldsymbol{f}, \boldsymbol{g}_j\right]\right] \cdots\right]$, where $j, k \in \{1, 2, 3\}$ and $j \neq k$. By Property 6.2, these brackets do *not* span the entire 12-dimensional state space. We therefore cannot express the bad bracket as a linear combination of good brackets of lower θ-degree or lower l-degree. Since there are bad brackets of the system that cannot be neutralized, the system does not satisfy the conditions for STLC stated in Theorem 4.2. ■

The result in Theorem 6.3 is similar to the result concerning STLC of the complex model in Theorem 4.6. Regarding local accessibility of the complex model, Theorem 4.5 and Property 4.1 in Section 4.3.2 state that the accessibility algebra of the complex model has full rank except for configurations where all joint coordinates are equal ($\phi_1 = \phi_2 = \ldots = \phi_{N-1}$), which will be the case when the snake robot is lying straight or forming an arc. The condition $\overline{\boldsymbol{e}}^T \boldsymbol{\phi} \neq 0$ stated in Theorem 6.2 states that a configuration is singular when the sum of the relative linear link displacements is zero. Since the sum of the relative linear link displacements

is zero for both straight and arc shaped snake robots with revolute joints, the singular configurations of the complex model revealed by Property 4.1 are actually contained in the singular configurations stated in Theorem 6.2. This similarity supports the conjecture that the simplified model captures the essential part of the dynamics of planar snake locomotion. Note that there are singular configurations of the simplified model that do not easily translate to the complex model. A wave shape where the sum of the relative linear link displacements is zero is a singular configuration in the simplified model, but is not singular to a snake robot with revolute joints. These additional singular configurations of the simplified model arise since the sum of the joint coordinates is employed to model the rotation of the robot in accordance with Property 4.7 in Section 4.8. Nonetheless, the most important conclusion to be drawn from Theorem 6.2 is that the snake robot is locally strongly accessible from *almost* any equilibrium point, except for certain singular configurations. This is in accordance with Theorem 4.5.

6.8 Simulation Study: Comparison between the Complex and the Simplified Model

This section presents simulation results in order to compare the complex snake robot model given by (2.46) and the simplified model given by (6.24).

Remark 6.4 *No definite mapping exists between the friction coefficients c_t and c_n of the complex model and the coefficients c_1 - c_4 of the simplified model. However, knowing this mapping is not critical to the intended use of the simplified model for controller design purposes as long as we know that the* qualitative *and the approximate* quantitative *behaviour of the complex model is* contained *within the simplified model for some choice of friction coefficients. The purpose of this section is to illustrate this qualitative and quantitative similarity between the two models.*

6.8.1 Simulation Parameters

Both models were implemented and simulated in *Matlab R2008b* on a laptop running *Windows XP*. The dynamics was calculated using the *ode45* solver in Matlab with a relative and absolute error tolerance of 10^{-6}.

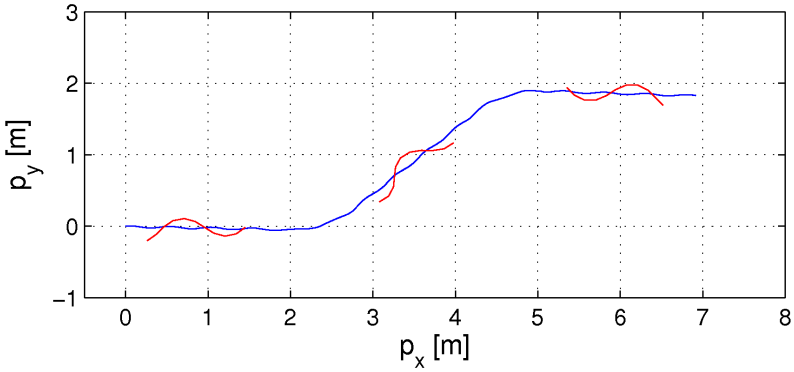
We considered a snake robot with $N = 10$ links of length $l = 0.14$ m and mass $m = 1$ kg. The links of the snake robot with revolute joints had moment of inertia $J = 0.0016$ kgm^2. The ground friction in the complex model

was defined by the viscous friction forces in (2.24) with friction coefficients $c_t = 0.5$ and $c_n = 3$. The friction coefficients of the simplified model were $c_1 = 0.45$, $c_2 = 3$, $c_3 = 0.5$, and $c_4 = 20$. The linearized control input $\overline{\boldsymbol{u}}$ of both models were set according to the exponentially stable joint controller defined in (4.34) with controller gains set to $k_p = 20$ and $k_d = 5$. The joint reference coordinates in $\boldsymbol{\phi}_{\text{ref}}$ were calculated according to the motion pattern lateral undulation defined in (4.32), and we calculated $\dot{\boldsymbol{\phi}}_{\text{ref}}$ and $\ddot{\boldsymbol{\phi}}_{\text{ref}}$ according to (4.36). The gait pattern parameters in the complex model were $\alpha = 30°$, $\omega = 70°/\text{s}$, and $\delta = 40°$, while the parameters in the simplified model were $\alpha = 0.1$ m, $\omega = 70°/\text{s}$, and $\delta = 40°$. The correspondance between $\alpha = 30°$ in the complex model and $\alpha = 0.1$ m in the simplified model was found from Fig. 4.4 in Section 4.9. The joint offset coordinate was set to $\phi_o = \frac{1}{6}\alpha$ in the time interval $t \in [20, 30]$ and $\phi_o = -\frac{1}{6}\alpha$ in the time interval $t \in [50, 60]$. The offset angle was set to $\phi_o = 0$ outside these two time intervals. The intial state of both models were set to the origin.

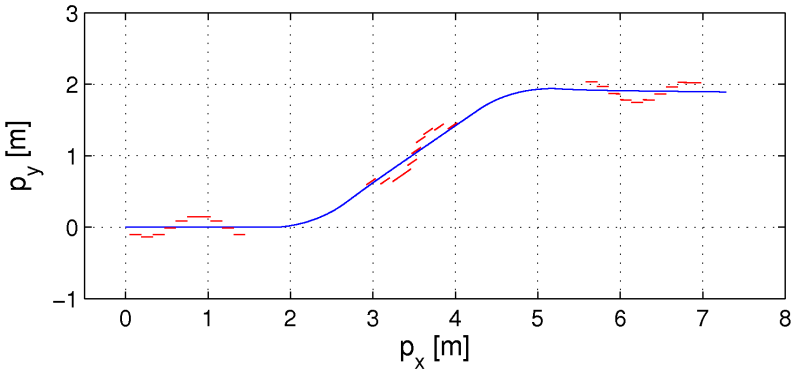
6.8.2 Simulation Results

The simulated motion of the CM of the snake robot with the complex and the simplified model is shown in Fig. 6.5(a) and Fig. 6.5(b), respectively. In both figures, the configuration of the snake robot is shown at $t = 10$ s, $t = 40$ s, and $t = 70$ s, respectively. The simulated orientation of the snake robot with both models is shown in Fig. 6.6(a). The orientation in the complex model was calculated according to (2.2), while the orientation in the simplified model was given by θ. Fig. 6.6(b) and (c) show the CM velocity of the snake robot in the global x and y direction, respectively.

The simulation results indicate that the *qualitative* behaviour of the snake robot from the simplified model is similar to the behaviour from the complex model. With the chosen numerical values of the friction coefficients, we also achieved a good *quantitative* similarity between the two models. The plots corresponding to the complex model contain high-frequency components that are not visible in the plots from the simplified model. This indicates that there are nonlinear components of the complex model that are not included in the simplified model. However, the similar behaviour of the two models indicates that the simplified model contains the parts of the complex model that determine the overall motion of the snake robot. This suggests that we may use the simplified model to develop general analysis and control design results that will also apply to the complex model.

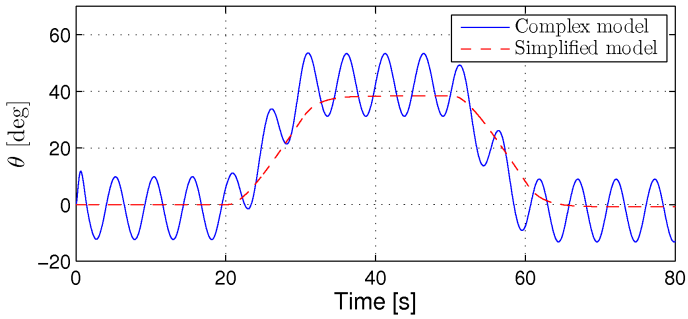


(a) The CM position from the complex model.

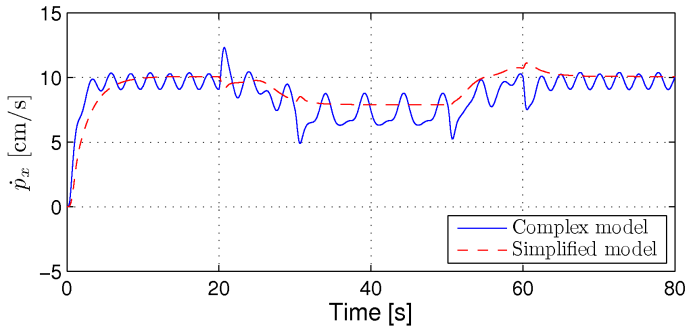


(b) The CM position from the simplified model.

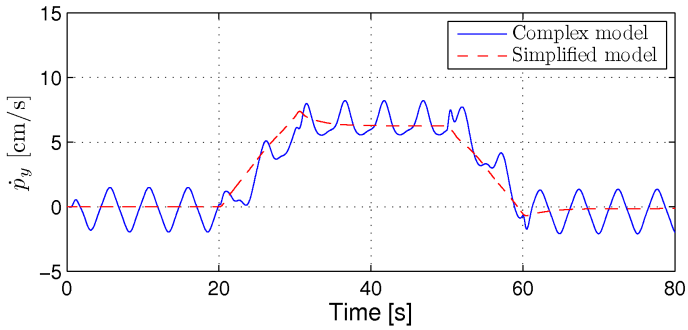
Figure 6.5: The simulated path of the CM of the snake robot.



(a) The orientation θ of the snake robot.



(b) The CM velocity in the global x direction, $\dot{p}_x$.



(c) The CM velocity in the global y direction, $\dot{p}_y$.

Figure 6.6: Simulation results that compare the complex and the simplified model of the snake robot.

6.9 Chapter Summary

This chapter is summarized as follows:

- We have presented a simplified model of a planar snake robot in (6.24), which is intended for control design and stability analysis purposes.
- We have provided support of the claim that the simplified model captures the essential part of the dynamics of planar snake robot locomotion. In particular, we have shown that:
  - The stabilizability properties of the simplified model (stated in Theorem 6.1) are similar to the stabilizability properties of the complex model (stated in Theorem 4.3).
  - The simplified model is locally strongly accessible from *almost* any equilibrium point (see Theorem 6.2), which is also the case for the complex model (see Theorem 4.5).
  - The simplified model does not satisfy sufficient conditions for STLC (see Theorem 6.3), which is also the case for the complex model (see Theorem 4.6).
  - The simulated behaviour of the snake robot from the simplified model is *qualitatively* similar to the behaviour from the complex model, and also *quantitatively* similar with a proper choice of friction coefficients.

Chapter 7

Analysis of Snake Robot Locomotion based on Averaging Theory

In the previous chapter, we extended our toolbox with a new and more manageable model of snake robot locomotion. With this new model at our disposal, an intriguing question is whether we can use the model to derive new properties of snake robot dynamics. In this chapter, we will show that this indeed is the case.

The simplified model maps the periodic motion of the joints into the resulting propulsion of the snake robot. Since the joint motion is periodic, there must be some *average* effect of the joint motion that propels the robot. In this chapter, we will use *averaging theory* (see Sanders *et al.*, 2007) to study this average effect of the joint motion during the gait pattern lateral undulation. The analysis will reveal new properties of snake robot locomotion that are both fundamental and useful from a motion planning perspective.

Contributions of this Chapter: The first contribution of this chapter is an averaged model of the velocity dynamics of a snake robot during lateral undulation. As a second contribution, we show that the average velocity of a snake robot during lateral undulation converges exponentially fast to a steady state velocity, and an analytical expression for calculating the steady state velocity is presented as a function of the gait pattern parameters. To our best knowledge, this is the first formal proof that a wheel-less snake robot with anisotropic

ground friction properties achieves forward propulsion when it moves by lateral undulation. The third contribution is a set of fundamental relationships between the gait pattern parameters of lateral undulation and the resulting forward velocity of a planar snake robot. In particular, the derived properties state that the average forward velocity of a snake robot 1) is proportional to the squared amplitude of the sinusoidal motion of each joint, 2) is proportional to the angular frequency of the sinusoidal motion of each joint, 3) is proportional to a particular function of the constant phase shift between the joints, and 4) is maximized by the phase shift between the joints that also maximizes the particular phase shift function. To our best knowledge, these fundamental properties of snake locomotion have never before been derived analytically. Each of the contributions described above are accompanied by simulation results that support the theoretical findings. Additionally, we provide experimental results with the snake robot Wheeko, which was presented in Chapter 3, in order to validate the derived relationships between the gait pattern parameters of lateral undulation and the resulting forward velocity of the robot.

Organization of this Chapter: Section 7.1 gives a brief introduction to averaging theory. The model of the velocity dynamics of the snake robot is presented in Section 7.2, and the averaged form of this model is developed in Section 7.3. Section 7.4 studies the steady state behaviour of the averaged velocity dynamics, while the relationships between the gait pattern parameters of lateral undulation and the resulting forward velocity of a planar snake robot are derived in Section 7.5. Simulation results are provided in Sections 7.6 and 7.7 in order to support the validity of the theoretical findings, while Section 7.8 presents experimental results in order to investigate the validity of the derived properties of the velocity dynamics. Finally, the chapter is summarized in Section 7.9.

Publications: The material in this chapter is based on the journal paper Liljebäck *et al.* (2011*d*) and the conference papers Liljebäck *et al.* (2010*l*), Liljebäck *et al.* (2010*f*), and Liljebäck *et al.* (2010*d*).

7.1 Introduction to Averaging Theory

Consider a system of the form

$$\dot{\boldsymbol{x}} = \varepsilon \boldsymbol{f}(t, \boldsymbol{x}), \tag{7.1}$$

where ε is a small positive parameter characterizing the magnitude of the perturbations of the system, $\boldsymbol{x} \in \mathbb{R}^n$, and $\boldsymbol{f}(t, \boldsymbol{x})$ is T-periodic, i.e. $\boldsymbol{f}(t + T, \boldsymbol{x}) = \boldsymbol{f}(t, \boldsymbol{x})$. A system that, in 'average', behaves similarly to the system in (7.1) is given by

$$\dot{\boldsymbol{x}} = \varepsilon \boldsymbol{f}_{\text{av}}(\boldsymbol{x}), \tag{7.2}$$

where

$$\boldsymbol{f}_{\text{av}}(\boldsymbol{x}) = \frac{1}{T} \int_0^T \boldsymbol{f}(\tau, \boldsymbol{x}) d\tau. \tag{7.3}$$

The above integral is calculated by treating the elements of the state vector $\boldsymbol{x}$ as constants since the underlying assumption is that the overall change of $\boldsymbol{x}$ is slow compared to the T-periodic fluctuations of $\boldsymbol{x}$. More specificly, the smallness requirement on ε ensures that $\boldsymbol{x}$ varies slowly with t relative to the periodic excitation of the system. The system response will thereby be determined predominantly by the average of the excitation. The following theorem follows directly from a more general theorem stated in Khalil (2002) (Theorem 10.4):

Theorem 7.1 *Let $\boldsymbol{f}(t, \boldsymbol{x})$ and its partial derivatives with respect to $\boldsymbol{x}$ be continuous and bounded for $(t, \boldsymbol{x}) \in [0, \infty) \times \mathbb{R}^n$. Suppose $\boldsymbol{f}$ is T-periodic in t for some $T > 0$ and ε is a positive parameter. Let $\boldsymbol{x}(t, \varepsilon)$ and $\boldsymbol{x}_{av}(t, \varepsilon)$ denote the solutions of (7.1) and (7.2), respectively. If the average system (7.2) has a* globally exponentially stable *equilibrium point and $\|\boldsymbol{x}(0, \varepsilon) - \boldsymbol{x}_{av}(0, \varepsilon)\| \leq k_0 \varepsilon$ for some $k_0 > 0$, then there exist $k > 0$ and $\varepsilon^* > 0$ such that for all $0 < \varepsilon < \varepsilon^*$,*

$$\|\boldsymbol{x}(t, \varepsilon) - \boldsymbol{x}_{av}(t, \varepsilon)\| \leq k\varepsilon \quad \textit{for all } t \in [0, \infty). \tag{7.4}$$

This theorem basically says that, for sufficiently small ε, the solutions of the original system (7.1) and the average system (7.2) remain close (of order ε) for all time if the initial conditions of the systems are close and the average system is *globally exponentially stable*. This implies that the original system will remain close to a trajectory which converges exponentially to the equilibrium point.

7.2 The Velocity Dynamics during Lateral Undulation

We will now study the velocity dynamics of the snake robot during the gait pattern lateral undulation. As defined in (4.32), lateral undulation is achieved by controlling joint $i \in \{1, \ldots, N-1\}$ according to

$$\phi_{i,\mathrm{ref}} = \alpha \sin\left(\omega t + (i-1)\,\delta\right) + \phi_o, \tag{7.5}$$

where α and ω are the amplitude and frequency, respectively, of the sinusoidal joint motion, δ determines the phase shift between the joints, and ϕ_o is a joint offset. In this chapter, we assume that ϕ_o is constant so that

$$\dot{\phi}_{i,\mathrm{ref}} = \alpha\omega \cos\left(\omega t + (i-1)\,\delta\right). \tag{7.6}$$

It was shown in Section 4.7.2 that we can achieve exponentially stable tracking of the joint reference coordinates (7.5) with the control law (4.34). In the following, we will therefore assume that the joint coordinates $\boldsymbol{\phi}$ and the joint velocities $\boldsymbol{v}_\phi = \dot{\boldsymbol{\phi}}$ are given by (7.5) and (7.6), respectively.

We define the velocity dynamics of the simplified model in terms of the dynamics of the forward direction velocity v_t, the normal direction velocity v_n, and the angular velocity v_θ of the snake robot. From (6.24f), (6.24g), and (6.24h), the velocity dynamics is thereby given as

$$\dot{v}_t = -\frac{c_1}{m} v_t + \frac{2c_2}{Nm} v_n \overline{\boldsymbol{e}}^T \boldsymbol{\phi} - \frac{c_2}{Nm} \boldsymbol{\phi}^T \boldsymbol{A} \overline{\boldsymbol{D}} \boldsymbol{v}_\phi, \tag{7.7a}$$

$$\dot{v}_n = -\frac{c_1}{m} v_n + \frac{2c_2}{Nm} v_t \overline{\boldsymbol{e}}^T \boldsymbol{\phi}, \tag{7.7b}$$

$$\dot{v}_\theta = -c_3 v_\theta + \frac{c_4}{N-1} v_t \overline{\boldsymbol{e}}^T \boldsymbol{\phi}. \tag{7.7c}$$

In order to arrive at a model of the velocity dynamics during lateral undulation which is in the standard averaging form (7.1), we assume that the amplitude α and frequency ω of the joint motion are always set according to the rule

$$\omega = \frac{k_{\alpha\omega}}{\alpha^2}, \tag{7.8}$$

where $k_{\alpha\omega} > 0$ is a controller parameter. Note that α and ω are still independent parameters since any choice of α and ω can be obtained by choosing $k_{\alpha\omega} = \alpha^2\omega$. Using (7.5), (7.6), and (7.8), and introducing the velocity state vector $\boldsymbol{v} = [v_t, v_n, v_\theta]^T \in \mathbb{R}^3$, the velocity dynamics of the

snake robot during lateral undulation can be written as

$$\dot{\boldsymbol{v}} = \begin{bmatrix} \dot{v}_t \\ \dot{v}_n \\ \dot{v}_\theta \end{bmatrix} = \boldsymbol{f}(t, \boldsymbol{v}), \tag{7.9}$$

where

$$\boldsymbol{f}(t, \boldsymbol{v}) = \begin{bmatrix} -\frac{c_1}{m} v_t + \frac{2c_2}{Nm} v_n f_1(\omega t) - \frac{c_2}{Nm} f_2(\omega t) \\ -\frac{c_1}{m} v_n + \frac{2c_2}{Nm} v_t f_1(\omega t) \\ -c_3 v_\theta + \frac{c_4}{N-1} v_t f_1(\omega t) \end{bmatrix}, \tag{7.10}$$

$$f_1(\omega t) = (N-1)\,\phi_o + \sum_{i=1}^{N-1} \alpha \sin\left(\omega t + (i-1)\,\delta\right), \tag{7.11}$$

$$\begin{aligned} f_2(\omega t) = \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} \big[& \tfrac{k_{\alpha\omega}}{\alpha} \phi_o a_{ij} \cos\left(\omega t + (j-1)\,\delta\right) \\ & + k_{\alpha\omega} a_{ij} \sin\left(\omega t + (i-1)\,\delta\right) \cos\left(\omega t + (j-1)\,\delta\right) \big], \end{aligned} \tag{7.12}$$

and a_{ij} denotes element ij of the matrix $\boldsymbol{A}\overline{\boldsymbol{D}} \in \mathbb{R}^{(N-1)\times(N-1)}$ (the matrices $\boldsymbol{A}$ and $\overline{\boldsymbol{D}}$ were defined in Section 6.1). To transform the model (7.9) into the standard form of averaging (7.1), we change the time scale from t to $\tau = \omega t$ and define $\varepsilon = 1/\omega$. Since $\frac{d}{dt} = \frac{1}{\varepsilon}\frac{d}{d\tau}$, the model (7.9) can now be written as

$$\frac{d\boldsymbol{v}}{d\tau} = \varepsilon \boldsymbol{f}(\tau, \boldsymbol{v}), \tag{7.13}$$

where

$$\boldsymbol{f}(\tau, \boldsymbol{v}) = \begin{bmatrix} -\frac{c_1}{m} v_t + \frac{2c_2}{Nm} v_n f_1(\tau) - \frac{c_2}{Nm} f_2(\tau) \\ -\frac{c_1}{m} v_n + \frac{2c_2}{Nm} v_t f_1(\tau) \\ -c_3 v_\theta + \frac{c_4}{N-1} v_t f_1(\tau) \end{bmatrix}. \tag{7.14}$$

This model is in the standard form defined in (7.1). Note that when we require ε to be small, we equivalently require that the frequency of the joint motion $\omega = 1/\varepsilon$ is large.

7.3 The Averaged Velocity Dynamics during Lateral Undulation

The averaged model of (7.13) is calculated in accordance with (7.2) as

$$\frac{d\boldsymbol{v}}{d\tau} = \varepsilon \frac{1}{2\pi} \int_0^{2\pi} \boldsymbol{f}(\tau, \boldsymbol{v}) d\tau. \tag{7.15}$$

It can be verified that

$$\frac{1}{2\pi}\int_0^{2\pi} f_1(\tau)d\tau = (N-1)\,\phi_o, \tag{7.16}$$

$$\frac{1}{2\pi}\int_0^{2\pi} f_2(\tau)d\tau = -\frac{1}{2}k_{\alpha\omega}k_\delta, \tag{7.17}$$

where the constant $k_\delta \in \mathbb{R}$ is defined as

$$k_\delta = \sum_{i=1}^{N-1}\sum_{j=1}^{N-1} a_{ij}\sin\left((j-i)\,\delta\right). \tag{7.18}$$

The averaged model can therefore be written as

$$\frac{d\boldsymbol{v}}{d\tau} = \varepsilon\left(\mathcal{A}\boldsymbol{v} + \boldsymbol{b}\right), \tag{7.19}$$

where

$$\mathcal{A} = \mathcal{A}\left(\phi_o\right) = \begin{bmatrix} -\frac{c_1}{m} & \frac{2(N-1)}{Nm}c_2\phi_o & 0 \\ \frac{2(N-1)}{Nm}c_2\phi_o & -\frac{c_1}{m} & 0 \\ c_4\phi_o & 0 & -c_3 \end{bmatrix}, \tag{7.20}$$

$$\boldsymbol{b} = \boldsymbol{b}\left(\alpha,\omega,\delta\right) = \begin{bmatrix} \frac{c_2}{2Nm}k_{\alpha\omega}k_\delta \\ 0 \\ 0 \end{bmatrix}. \tag{7.21}$$

By changing time scale back to t using that $\frac{d}{d\tau} = \varepsilon\frac{d}{dt}$, the averaged model is given by

$$\dot{\boldsymbol{v}} = \mathcal{A}\boldsymbol{v} + \boldsymbol{b}. \tag{7.22}$$

We see that the averaged model of the velocity dynamics is a linear system characterized by the parameters of the joint reference coordinates, i.e. by α, ω, δ, and ϕ_o.

Remark 7.1 *The term* average velocity *will hereafter be used to denote the velocity described by the averaged model (7.22) and should* not *be confused with the average velocity over time that we normally would define as*

$$\frac{1}{t}\int_0^t \boldsymbol{v}(\tau)d\tau. \tag{7.23}$$

In other words, the average velocity *described by the averaged model (7.22) is the velocity that changes according to the average changes of the original model (7.9).*

7.4 The Steady State Behaviour of the Velocity Dynamics during Lateral Undulation

In order to determine the stability properties of the averaged model (7.22), we remove the constant offset term $\boldsymbol{b}$ with the coordinate transformation $\boldsymbol{z} = \boldsymbol{v} + \mathcal{A}^{-1}\boldsymbol{b}$. This gives

$$\dot{\boldsymbol{z}} = \dot{\boldsymbol{v}} = \mathcal{A}\left(\boldsymbol{z} - \mathcal{A}^{-1}\boldsymbol{b}\right) + \boldsymbol{b} = \mathcal{A}\boldsymbol{z}. \tag{7.24}$$

By employing a computer software for symbolic mathematics, such as *Matlab Symbolic Toolbox*, the eigenvalues of $\mathcal{A}$ are easily calculated as

$$\operatorname{eig}(\mathcal{A}) = \begin{bmatrix} -\frac{c_1}{m} - \frac{2(N-1)}{Nm}c_2\phi_o \\ -\frac{c_1}{m} + \frac{2(N-1)}{Nm}c_2\phi_o \\ -c_3 \end{bmatrix}. \tag{7.25}$$

The equilibrium point $\boldsymbol{z} = \boldsymbol{0}$ is *globally exponentially stable* if all eigenvalues of $\mathcal{A}$ are negative (see Khalil, 2002), which is easily seen to be the case if

$$|\phi_o| < \frac{N}{2(N-1)}\frac{c_1}{c_2}. \tag{7.26}$$

This is a limit on the amplitude of the joint coordinate offset ϕ_o, and is a function of the friction coefficients c_1 and c_2. It is not surprising that the model of the snake robot (6.24) can become unstable since the approach of modelling the link motion as translational displacements must naturally break down at some point. The instability issue in (7.26) is not relevant to a snake robot with revolute joints since the normal direction distance between the links of this mechanism will be physically constrained by the revolute joints.

Assuming that we choose ϕ_o to satisfy the limit (7.26), then $\boldsymbol{z}$ will converge exponentially to zero, which means that $\boldsymbol{v}$ will converge exponentially to $-\mathcal{A}^{-1}\boldsymbol{b}$, which means that the average velocity will converge exponentially to the steady state velocity

$$\boldsymbol{v}^* = \begin{bmatrix} v_t^* \\ v_n^* \\ v_\theta^* \end{bmatrix} = -\mathcal{A}^{-1}\boldsymbol{b}, \tag{7.27}$$

which is given analytically by

$$v_t^* = k_{\alpha\omega} k_\delta \frac{N c_1 c_2}{2\left(N^2 c_1^2 - (4N^2 - 8N + 4)\, c_2^2 \phi_o^2\right)}, \tag{7.28a}$$

$$v_n^* = k_{\alpha\omega} k_\delta \frac{\phi_o\,(N-1)\, c_2^2}{N^2 c_1^2 - (4N^2 - 8N + 4)\, c_2^2 \phi_o^2}, \tag{7.28b}$$

$$v_\theta^* = k_{\alpha\omega} k_\delta \frac{\phi_o N c_1 c_2 c_4}{2c_3\left(N^2 c_1^2 - (4N^2 - 8N + 4)\, c_2^2 \phi_o^2\right)}. \tag{7.28c}$$

We can see that the resulting steady state velocity of the snake robot is proportional to the controller parameters $k_{\alpha\omega} = \alpha^2\omega$ and k_δ, and that the velocity also depends on nonlinear terms involving the joint coordinate offset ϕ_o.

Since the averaged model of the velocity dynamics given by (7.22) is *globally exponentially stable* (assuming that (7.26) is satisfied), it follows from Theorem 7.1 that, for sufficiently small ε (i.e. for sufficiently large ω), the average velocity given by (7.22) will approximate the exact velocity (7.9) for all time, and that the error of this approximation is of order ε, i.e. bounded in accordance with (7.4). In this thesis, we will not investigate the lower limit of ω corresponding to some maximum error bound. However, the simulation results presented in Section 7.6 show that the exact and the average velocity agree well when ω is set to values that are commonly used for snake robot locomotion.

We now summarize the above conclusions.

Theorem 7.2 *Consider a planar snake robot described by (6.24). Suppose the joint coordinates $\boldsymbol{\phi}$ are controlled in exact accordance with (7.5) and (7.6), and that the joint coordinate offset ϕ_o satisfies (7.26). Then there exist $k > 0$ and $\omega^* > 0$ such that for all $\omega > \omega^*$,*

$$\|\boldsymbol{v}(t) - \boldsymbol{v}_{av}(t)\| \leq \tfrac{k}{\omega} \quad \text{for all } t \in [0, \infty)\,, \tag{7.29}$$

where $\boldsymbol{v}(t)$ denotes the exact velocity of the snake robot given by (7.9) and $\boldsymbol{v}_{av}(t)$ denotes the average velocity given by (7.22). Furthermore, the average velocity $\boldsymbol{v}_{av}(t)$ of the snake robot will converge exponentially fast to the steady state velocity $\boldsymbol{v}^*$ given by (7.27).

7.5 Relationships between the Gait Parameters and the Forward Velocity during Lateral Undulation

Theorem 7.2 is a powerful result. First of all, it proves mathematically a hypothesis on snake robot locomotion developed in the research community through empirical studies of biological snakes and simulation results, but for which there has, to our best knowledge, not been presented any proof before. In particular, it proves that lateral undulation enables a wheel-less snake robot with anisotropic ground friction properties to achieve forward propulsion. Secondly, the result gives an analytical expression for the steady state velocity as a function of the controller parameters α, ω, δ, and ϕ_o, i.e. the amplitude, frequency, phase shift and offset of the joint motion during lateral undulation. This information is relevant for motion planning purposes. We can for example immediately see that the steady state velocity of the snake robot when it conducts lateral undulation with zero joint offset ($\phi_o = 0$) is given by $v_t^* = \frac{c_2}{2Nc_1} k_{\alpha\omega} k_\delta$, $v_n^* = 0$, and $v_\theta^* = 0$. A final powerful feature of Theorem 7.2 is that it applies to snake robots with an arbitrary number of links N.

In the following, we will use Theorem 7.2 to deduce some fundamental relationships between the gait pattern parameters and the forward velocity of the snake robot. The forward velocity is seen from (7.28a) to be proportional to the controller parameter $k_{\alpha\omega} = \alpha^2\omega$, i.e. the forward velocity is proportional to the square of the amplitude of the joint motion, α^2, and also proportional to the angular frequency, ω, of the joint motion. This information is useful from a motion planning perspective since it tells us that an increase/decrease of the forward velocity by a certain factor can be achieved by increasing/decreasing ω by the same factor or by increasing/decreasing α by the square root of this factor.

It is also seen from (7.28a) that the forward velocity of the snake robot is proportional to the function k_δ defined in (7.18). Since k_δ is a function of the phase shift δ between the joints, this means that the phase shift δ that will maximize the forward velocity can be determined analytically as the δ that maximizes k_δ. This is particularly interesting since we are now able to analytically determine the optimal phase shift δ that maximizes the forward velocity of a planar snake robot with an arbitrary number of links N. Fig. 7.1 presents a plot of the maximum value of k_δ as a function of the number of links N. For each N, the maximum value of k_δ was found using the mathematical computer software *Matlab*. The optimal phase shift

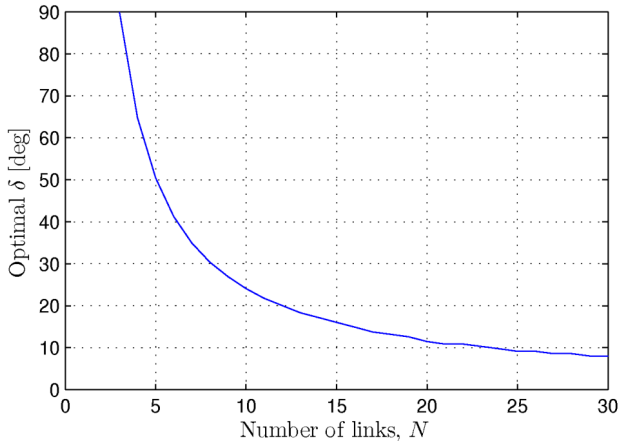


Figure 7.1: The phase shift δ that maximizes the forward velocity of a planar snake robot as a function of the number of links N.

is e.g. $\delta = 90°$ for $N = 3$ links, $\delta = 50.4°$ for $N = 5$ links, $\delta = 24.1°$ for $N = 10$ links, and $\delta = 11.5°$ for $N = 20$ links.

The above results can be summarized as follows:

Theorem 7.3 *Consider a planar snake robot with N links modelled by (6.24) and controlled in exact accordance with (7.5) and (7.6). The average forward velocity of the snake robot given by (7.22) will converge exponentially to a value which is proportional to:*
- the squared amplitude of the sinusoidal joint motion, α^2.
- the angular frequency of the sinusoidal joint motion, ω.
- the function of the constant phase shift, δ, between the joints given by

$$k_\delta = \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} a_{ij} \sin\left((j-i)\,\delta\right), \tag{7.30}$$

where a_{ij} denotes element ij of the matrix $\boldsymbol{A}\overline{\boldsymbol{D}}$. Moreover, for a given α and ω, the phase shift, δ, that maximizes the average forward velocity is given by the δ that maximizes k_δ.

7.6 Simulation Study: Comparison between the Exact and the Averaged Velocity Dynamics

This section presents simulation results in order to investigate the validity of Theorem 7.2, i.e. to validate the agreement between the exact model of the velocity dynamics (7.9) and the averaged model (7.22).

7.6.1 Simulation Parameters

The exact model of the snake robot was given by (6.24) under the assumption that $\boldsymbol{\phi}$ was controlled in exact accordance with (7.5). The averaged model of the snake robot was given by (7.22). Both models were implemented and simulated in *Matlab R2008b* on a laptop running *Windows XP*. The dynamics was calculated using the *ode45* solver in Matlab with a relative and absolute error tolerance of 10^{-6}.

We considered a snake robot with $N = 10$ links of length $l = 0.14$ m and mass $m = 1$ kg. The ground friction coefficients were $c_1 = 0.45$, $c_2 = 3$, $c_3 = 0.5$, and $c_4 = 20$, and the intial state of both models were set to the origin. The values of the gait pattern parameters α, ω, δ, and ϕ_o are presented with each simulation result below.

7.6.2 Simulation Results

The motion of the snake robot during lateral undulation was first simulated with the gait parameters $\alpha = 0.1$ m, $\omega = 70°/\text{s}$, $\delta = 40°$, and $\phi_o = 0$ m. Theorem 7.2 then states that the average velocity of the snake robot will converge exponentially fast to the steady state velocity $v_t^* = \frac{c_2}{2Nc_1} k_{\alpha\omega} k_\delta \approx 0.10$ m/s, $v_n^* = 0$ m/s, and $v_\theta^* = 0°/\text{s}$. This is in agreement with the simulation result shown in Fig. 7.2. The top left plot illustrates the CM position of the snake robot from the exact model together with the body shape at $t = 1$ s and $t = 30$ s, respectively. The other three plots show the exact (blue) and the average (red) velocity of the snake robot. There is almost an exact overlap between the plots from the exact and the averaged model. This suggests that $\omega = 70°/\text{s}$ is well above the (unknown) value of ω^* described in Theorem 7.2.

In the next simulation, the gait parameters were $\alpha = 0.1$ m, $\omega = 70°/\text{s}$, $\delta = 40°$, and $\phi_o = l/8$ m. The joint coordinates were, in other words, offseted by 1/8 of the link length l. In accordance with Theorem 7.2, the average velocity of the snake robot should then converge exponentially fast to $v_t^* \approx 0.11$ m/s, $v_n^* \approx 0.022$ m/s, and $v_\theta^* \approx 4.23°/\text{s}$. This agrees very

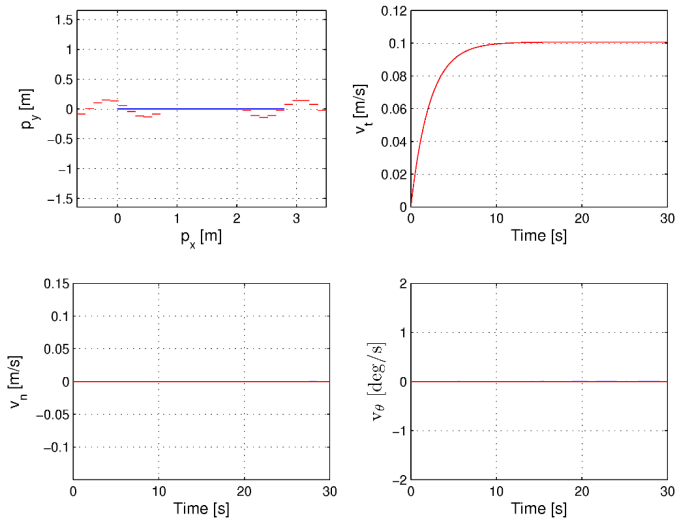


Figure 7.2: Lateral undulation along a straight line with the gait parameters $\alpha = 0.1$ m, $\omega = 70°/\text{s}$, $\delta = 40°$, and $\phi_o = 0$ m. Both the exact (blue) and the average (red) velocities are plotted.

well with the simulation result shown in Fig. 7.3, which also shows a close overlap between the velocity plots from the exact (blue) and the averaged (red) model.

In the final simulation, the gait parameters were set to $\alpha = 0.1$ m, $\omega = 30°/\text{s}$, $\delta = 40°$, and $\phi_o = -l/4$ m. The joint coordinates were, in other words, offseted by $1/4$ of the link length l. In addition, we reduced the frequency of the sinusoidal motion from $\omega = 70°/\text{s}$ to $\omega = 30°/\text{s}$ to see how this affected the estimate of the average velocity. From Theorem 7.2, the average velocity should converge to $v_t^* \approx 0.052$ m/s, $v_n^* \approx -0.022$ m/s, and $v_\theta^* \approx -4.20°/\text{s}$. This agrees very well with the simulation result shown in Fig. 7.4. The figure shows that there is still a good agreement between the velocities from the exact and the averaged model even though we reduced ω considerably.

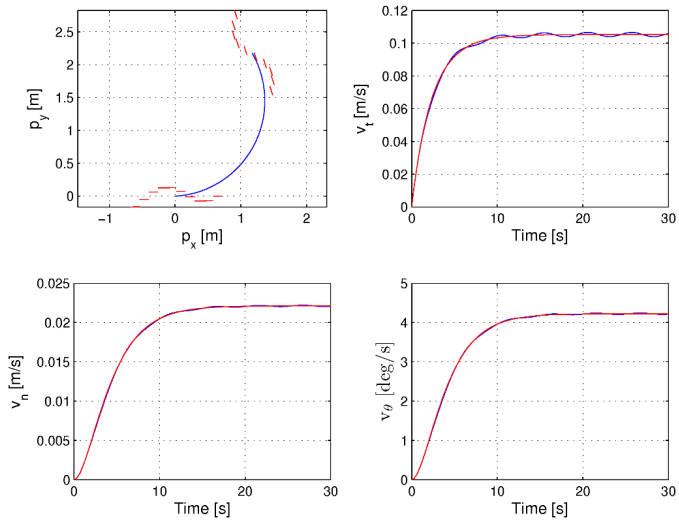


Figure 7.3: Counterclockwise turning during lateral undulation with the gait parameters $\alpha = 0.1$ m, $\omega = 70°/\text{s}$, $\delta = 40°$, and $\phi_o = l/8$ m. Both the exact (blue) and the average (red) velocities are plotted.

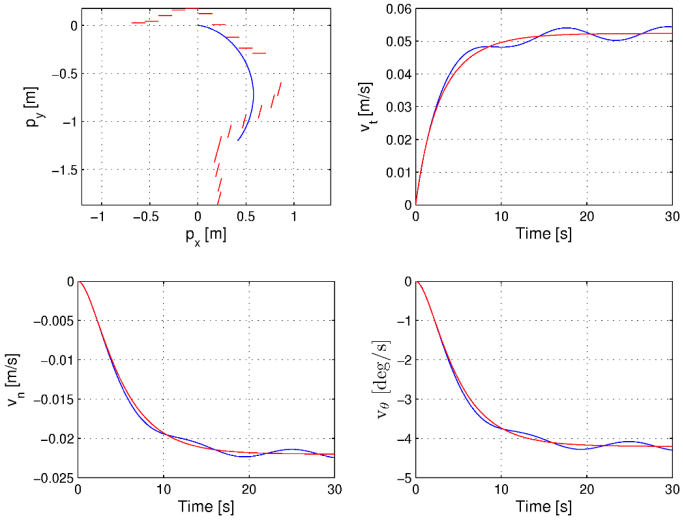


Figure 7.4: Clockwise turning during lateral undulation with the gait parameters $\alpha = 0.1$ m, $\omega = 30°/\text{s}$, $\delta = 40°$, and $\phi_o = -l/4$ m. Both the exact (blue) and the average (red) velocities are plotted.

7.7 Simulation Study: Investigation of the Relationships between Gait Parameters and Forward Velocity

This section presents simulation results in order to investigate the validity of Theorem 7.3, i.e. the validity of the relationships between the gait pattern parameters of lateral undulation and the resulting forward velocity of the snake robot. In addition to presenting simulation results from the simplified model (6.24), this section also shows that the relationships in Theorem 7.3 apply to the velocity from the complex model given by (2.46).

Remark 7.2 *The joint coordinates of the simplified model (linear link displacements) are different from the joint coordinates of the complex model (joint angles). However, it still makes sense to investigate the validity of Theorem 7.3 for a snake robot with revolute joints since, as implied by Property 4.8, the rotational link motion is what produces the linear displacements captured by the simplified model. Note also that there is an approximately linear relationship between the amplitude of the angular joint motion and the corresponding amplitude of the transversal link displacments. The relationship stated in Theorem 7.3 between the forward velocity and the squared amplitude of the sinusoidal joint motion, α^2, can therefore be expected to hold also when α denotes the amplitude of the angular joint motion of a snake robot with revolute joints.*

7.7.1 Simulation Parameters

Both the complex model (2.46) and the simplified model (6.24) were implemented and simulated in *Matlab R2008b* on a laptop running *Windows XP*. The dynamics was calculated using the *ode45* solver in Matlab with a relative and absolute error tolerance of 10^{-3} and 10^{-6} in the complex and the simplified model, respectively.

We considered snake robots with $N = 3$, $N = 5$, $N = 10$, and $N = 20$ links of length $l = 0.14$ m, mass $m = 1$ kg, and moment of inertia $J = 0.0016$ kgm^2. For the complex model, both *viscous* and *Coulomb* ground friction were considered. The friction coefficients of the viscous friction model defined in (2.24) were $c_t = 0.5$ and $c_n = 3$, while the friction coefficients of the Coulomb friction model defined in (2.19) were $\mu_t = 0.1$ and $\mu_n = 0.4$. The friction coefficients of the simplified model were $c_1 = 1$, $c_2 = 3$, $c_3 = 3$, and $c_4 = 10$. Note that the friction coefficients were more

or less arbitrarily chosen, so we did not seek to achieve agreement between the simulated velocities from the two models.

The linearized control input $\overline{\boldsymbol{u}}$ of both models were set according to the exponentially stable joint controller defined in (4.34) with controller gains $k_p = 20$ and $k_d = 5$. The joint reference coordinates were calculated according to the motion pattern lateral undulation defined in (7.5) with zero joint angle offset ($\phi_o = 0$), and the values of the controller parameters α, ω, and δ are presented with each simulation result below. The intial state of both models were set to the origin.

The simulation results below present the forward speed of the snake robot, denoted by $\overline{v}$, for different sets of controller parameters. This speed was calculated at the end of each simulation run as the linear distance travelled by the CM of the snake robot divided by the simulation time, which was chosen to be $t_{\text{sim}} = 10$ s. The forward speed was, in other words, calculated as

$$\overline{v} = \frac{\sqrt{\left(p_x\left(10\right) - p_x\left(0\right)\right)^2 + \left(p_y\left(10\right) - p_y\left(0\right)\right)^2}}{10}. \tag{7.31}$$

7.7.2 Simulation Results

Relationship between the Forward Velocity and α

Theorem 7.3 states that the average forward velocity of a planar snake robot is proportional to the squared amplitude of the sinusoidal joint motion, α^2. We investigated the validity of this result by simulating the snake robot with different values of α and calculating the resulting average forward velocity according to (7.31). The simulation results from the simplified model are shown in Fig. 7.5(a), while the simulation results from the complex model with viscous and Coulomb ground friction, respectively, are shown in Fig. 7.5(b). The number of links N and the corresponding values of ω and δ are shown at the top of each plot. In the complex model, the range of α values is shorter for large N compared to for small N since a large angle amplitude would cause a 'collision' between the head and the tail of the snake when N is large. The plots clearly show an exponential increase in the forward speed $\overline{v}$ as the amplitude α increases. This is in accordance with Theorem 7.3.

Note that the amplitude of the joint motion cannot be increased indefinitely in the complex model. For sufficiently large α, the relative link velocity components that are *tangential* to the forward direction will no longer be negligible, which is assumed in the simplified model of the snake

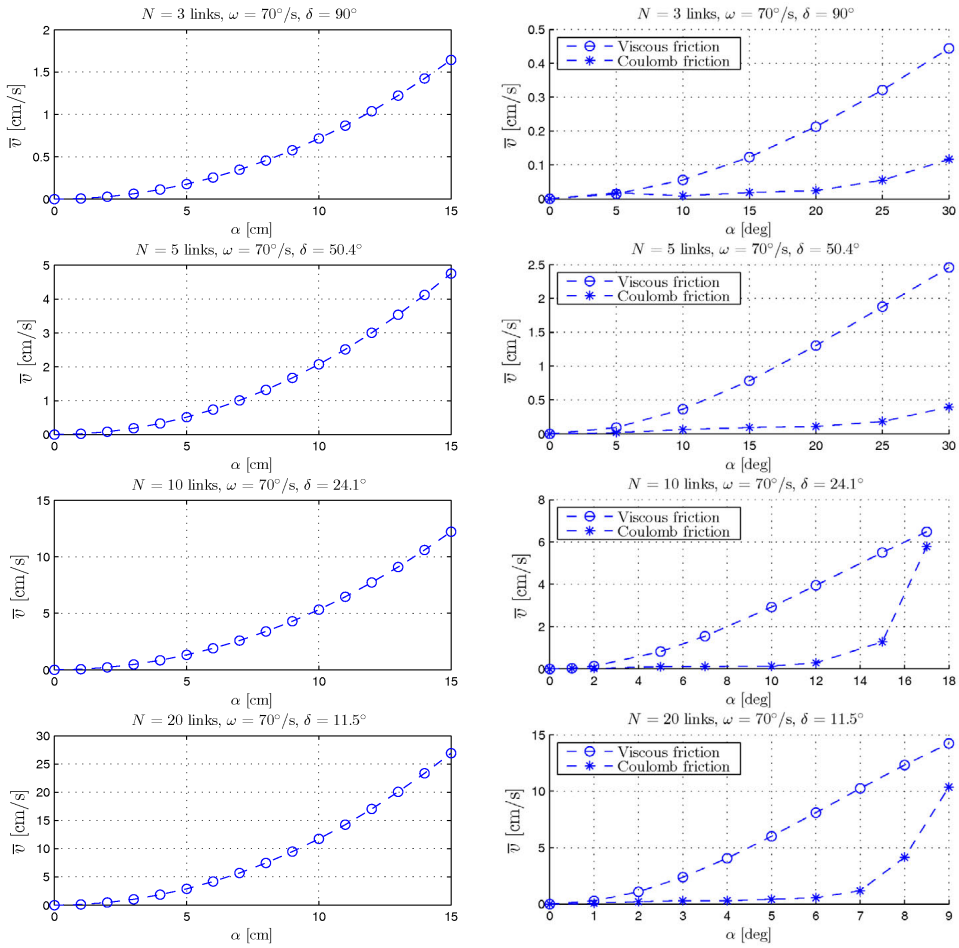
robot. It is therefore reasonable to expect that the increase in the forward velocity from the complex model will decay for large α. This decay can be seen in the plots with viscous friction in Fig. 7.5(b), which shows that the increase of the velocity has a more linear character when α becomes large.

Relationship between the Forward Velocity and ω

Theorem 7.3 states that the average forward velocity of a planar snake robot is proportional to the angular frequency, ω, of the joint motion. This result was investigated by simulating the snake robot with different values of ω and calculating the resulting average forward velocity. The simulation results from the simplified model are shown in Fig. 7.6(a), while the simulation results from the complex model with viscous and Coulomb ground friction, respectively, are shown in Fig. 7.6(b). The number of links N and the corresponding values of α and δ are shown at the top of each plot. The linear increase in the forward speed $\overline{v}$ as the frequency ω increases is clearly present in the plots from the simplified model and also the plots from the complex model with viscous ground friction. This is in accordance with Theorem 7.3. With Coulomb friction, the forward speed increases linearly for $N = 3$ links. However, for $N = 5$, $N = 10$, and $N = 20$ links, the forward speed seems to increase linearly up to a certain frequency, after which the forward speed decreases. This suggests that Coulomb friction in the complex model introduces nonlinear couplings between the controller parameters α, ω, and δ that are not present with viscous ground friction.

Relationship between the Forward Velocity and δ

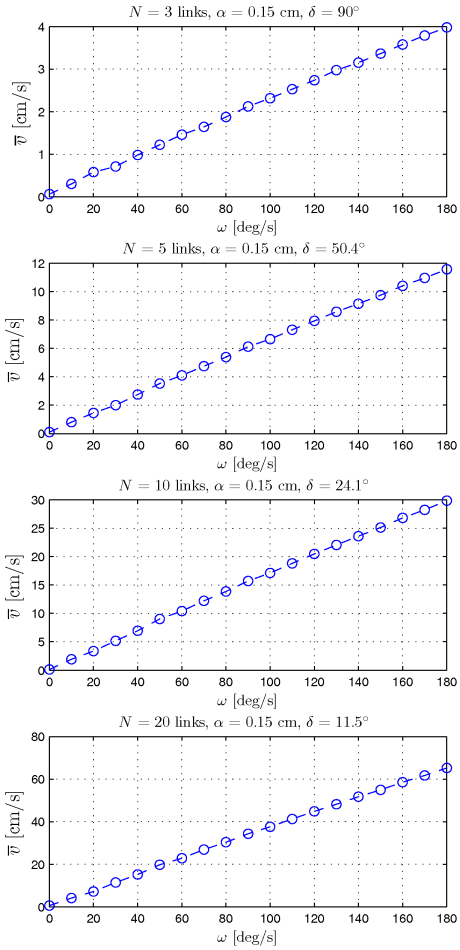
The final result stated in Theorem 7.3 is that the average forward velocity is maximized by the phase shift δ that maximizes the function k_δ. To investigate the validity of this result, we simulated the snake robot with different values of δ to identify the phase shift that produced the highest forward velocity. The simulation results from the simplified model are shown in Fig. 7.7(a), while the simulation results from the complex model with viscous and Coulomb ground friction, respectively, are shown in Fig. 7.7(b). The number of links N and the corresponding values of α and ω are shown at the top of each plot. The δ value that maximizes k_δ is indicated with a vertical dashed line in each plot. Except for the case of Coulomb friction in the complex model with $N = 3$ links, the maximum velocity of each plot in Fig. 7.7 seems to agree well with the δ value that maximizes k_δ. The reason for the disagreement for the case of Coulomb friction with $N = 3$ links is



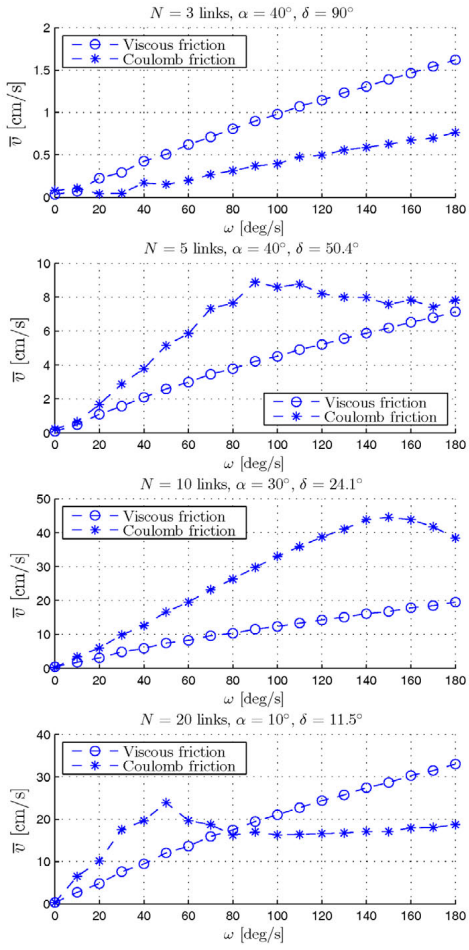
(a) Simulation with the simplified model.

(b) Simulation with the complex model.

Figure 7.5: The forward velocity of the snake robot for different values of α. The number of links N and the corresponding values of ω and δ are shown at the top of each plot.

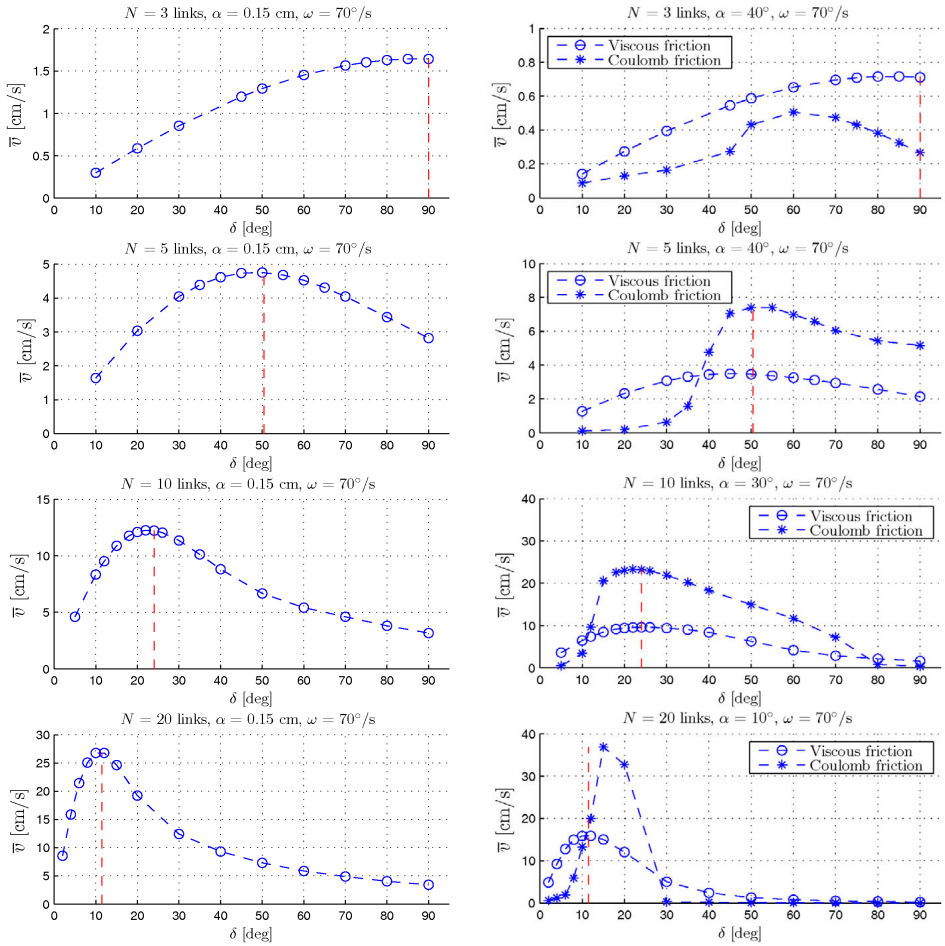


(a) Simulation with the simplified model.



(b) Simulation with the complex model.

Figure 7.6: The forward velocity of the snake robot for different values of ω. The number of links N and the corresponding values of α and δ are shown at the top of each plot.



(a) Simulation with the simplified model.

(b) Simulation with the complex model.

Figure 7.7: The forward velocity of the snake robot for different values of δ. The number of links N and the corresponding values of α and ω are shown at the top of each plot. The vertical dashed line indicates the estimated δ value that, by Theorem 7.3, maximizes the forward velocity.

probably due to nonlinear effects not captured by the simplified model on which Theorem 7.3 is based. Note that the forward velocity in this case is very small.

In summary, the simulation results with both the simplified model (6.24) and the complex model (2.46) agree well with the prediction in Theorem 7.3 concerning the phase shift δ that maximizes the average forward velocity of a snake robot during lateral undulation.

7.8 Experimental Study: Investigation of the Relationships between Gait Parameters and Forward Velocity

In order to provide further support of the validity of Theorem 7.3, we present in this section results from an experimental investigation with the snake robot Wheeko. The design of the robot and the experimental setup (including the camera-based position measurement system) were presented in Chapter 3.

7.8.1 Layout of the Experiment

Controlling the Joints according to Lateral Undulation

During the experiment, the joint reference angles were calculated on an external computer and sent to the snake robot through a wireless Bluetooth connection. The reference angles corresponding to the horizontal joint motion of the robot were calculated according to (7.5) with $N = 10$ links. The reference angles corresponding to the vertical joint motion were set to zero. The joint torque controller given by (4.34) was not employed since accurate torque control is not supported by the servo motors installed in the snake robot. Instead, the joint angles in the robot were controlled according to a proportional controller implemented in the microcontroller of each joint module. In order to show that the joint modules were able to track their joint reference angles, we provide in Fig. 7.8 a plot of the measured and the corresponding reference angles of two arbitrarily chosen joints (joint 3 and joint 7) during a run of lateral undulation with the snake robot. The plot indicates that the tracking of the joint reference angles was satisfactory.

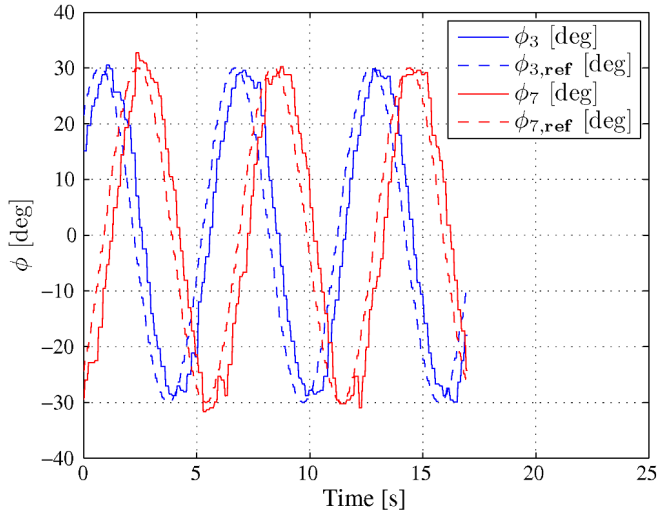


Figure 7.8: A plot of the measured (solid) and the corresponding reference angles (dashed) of joint 3 and joint 7 during lateral undulation.

Calculating the Forward Velocity of the Robot

To visualize how lateral undulation was carried out by the snake robot, we provide in Fig. 7.9 a few screen shots from a video recording of the robot. The position of the robot was recorded by the camera system described in Section 3.4, and the average forward velocity was calculated after each run as the travelled distance divided by the travel time. A typical plot of the measured position of the snake robot from a single run is shown in Fig. 7.10, which shows that the foremost joint module moves from side to side along the x direction, but has a steady increase in the position along the y direction. The markers $\boldsymbol{p}_{\text{start}}$ and $\boldsymbol{p}_{\text{stop}}$ in the plot have been placed near the beginning and near the end of the dataset, respectively, at the approximate centre point of the cyclic sideways motion of the snake. We used the distance between these two markers to represent the distance travelled by the snake robot and calculated the travel time as the difference in sample time between the position measurements corresponding to the two markers. The average forward velocity of the snake robot was then calculated as

$$\overline{v} = \frac{\sqrt{(p_{\text{stop},x} - p_{\text{start},x})^2 + (p_{\text{stop},y} - p_{\text{start},y})^2}}{t_{\text{stop}} - t_{\text{start}}}. \tag{7.32}$$

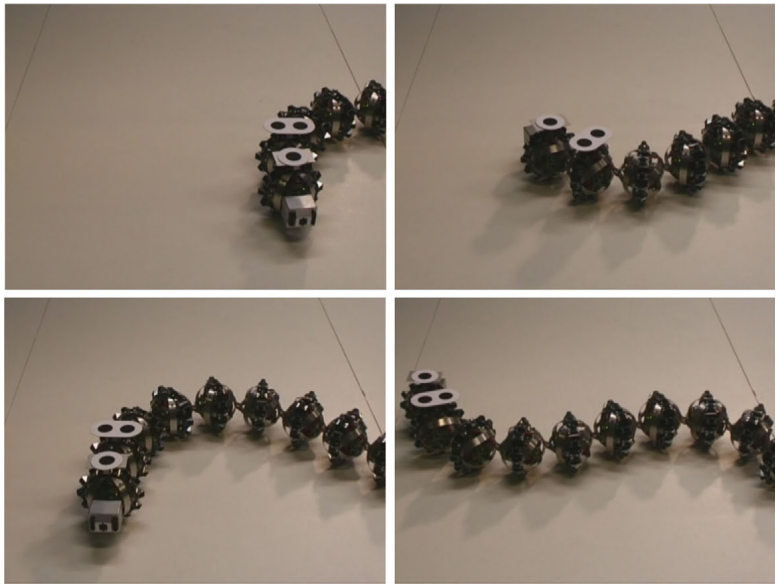


Figure 7.9: The motion of the snake robot during a single run of lateral undulation.

When the duration, $t_{\text{stop}} - t_{\text{start}}$, of a single run of the robot is long, we conjecture that the accuracy of this velocity estimate will be sufficient for investigating the validity of Theorem 7.3. We developed a special software based on *Matlab* in order to easily identify the markers $\boldsymbol{p}_{\text{start}}$ and $\boldsymbol{p}_{\text{stop}}$ in the position plot from each run of the robot.

7.8.2 Experimental results

Relationship between the Forward Velocity and α

Theorem 7.3 states that the average forward velocity of a planar snake robot is proportional to the squared amplitude of the sinusoidal joint motion, α^2. We investigated the validity of this result by running the snake robot with different values of α and calculating the resulting average forward velocity according to (7.32). For each value of α, we ran the snake robot three times in order to get multiple velocity measurements. The remaining controller parameters were set to $\omega = 80°/\text{s}$, $\delta = 25°$ and $\phi_o = 0°$. Fig. 7.11 presents the experimental results together with a dashed line between the average of the three velocities measured for each value of α. The plot clearly shows an

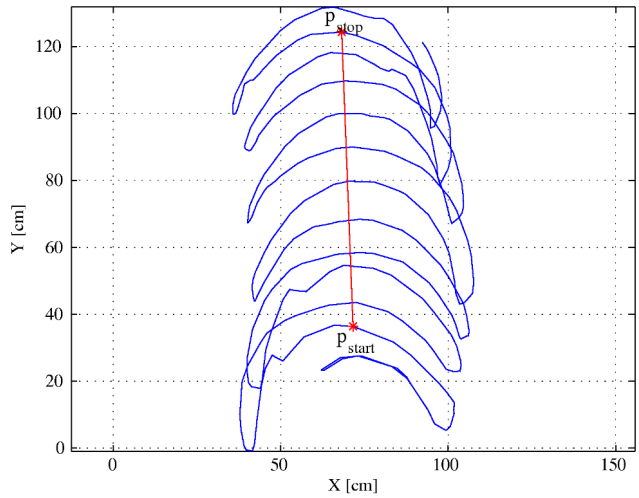


Figure 7.10: A typical plot of the measured position of the snake robot during lateral undulation. The distance between the markers p_{start} and p_{stop} represents the distance travelled by the snake robot.

exponential increase in the forward speed $\overline{v}$ as the amplitude α increases. This is in accordance with Theorem 7.3.

Relationship between the Forward Velocity and ω

Theorem 7.3 states that the average forward velocity of a planar snake robot is proportional to the angular frequency, ω, of the joint motion. This result was investigated by running the snake robot with different values of ω and calculating the resulting average forward velocity according to (7.32). For each value of ω, we ran the snake robot three times in order to get multiple velocity measurements. The remaining controller parameters were set to $\alpha = 30°$, $\delta = 25°$ and $\phi_o = 0°$. Fig. 7.12 presents the experimental results together with a dashed line between the average of the three velocities measured for each value of ω. The plot clearly shows a linear increase in the forward speed $\overline{v}$ as the frequency ω increases. This is in accordance with Theorem 7.3.

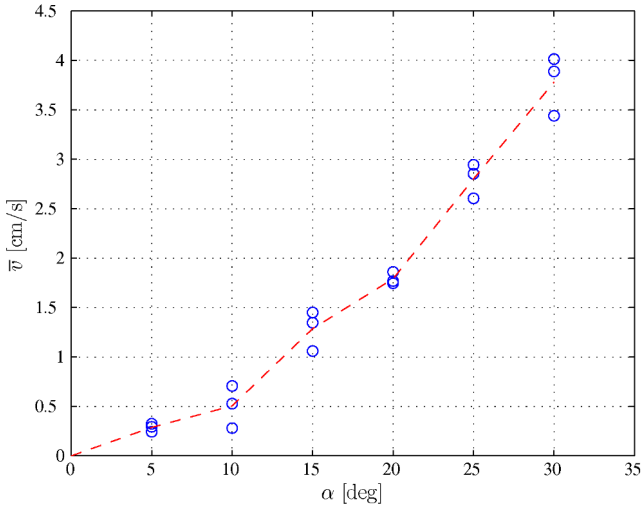


Figure 7.11: The average forward velocity of the snake robot from three trials at different values of α. The remaining controller parameters were set to $\omega = 80°/\text{s}$, $\delta = 25°$ and $\phi_o = 0°$.

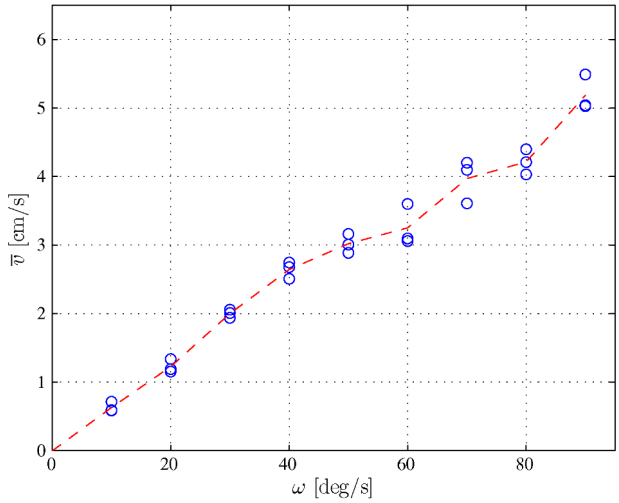


Figure 7.12: The average forward velocity of the snake robot from three trials at different values of ω. The remaining controller parameters were set to $\alpha = 30°$, $\delta = 25°$ and $\phi_o = 0°$.

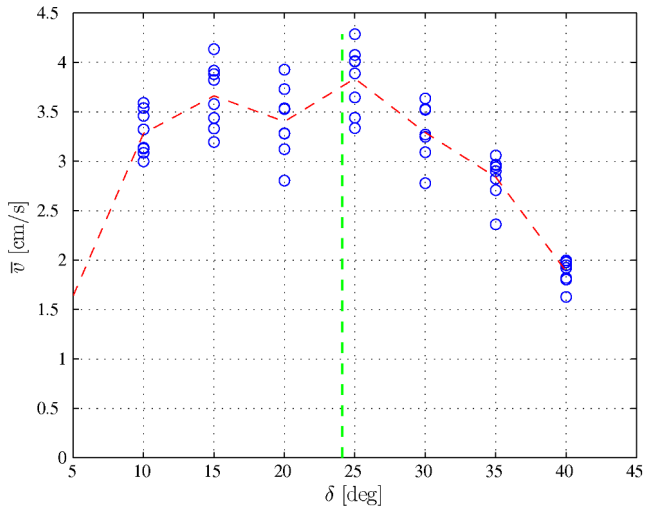


Figure 7.13: The average forward velocity of the snake robot from eight trials at different values of δ. The remaining controller parameters were set to $\alpha = 30°$, $\omega = 80°/\text{s}$ and $\phi_o = 0°$.

Relationship between the Forward Velocity and δ

The final result stated in Theorem 7.3 is that the average forward velocity is maximized by the phase shift δ that maximizes the function k_δ. To investigate the validity of this result, we ran the snake robot with different values of δ to identify the phase shift that produced the highest forward velocity. For each value of δ, we ran the snake robot eight times in order to get multiple velocity measurements. The remaining controller parameters were set to $\alpha = 30°$, $\omega = 80°/\text{s}$ and $\phi_o = 0°$. Fig. 7.13 presents the experimental results together with a dashed line between the average of the eight velocities measured for each value of δ. The δ value that maximizes k_δ for $N = 10$ links is $\delta = 24.1°$, and is indicated with a vertical dashed line in Fig. 7.13. The plot indicates that the phase shift $\delta = 25°$ produced the highest forward velocity. This agrees well with the phase shift $\delta = 24.1°$ that maximizes k_δ. The average velocity of the eight trials at $\delta = 25°$ was slightly below 4 cm/s. In summary, the experimental results indicate that Theorem 7.3 provides a reasonable prediction of the phase shift, δ, that maximizes the average forward velocity of a planar snake robot during lateral undulation.

7.9 Chapter Summary

This chapter is summarized as follows:

- We have in (7.22) presented an averaged model of the velocity dynamics of a planar snake robot during lateral undulation.
- We have shown that the upper bound of the error between the exact and the averaged velocity during lateral undulation is proportional to the inverse of the frequency of the sinusoidal joint motion, ω (see Theorem 7.2).
- We have shown that the average velocity of a snake robot during lateral undulation converges exponentially fast to the steady state velocity given analytically in (7.27) as a function of the gait pattern parameters.
- We have derived fundamental relationships between the gait pattern parameters of lateral undulation and the resulting forward velocity of a snake robot (see Theorem 7.3). In particular, the derived properties state that the average forward velocity of a snake robot 1) is proportional to the squared amplitude of the sinusoidal motion of each joint, 2) is proportional to the angular frequency of the sinusoidal motion of each joint, 3) is proportional to the phase shift function given by (7.18), and 4) is maximized by the phase shift between the joints that also maximizes the phase shift function in (7.18).
- We have presented simulation results that support the theoretical findings.
- We have presented experimental results with the snake robot Wheeko that support the derived relationships between the gait pattern parameters of lateral undulation and the resulting forward velocity of the robot.

Chapter 8

Path Following Control of Snake Robots through a Cascaded Approach

In this chapter, we return to the problem of enabling a snake robot to track a planar path. In Chapter 5, we proposed a straight line path following controller and employed a Poincaré map to prove that the state variables of the snake robot, except for the position along the path, trace out a *locally exponentially stable* periodic orbit during motion along the desired path. The drawback of the analysis based on the Poincaré map, however, is that we are only able to infer conclusions regarding the *local* stability properties in the vicinity of the desired path. Moreover, since the analysis is based on simulating the model of the snake robot, the stability proof is only valid for the given choice of numerical controller parameters.

In order to elude the shortcomings of the previous analysis, this chapter extends the path following controller from Chapter 5 based on the simplified model of the snake robot. Using cascaded systems theory, we prove that the modified path following controller $\mathcal{K}$-exponentially stabilizes the snake robot to any desired straight path. In particular, under the assumption that the forward velocity of the snake robot is nonzero and positive, we show that the model of the snake robot and the controller can be written as a cascaded system where the body shape changes affect the global orientation of the robot, which subsequently affects the cross-track error between the robot and the desired path. The $\mathcal{K}$-exponential stability of the cascaded system guarantees that the cross-track error and the heading of the snake robot with respect to the direction of the path converge to zero.

The performance of the path following controller is investigated through simulations and through experiments with the snake robot Wheeko. The simulations and the experimental results show that the proposed controller successfully steers the snake robot towards and along the desired straight path.

This chapter also considers path following of more general paths. In particular, we propose a waypoint guidance strategy which provably steers a snake robot along a path defined by waypoints interconnected by straight lines. In addition, we describe how the straight line path following controller can be extended to path following of curved paths by employing an approach previously presented in Børhaug (2008) in the context of path following control of marine vessels.

Note that this chapter considers path following, in contrast to trajectory tracking, where the goal is additionally to control the position of the system *along* the path. During path following, we steer the system towards and along the path, but do not consider the temporal position of the system along the path. We will not consider trajectory tracking in this thesis.

Contributions of this Chapter: The first and main contribution of this chapter is the straight line path following controller, which, using cascaded systems theory, is proved to $\mathcal{K}$-exponentially stabilize a snake robot to any desired straight path. To our best knowledge, this is the first time the stability properties of a path following controller for a snake robot without nonholonomic constraints are formally proved. The experimental investigation of the path following controller using the snake robot Wheeko is also considered to be a contribution. The second contribution is the description of how the straight line path following controller can be extended to path following of general curved paths. Finally, the third contribution is the waypoint guidance strategy for steering a snake robot along a path defined by waypoints interconnected by straight lines. Waypoint guidance has, to our best knowledge, not previously been considered for motion control of snake robots. The waypoint guidance strategy builds on the straight line path following controller and represents an operator-friendly framework for steering a snake robot between arbitrary locations on a flat surface.

Organization of this Chapter: Section 8.1 presents some mathematical preliminaries. The straight line path following controller is presented and analysed in Section 8.2, while Section 8.3 describes how

the controller can be extended to path following of general curved paths. The waypoint guidance strategy is proposed in Section 8.4. Simulation results and experimental results concerning the straight line path following controller are presented in Section 8.6, while Section 8.7 presents simulation results concerning the waypoint guidance strategy. Finally, the chapter is summarized in Section 8.8.

Publications: The material in this chapter is based on the journal paper Liljebäck *et al.* (2011*a*) and the conference papers Liljebäck *et al.* (2010*b*), Liljebäck *et al.* (2010*a*), and Liljebäck and Pettersen (2011).

8.1 Mathematical Preliminaries

We begin by presenting some stability concepts that will be employed to analyse the straight line path following controller of the snake robot. The stability concepts make use of class $\mathcal{K}$ and class $\mathcal{KL}$ functions. A function being of class $\mathcal{K}$ basically means that the function is strictly increasing with respect to its argument. A function of class $\mathcal{KL}$ has two arguments, and is strictly increasing with respect to the first argument when the second argument is fixed, and is decreasing with respect to the second argument when the first argument is fixed. Class $\mathcal{K}$ and class $\mathcal{KL}$ functions are formally defined as follows.

Definition 8.1 *(Class $\mathcal{K}$ functions, see Definition 4.2 in Khalil (2002)). A continuous function $\alpha : \mathbb{R}^+ \to \mathbb{R}^+$ is said to belong to class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$.*

Definition 8.2 *(Class $\mathcal{KL}$ functions, see Definition 4.3 in Khalil (2002)). A continuous function $\beta : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ is said to belong to class $\mathcal{KL}$ if, for each fixed s, the mapping $\beta(r, s)$ belongs to class $\mathcal{K}$ with respect to r and, for each fixed r, the mapping $\beta(r, s)$ is decreasing with respect to s and $\beta(r, s) \to 0$ as $s \to \infty$.*

We now present some stability concepts for systems of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}), \tag{8.1}$$

where $\boldsymbol{f} : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \to \mathbb{R}^n$ is piecewise continuous in t and locally Lipschitz in $\boldsymbol{x}$.

Definition 8.3 *(UGAS, see Lemma 4.5 in Khalil (2002)).*
The equilibrium point $\boldsymbol{x} = \mathbf{0}$ of the system (8.1) is uniformly globally asymptotically stable (UGAS) if there exists a class $\mathcal{KL}$ function β such that for any initial state $\boldsymbol{x}(t_0)$

$$\|\boldsymbol{x}(t)\| \leq \beta(\|\boldsymbol{x}(t_0)\|, t - t_0), \quad \forall t \geq t_0 \geq 0. \tag{8.2}$$

A system being UGAS basically means that the state $\boldsymbol{x}$ converges to zero as $t \to \infty$. A special case of UGAS arises when the class $\mathcal{KL}$ function β takes the form of an exponential function as follows.

Definition 8.4 *(UGES, see Definition 4.5 in Khalil (2002)).*
The equilibrium point $\boldsymbol{x} = \mathbf{0}$ of the system (8.1) is uniformly globally exponentially stable (UGES) if there exist positive constants k and λ such that for any initial state $\boldsymbol{x}(t_0)$

$$\|\boldsymbol{x}(t)\| \leq k \|\boldsymbol{x}(t_0)\| e^{-\lambda(t-t_0)}, \quad \forall t \geq t_0 \geq 0. \tag{8.3}$$

A slightly weaker form of stability than exponential stability is $\mathcal{K}$-exponential stability, which is defined as follows.

Definition 8.5 *(Global $\mathcal{K}$-exponential stability, see Definition 2 in Sørdalen and Egeland (1995)).*
The equilibrium point $\boldsymbol{x} = \mathbf{0}$ of the system (8.1) is globally $\mathcal{K}$-exponentially stable if there exist a positive constant λ and a class $\mathcal{K}$ function α such that for any initial state $\boldsymbol{x}(t_0)$

$$\|\boldsymbol{x}(t)\| \leq \alpha(\|\boldsymbol{x}(t_0)\|) e^{-\lambda(t-t_0)}, \quad \forall t \geq t_0 \geq 0. \tag{8.4}$$

As first noted in Lefeber (2000), the following Corollary holds.

Corollary 8.1 *Global $\mathcal{K}$-exponential stability is equivalent to the system being both UGAS and ULES (uniformly locally exponentially stable).*

Remark 8.1 *For simplicity, if the equilibrium point $\boldsymbol{x} = \mathbf{0}$ of a system is UGAS/UGES/globally $\mathcal{K}$-exponentially stable, we often say that the system itself is UGAS/UGES/globally $\mathcal{K}$-exponentially stable.*

Next, we present some stability concepts for cascaded systems of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}_1(t, \boldsymbol{x}) + \boldsymbol{g}(t, \boldsymbol{x}, \boldsymbol{y})\boldsymbol{y}, \tag{8.5}$$
$$\dot{\boldsymbol{y}} = \boldsymbol{f}_2(t, \boldsymbol{y}), \tag{8.6}$$

where $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{y} \in \mathbb{R}^m$, $\boldsymbol{f}_1(t, \boldsymbol{x})$ is continuously differentiable in $(t, \boldsymbol{x})$, and $\boldsymbol{f}_2(t, \boldsymbol{y})$, $\boldsymbol{g}(t, \boldsymbol{x}, \boldsymbol{y})$ are continuous in their arguments and locally Lipschitz in $\boldsymbol{y}$ and $(\boldsymbol{x}, \boldsymbol{y})$, respectively. Many dynamical systems can be written in this cascaded form, where we see that the $\boldsymbol{y}$-dynamics in (8.6) perturbs the $\boldsymbol{x}$-dynamics in (8.5) through the interconnection term $\boldsymbol{g}(t, \boldsymbol{x}, \boldsymbol{y})\boldsymbol{y}$.

Theorem 8.1 *(See Theorem 2 in Panteley and Loria (1998)).*
The cascaded system (8.5), (8.6) is UGAS if the following three assumptions are satisfied:
(A1) *The system $\dot{\boldsymbol{x}} = \boldsymbol{f}_1(t, \boldsymbol{x})$ is UGAS with a radially unbounded Lyapunov function satisfying*

$$\left\| \frac{\partial V}{\partial \boldsymbol{x}} \right\| \|\boldsymbol{x}\| \leq cV(t, \boldsymbol{x}), \quad \forall \|\boldsymbol{x}\| \geq \eta, \tag{8.7}$$

where $c > 0$ and $\eta > 0$ are constants.
(A2) *The function $\boldsymbol{g}(t, \boldsymbol{x}, \boldsymbol{y})$ satisfies*

$$\|\boldsymbol{g}(t, \boldsymbol{x}, \boldsymbol{y})\| \leq \theta_1(\|\boldsymbol{y}\|) + \theta_2(\|\boldsymbol{y}\|)\|\boldsymbol{x}\|, \tag{8.8}$$

where $\theta_1, \theta_2 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ are continuous.
(A3) *The system $\dot{\boldsymbol{y}} = \boldsymbol{f}_2(t, \boldsymbol{y})$ is UGAS and for all $t_0 \geq 0$*

$$\int_{t_0}^{\infty} \|\boldsymbol{y}(t)\| \, dt \leq \kappa(\|\boldsymbol{y}(t_0)\|), \tag{8.9}$$

where the function $\kappa(\cdot)$ is a class $\mathcal{K}$ function.

Lemma 8.1 *(See Lemma 8 in Panteley* et al. *(1998)).*
If in addition to the assumptions in Theorem 8.1 both $\dot{\boldsymbol{x}} = \boldsymbol{f}_1(t, \boldsymbol{x})$ and $\dot{\boldsymbol{y}} = \boldsymbol{f}_2(t, \boldsymbol{y})$ are globally $\mathcal{K}$-exponentially stable, then the cascaded system (8.5), (8.6) is globally $\mathcal{K}$-exponentially stable.

8.2 Straight Line Path Following Control of Snake Robots

In this section, we design and analyse a straight line path following controller for a snake robot described by the simplified model in (6.24).

8.2.1 Control Objective

The control objective is to steer the snake robot so that it converges to and subsequently tracks a straight path while maintaining a heading which is parallel to the path. To this end, we define the global coordinate system so that the global x axis is aligned with the desired straight path. The position of the snake robot along the global y axis, p_y, is thereby the shortest distance from the robot to the desired path (i.e. the cross-track error) and the orientation of the snake robot, θ, is the angle that the robot forms with the desired path. The control objective is thus to regulate p_y and θ to zero.

Since snake robot locomotion is a slow form of robotic mobility, which is generally employed for traversability purposes, we consider it less important to accurately control the forward velocity of the robot. During path following with a snake robot, it therefore makes sense to focus all the control efforts on converging to the path and subsequently progressing along the path at some nonzero forward velocity $v_t \in [V_{\min}, V_{\max}]$, where $V_{\min}$ and $V_{\max}$ represent the boundaries of some positive interval in which we would like the forward velocity to be contained.

From the above discussion, the control problem is to design a feedback control law

$$\boldsymbol{u} = \boldsymbol{u}(t, \boldsymbol{\phi}, \theta, p_y, \boldsymbol{v}_\phi, v_\theta, v_t, v_n) \in \mathbb{R}^{N-1}, \tag{8.10}$$

such that the following control objectives are reached:

$$\lim_{t\to\infty} \; p_y(t) = 0, \tag{8.11}$$

$$\lim_{t\to\infty} \; \theta(t) = 0. \tag{8.12}$$

Remark 8.2 *The path following controller proposed in Chapter 5 did not attempt to suppress the oscillatory behaviour of the heading and position of the snake robot during motion along the desired path. However, since the path following controller proposed in the following is based on the simplified model of the snake robot, circumventing this oscillating behaviour is a more manageable task. In this chapter, we therefore do not just attempt to regulate the cross-track error and the heading so that they oscillate about zero, but also so that they converge to zero.*

8.2.2 Assumptions

Similar to the approach in Chapter 5, we will base the path following controller on the gait pattern lateral undulation, which was defined in (4.32). In Chapter 7, we investigated the velocity dynamics of a snake robot during

lateral undulation. In particular, Theorem 7.3 implies that the forward velocity during lateral undulation oscillates around a positive nonzero average velocity that can be predetermined based on the gait pattern parameters. In other words, when the snake robot conducts lateral undulation, Theorem 7.3 suggests that the forward velocity is contained in some nonzero and positive interval $[V_{\min}, V_{\max}]$ that can be scaled based on the gait pattern parameters. We therefore choose to base the path following controller of the snake robot on the following assumption:

Assumption 8.1 *The snake robot moves by lateral undulation and has a forward velocity which is always nonzero and positive, i.e.* $v_t \in [V_{\min}, V_{\max}]$ $\forall\ t \geq 0$ *where* $V_{\max} \geq V_{\min} > 0$.

Remark 8.3 *In addition to the support from Theorem 7.3, the validity of Assumption 8.1 can be seen by inspecting the equations of motion in (6.24). The dynamics of the forward velocity in (6.24g) contains three terms. As implied by the averaging analysis in Chapter 7, the term* $-\frac{c_2}{Nm}\boldsymbol{\phi}^T \boldsymbol{A}\overline{\boldsymbol{D}}\boldsymbol{v}_\phi$ *is positive and accelerates the robot forward during lateral undulation, while the term* $-\frac{c_1}{m}v_t$ *is the ground friction force. The combined effect of these two terms can never make the forward velocity* v_t *become zero during lateral undulation. This leaves* $\frac{2c_2}{Nm}v_n\overline{\boldsymbol{e}}^T\boldsymbol{\phi}$ *as the only term that can produce a negative forward acceleration that forces* v_t *to zero. This term is negative when the sideways velocity* v_n *and the sum of the joint coordinates* $\overline{\boldsymbol{e}}^T\boldsymbol{\phi}$ *have opposite signs. However, it can be seen from (6.24h) that* v_n *and* $\overline{\boldsymbol{e}}^T\boldsymbol{\phi}$ *will always tend in the same rather than the opposite direction when* $v_t > 0$. *It is therefore unlikely for* v_n *and* $\overline{\boldsymbol{e}}^T\boldsymbol{\phi}$ *to have opposite signs over the long period required to force* v_t *to zero.*

8.2.3 Model Transformation

On the basis of the preceding discussion and Assumption 8.1, we will not control the dynamics of the forward velocity v_t given by (6.24g) and instead treat the forward velocity as a positive parameter satisfying $v_t \in [V_{\min}, V_{\max}]$.

As seen in (6.24f) and (6.24h), the joint coordinates $\boldsymbol{\phi}$ are present in the dynamics of both the angular velocity v_θ and the sideways velocity v_n of the snake robot. This complicates the controller design since the body shape changes will affect both the heading and the sideways motion of the robot. Motivated by Do and Pan (2003), we see that it is possible to remove the effect of $\boldsymbol{\phi}$ on the sideways velocity by a coordinate transformation. In

particular, we move the point that determines the position of the snake robot a distance ϵ along the tangential direction of the robot from the CM to a new location, which is precisely where the body shape changes of the robot (characterized by $\overline{\boldsymbol{e}}^T\boldsymbol{\phi}$) generate a pure rotational motion and no sideways force. This coordinate transformation is illustrated to the left in Fig. 8.1 and is defined as

$$\overline{p}_x = p_x + \epsilon\cos\theta, \tag{8.13a}$$
$$\overline{p}_y = p_y + \epsilon\sin\theta, \tag{8.13b}$$
$$\overline{v}_n = v_n + \epsilon v_\theta, \tag{8.13c}$$

where ϵ is a constant parameter defined as

$$\epsilon = -\frac{2\,(N-1)}{Nm}\frac{c_2}{c_4}. \tag{8.14}$$

With the new coordinates in (8.13), the model (6.24) is transformed into

$$\dot{\boldsymbol{\phi}} = \boldsymbol{v}_\phi, \tag{8.15a}$$
$$\dot{\theta} = v_\theta, \tag{8.15b}$$
$$\dot{\overline{p}}_y = v_t\sin\theta + \overline{v}_n\cos\theta, \tag{8.15c}$$
$$\dot{\boldsymbol{v}}_\phi = -\frac{c_1}{m}\boldsymbol{v}_\phi + \frac{c_2}{m}v_t\boldsymbol{A}\boldsymbol{D}^T\boldsymbol{\phi} + \frac{1}{m}\boldsymbol{D}\boldsymbol{D}^T\boldsymbol{u}, \tag{8.15d}$$
$$\dot{v}_\theta = -c_3 v_\theta + \frac{c_4}{N-1}v_t\overline{\boldsymbol{e}}^T\boldsymbol{\phi}, \tag{8.15e}$$
$$\dot{\overline{v}}_n = Xv_\theta + Y\overline{v}_n, \tag{8.15f}$$

where, by Assumption 8.1, the parameter $v_t \in [V_{\min}, V_{\max}]$ with $V_{\max} \geq V_{\min} > 0$, and where

$$X = \epsilon\left(\frac{c_1}{m} - c_3\right), \tag{8.16a}$$
$$Y = -\frac{c_1}{m}. \tag{8.16b}$$

The two scalar constants X and Y have been introduced in (8.15f) for simplicity of notation in the following sections. Note also that (6.24c) is not included in (8.15) since we do not consider the temporal position of the system along the path during path following.

8.2.4 The Straight Line Path Following Controller

The path following controller of the snake robot consists of two main components. The first component is the gait pattern controller, which propels the snake robot forward according to the gait pattern lateral undulation (as stated in Assumption 8.1). The second component is the heading controller, which steers the snake robot towards and subsequently along the desired path. The two components of the path following controller are now presented.

Gait Pattern Controller

As stated in Definition 4.1, a snake robot moves by lateral undulation by controlling joint $i \in \{1, \ldots, N-1\}$ according to

$$\phi_{i,\text{ref}} = \alpha \sin\left(\omega t + (i-1)\,\delta\right) + \phi_o, \tag{8.17}$$

where α and ω are the amplitude and frequency, respectively, of the sinusoidal joint motion, δ determines the phase shift between the joints, and ϕ_o is a joint offset, which the heading controller will use to control the direction of the locomotion. The average forward velocity v_t^* of the snake robot during straight path motion is given from Theorem 7.3 as

$$v_t^* = \frac{c_2}{2Nc_1}\alpha^2\omega k_\delta, \tag{8.18}$$

where k_δ is a constant parameter determined by the phase shift δ. This relation can be used to choose the gait parameters α, ω, and δ in order to achieve the desired average forward velocity.

As proposed in Section 6.5, we set the actuator forces according to the linearizing control law

$$\boldsymbol{u} = m\left(\boldsymbol{D}\boldsymbol{D}^T\right)^{-1}\left(\overline{\boldsymbol{u}} + \frac{c_1}{m}\dot{\boldsymbol{\phi}} - \frac{c_2}{m}v_t\boldsymbol{A}\boldsymbol{D}^T\boldsymbol{\phi}\right), \tag{8.19}$$

where $\overline{\boldsymbol{u}} \in \mathbb{R}^{N-1}$ is a new set of control inputs. This control law transforms the joint dynamics (8.15d) into $\dot{\boldsymbol{v}}_\phi = \ddot{\boldsymbol{\phi}} = \overline{\boldsymbol{u}}$. In order to make the joints track the joint reference coordinates given by (8.17), we employ the joint controller from Section 4.7.2, i.e. we set the new control input $\overline{\boldsymbol{u}}$ according to

$$\overline{\boldsymbol{u}} = \ddot{\boldsymbol{\phi}}_{\text{ref}} + k_{v_\phi}\left(\dot{\boldsymbol{\phi}}_{\text{ref}} - \dot{\boldsymbol{\phi}}\right) + k_\phi\left(\boldsymbol{\phi}_{\text{ref}} - \boldsymbol{\phi}\right), \tag{8.20}$$

where $k_\phi > 0$ and $k_{v_\phi} > 0$ are scalar controller gains and $\boldsymbol{\phi}_{\text{ref}} \in \mathbb{R}^{N-1}$ are the joint reference coordinates given by (8.17). By introducing the error

variable

$$\widetilde{\boldsymbol{\phi}} = \boldsymbol{\phi} - \boldsymbol{\phi}_{\text{ref}}, \tag{8.21}$$

the joint dynamics given by (8.15a) and (8.15d) can be written in terms of the error dynamics

$$\ddot{\widetilde{\boldsymbol{\phi}}} + k_{v_\phi}\dot{\widetilde{\boldsymbol{\phi}}} + k_\phi \widetilde{\boldsymbol{\phi}} = \mathbf{0}, \tag{8.22}$$

which is clearly *exponentially stable* (see Khalil, 2002). This means that the joint coordinates exponentially track the reference coordinates given by (8.17).

Heading Controller

In order to steer the snake robot towards the desired straight path, we employ the Line-of-Sight (LOS) guidance law that was also employed in Chapter 5, which is defined as

$$\theta_{\text{ref}} = -\arctan(\frac{\overline{p}_y}{\Delta}), \tag{8.23}$$

where $\overline{p}_y$ is the cross-track error and $\Delta > 0$ is a design parameter referred to as the *look-ahead distance*. This LOS guidance law is commonly used during e.g. path following control of marine surface vessels (see e.g. Fossen, 2002; Fredriksen and Pettersen, 2006). As illustrated to the right in Fig. 8.1, the LOS angle θ_{ref} corresponds to the orientation of the snake robot when it is headed towards the point located a distance Δ ahead of the snake robot along the desired path. The value of Δ is important since it determines the rate of convergence to the desired path.

Based on Property 4.7, we will use the joint offset coordinate ϕ_o in (8.17) to ensure that the heading of the snake robot θ tracks the LOS angle given by (8.23). Motivated by Fredriksen and Pettersen (2006) and Pettersen and Lefeber (2001), we conjecture that making θ track the LOS angle θ_{ref} will make the snake converge to the desired path and subsequently follow the path with its heading parallel to the path. In other words, we conjecture that a control law that makes θ track θ_{ref} will fulfill the control objectives (8.11) and (8.12). To derive the control law for ϕ_o, we first rewrite the dynamics of v_θ given by (8.15e) with the new coordinates $\widetilde{\boldsymbol{\phi}}$ in (8.21), which gives the dynamics of v_θ as a function of the joint reference coordinates given by (8.17). From (8.21), we have that $\boldsymbol{\phi} = \boldsymbol{\phi}_{\text{ref}} + \widetilde{\boldsymbol{\phi}}$. Using (8.17), we can therefore rewrite (8.15e) as

$$\dot{v}_\theta = -c_3 v_\theta + c_4 v_t \phi_o + \frac{c_4}{N-1} v_t \left(\sum_{i=1}^{N-1} \alpha \sin(\omega t + (i-1)\,\delta) + \overline{\boldsymbol{e}}^T \widetilde{\boldsymbol{\phi}} \right). \tag{8.24}$$

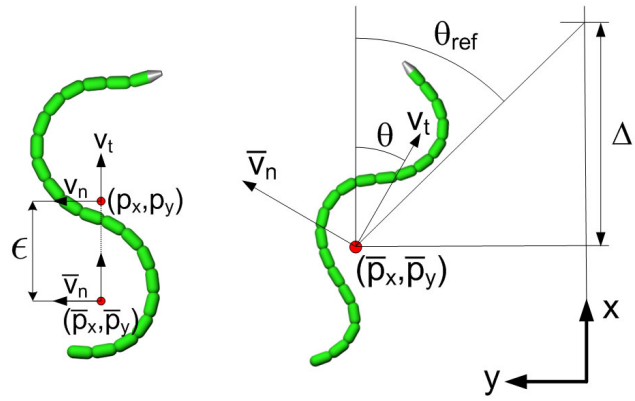


Figure 8.1: Left: The coordinate transformation of the snake robot. Right: The Line-of-Sight (LOS) guidance law.

Consequently, choosing ϕ_o as

$$\phi_o = \frac{1}{c_4 v_t}\left(\ddot{\theta}_{\text{ref}} + c_3\dot{\theta}_{\text{ref}} - k_\theta(\theta - \theta_{\text{ref}}) - \frac{c_4}{N-1}v_t\sum_{i=1}^{N-1}\alpha\sin(\omega t + (i-1)\,\delta)\right), \tag{8.25}$$

where $k_\theta > 0$ is a scalar controller gain, enables us to write the dynamics of the heading angle θ, which is given by (8.15b) and (8.15e), in terms of the error dynamics

$$\ddot{\widetilde{\theta}} + c_3\dot{\widetilde{\theta}} + k_\theta\widetilde{\theta} = \frac{c_4}{N-1}v_t\overline{\boldsymbol{e}}^T\widetilde{\boldsymbol{\phi}}, \tag{8.26}$$

where we have introduced the error variable

$$\widetilde{\theta} = \theta - \theta_{\text{ref}}. \tag{8.27}$$

Remark 8.4 *The joint coordinate offset in (8.25) depends on the inverse of the forward velocity v_t. This does not represent a problem since, by Assumption 8.1, the forward velocity is always nonzero. When implementing the path following controller, this issue can be avoided by activating the controller* after *the snake robot has obtained a positive forward velocity.*

Remark 8.5 *The error dynamics of the joints in (8.22) and the error dynamics of the heading in (8.26) represent a cascaded system. In particular, the system (8.22) perturbs the system (8.26) through the interconnection*

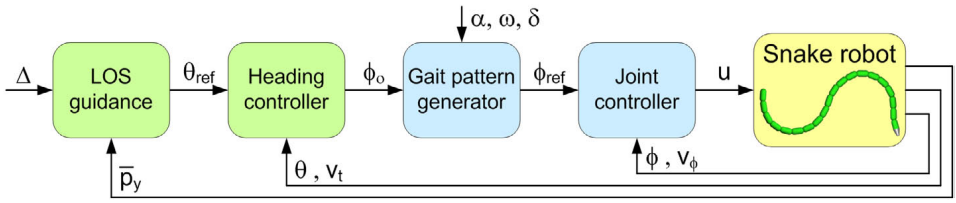


Figure 8.2: The structure of the straight line path following controller.

term $\frac{c_4}{N-1} v_t \bar{\boldsymbol{e}}^T \tilde{\boldsymbol{\phi}}$. *Using cascaded systems theory, it will be shown in Section 8.2.6 that the origin of this cascaded system is globally* $\mathcal{K}$*-exponentially stable.*

We have now presented the complete path following controller of the snake robot. The structure of the complete controller is summarized in Fig. 8.2.

8.2.5 The Stability Properties of the Path Following Controller

Based on the guidance and control laws presented in the previous subsection, we now state the main result concerning the straight line path following controller. The result specifies a lower bound on the look-ahead distance Δ employed in (8.23). This bound is given a physical interpretation in Remark 8.7 below, and is derived in the proof presented in Section 8.2.6.

Theorem 8.2 *Consider a planar snake robot described by the model (8.15) and suppose that Assumption 8.1 is satisfied. If the look-ahead distance* Δ *of the LOS guidance law (8.23) is chosen such that*

$$\Delta > \frac{|X|}{|Y|} \left(1 + \frac{V_{\max}}{V_{\min}} \right), \tag{8.28}$$

then the path following controller defined by (8.17), (8.19), (8.20), (8.23), and (8.25) guarantees that the control objectives (8.11) and (8.12) are achieved for any set of initial conditions satisfying $v_t \in [V_{\min}, V_{\max}]$.

Proof. The proof of this theorem is given in Section 8.2.6. ■

Remark 8.6 *Theorem 8.2 does not specify the boundary values $V_{\min}$ and $V_{\max}$ of the interval in which the forward velocity v_t is contained. By Assumption 8.1, however, there exists a positive interval that contains v_t for all time $t \geq 0$. In practice, conservative values for these boundary values can be chosen, but in order to achieve a tighter bound on Δ, we would like to specify $V_{\min}$ and $V_{\max}$ as a function of the gait pattern parameters α, ω, δ, and ϕ_o. This remains a topic of future work.*

Remark 8.7 *The lower bound on the look-ahead distance Δ in (8.28) ensures that the sideways velocity $\overline{v}_n$ of the snake robot in (8.15f) is well behaved under the perturbations from the angular velocity v_θ. In particular, the magnitude of v_θ during convergence to the desired path is determined by the look-ahead distance Δ, i.e. the robot rotates fast when Δ is small (and vice versa). We see from (8.15f) that v_θ only has a small influence on $\overline{v}_n$ when $|X| \ll |Y|$, which means that we then can allow the magnitude of v_θ to be large, i.e. Δ can be small. Similarly, v_θ has a great influence on $\overline{v}_n$ when $|X| \gg |Y|$, which means that the magnitude of v_θ must be restricted, i.e. Δ must be large. These conditions are directly reflected by the lower bound in (8.28).*

Remark 8.8 *As explained in Remark 6.1 in Section 6.1, the assumptions underlying the simplified model are only valid as long as the link angles with respect to the forward direction are limited. The stability result in Theorem 8.2 is therefore claimed only for snake robots conducting lateral undulation with limited link angles.*

8.2.6 Proof of Theorem 8.2

We will prove Theorem 8.2 in three steps. In the first step, we show that the complete system, including the path following controller, can be written as a cascaded system. In the second step, we prove stability of the nominal systems in the cascade. Finally, we derive bounds on the interconnection terms between the nominal systems, which, by Theorem 8.1 and Lemma 8.1, allow us to conclude stability of the complete cascaded system. We will follow the steps of a similar proof presented in Pavlov *et al.* (2007).

We begin by rewriting the dynamics of the cross-track error $\overline{p}_y$ and the sideways velocity $\overline{v}_n$ in terms of the heading error $\widetilde{\theta}$. From (8.27) and (8.23) we have that

$$\theta = -\arctan(\frac{\overline{p}_y}{\Delta}) + \widetilde{\theta}. \tag{8.29}$$

By using the relations

$$\sin(-\arctan(\tfrac{\overline{p}_y}{\Delta})) = -\tfrac{\overline{p}_y}{\sqrt{\overline{p}_y^2+\Delta^2}}, \quad \cos(-\arctan(\tfrac{\overline{p}_y}{\Delta})) = \tfrac{\Delta}{\sqrt{\overline{p}_y^2+\Delta^2}}, \tag{8.30}$$

it can be verified that (8.15c) can be written in terms of the heading error $\widetilde{\theta}$ as

$$\dot{\overline{p}}_y = -\frac{v_t}{\sigma}\overline{p}_y + \frac{\Delta}{\sigma}\overline{v}_n + \gamma\widetilde{\theta}, \tag{8.31}$$

where

$$\sigma = \sqrt{\overline{p}_y^2 + \Delta^2}, \tag{8.32}$$

$$\gamma = \frac{\sin\widetilde{\theta}}{\widetilde{\theta}}\frac{\left(v_t\Delta + \overline{v}_n\overline{p}_y\right)}{\sigma} + \frac{1-\cos\widetilde{\theta}}{\widetilde{\theta}}\frac{\left(v_t\overline{p}_y - \overline{v}_n\Delta\right)}{\sigma}. \tag{8.33}$$

Through similar manipulations, we can rewrite (8.15f) in the new coordinates as

$$\dot{\overline{v}}_n = \frac{X\Delta v_t}{\sigma^3}\overline{p}_y + \left(Y - \frac{X\Delta^2}{\sigma^3}\right)\overline{v}_n - \frac{X\Delta}{\sigma^2}\gamma\widetilde{\theta} + X\dot{\widetilde{\theta}}. \tag{8.34}$$

Collecting the error variables as

$$\boldsymbol{\eta} = \begin{bmatrix}\widetilde{\boldsymbol{\phi}} \\ \dot{\widetilde{\boldsymbol{\phi}}}\end{bmatrix} \in \mathbb{R}^{2N-2}, \quad \boldsymbol{\xi} = \begin{bmatrix}\widetilde{\theta} \\ \dot{\widetilde{\theta}}\end{bmatrix} \in \mathbb{R}^2, \tag{8.35}$$

and using (8.22), (8.26), (8.31), and (8.34), the model of the snake robot (8.15) during path following can be written as

$$\begin{bmatrix}\dot{\overline{p}}_y \\ \dot{\overline{v}}_n\end{bmatrix} = \boldsymbol{C}(\overline{p}_y)\begin{bmatrix}\overline{p}_y \\ \overline{v}_n\end{bmatrix} + \boldsymbol{H}_\xi(\overline{p}_y, \overline{v}_n, \boldsymbol{\xi})\boldsymbol{\xi}, \tag{8.36a}$$

$$\dot{\boldsymbol{\xi}} = \begin{bmatrix}0 & 1 \\ -k_\theta & -c_3\end{bmatrix}\boldsymbol{\xi} + \boldsymbol{H}_\eta\boldsymbol{\eta}, \tag{8.36b}$$

$$\dot{\boldsymbol{\eta}} = \begin{bmatrix}\boldsymbol{0}_{(N-1)\times(N-1)} & \boldsymbol{I}_{N-1} \\ -k_\phi\boldsymbol{I}_{N-1} & -k_{v_\phi}\boldsymbol{I}_{N-1}\end{bmatrix}\boldsymbol{\eta}, \tag{8.36c}$$

where

$$\boldsymbol{H}_\eta = \begin{bmatrix}\boldsymbol{0}_{1\times(N-1)} & \boldsymbol{0}_{1\times(N-1)} \\ \frac{c_4}{N-1}v_t\overline{\boldsymbol{e}}^T & \boldsymbol{0}_{1\times(N-1)}\end{bmatrix} \in \mathbb{R}^{2\times(2N-2)}, \tag{8.37}$$

$$\boldsymbol{H}_\xi(\overline{p}_y, \overline{v}_n, \boldsymbol{\xi}) = \begin{bmatrix}\gamma & 0 \\ -\frac{X\Delta}{\sigma^2}\gamma & X\end{bmatrix} \in \mathbb{R}^{2\times2}, \tag{8.38}$$

$$\boldsymbol{C}(\overline{p}_y) = \begin{bmatrix}-\frac{v_t}{\sigma} & \frac{\Delta}{\sigma} \\ \frac{X\Delta v_t}{\sigma^3} & \left(Y - \frac{X\Delta^2}{\sigma^3}\right)\end{bmatrix} \in \mathbb{R}^{2\times2}. \tag{8.39}$$

The system (8.36) is a cascaded system. In particular, the $\boldsymbol{\eta}$-dynamics in (8.36c) perturbs the $\boldsymbol{\xi}$-dynamics in (8.36b) through the interconnection term $\boldsymbol{H}_\eta \boldsymbol{\eta}$, and the $\boldsymbol{\xi}$-dynamics perturbs the $(\overline{p}_y, \overline{v}_n)$-dynamics in (8.36a) through the interconnection term $\boldsymbol{H}_\xi(\overline{p}_y, \overline{v}_n, \boldsymbol{\xi})\boldsymbol{\xi}$.

We now investigate the stability of the nominal systems of the cascade, i.e. all parts of (8.36) except the interconnection terms. The origin $\boldsymbol{\eta} = \boldsymbol{0}$ of the linear system (8.36c) and the origin $\boldsymbol{\xi} = \boldsymbol{0}$ of the linear nominal system in (8.36b) are UGES (see Definition 8.4) since the system matrices clearly are Hurwitz (Khalil, 2002) for k_θ, c_3, k_ϕ, $k_{v_\phi} > 0$. The nominal system of (8.36a) is given by

$$\begin{bmatrix} \dot{\overline{p}}_y \\ \dot{\overline{v}}_n \end{bmatrix} = \boldsymbol{C}(\overline{p}_y) \begin{bmatrix} \overline{p}_y \\ \overline{v}_n \end{bmatrix}, \tag{8.40}$$

and has the stability properties established by the following two Lemmas.

Lemma 8.2 *Under the conditions of Theorem 8.2, the origin of the system (8.40) is UGAS with a quadratic Lyapunov function.*

Proof. The proof of this lemma has previously been presented in Pavlov *et al.* (2007) and is included in Appendix A for completeness. ■

Lemma 8.3 *Under the conditions of Theorem 8.2, the origin of the system (8.40) is globally $\mathcal{K}$-exponentially stable.*

Proof. The proof of this lemma is presented in Appendix B. ■

Since exponential stability implies $\mathcal{K}$-exponential stability, we can conclude that all nominal systems of the cascade (8.36) are globally $\mathcal{K}$-exponentially stable. Next, we derive bounds on the interconnection terms in the cascade. The induced 2-norm of the matrix $\boldsymbol{H}_\eta$ satisfies (see Appendix A in Khalil, 2002)

$$\|\boldsymbol{H}_\eta\|_2 \leq \sqrt{2N-2} \max_j \sum_{i=1}^{2} \{\boldsymbol{H}_\eta\}_{ij} \leq \frac{\sqrt{2} c_4 V_{\max}}{\sqrt{N-1}}, \tag{8.41}$$

while the induced 2-norm of the matrix $\boldsymbol{H}_\xi(\overline{p}_y, \overline{v}_n, \boldsymbol{\xi})$ satisfies

$$\begin{aligned} \|\boldsymbol{H}_\xi\|_2 \leq \sqrt{2} \max_j \textstyle\sum_{i=1}^{2} \{\boldsymbol{H}_\xi\}_{ij} &\leq \sqrt{2} \max\left(|\gamma| + \tfrac{|X|\Delta}{\sigma^2}|\gamma|, |X| \right) \\ &\leq \sqrt{2}\left(|\gamma| + \tfrac{|X|\Delta}{\sigma^2}|\gamma| + |X| \right). \end{aligned} \tag{8.42}$$

The function γ given by (8.33) is bounded according to

$$\begin{aligned}\gamma &\leq \left|\frac{\sin\widetilde{\theta}}{\widetilde{\theta}}\right| \frac{V_{\max}\Delta+|\overline{v}_n||\overline{p}_y|}{\sigma} + \left|\frac{1-\cos\widetilde{\theta}}{\widetilde{\theta}}\right| \frac{V_{\max}|\overline{p}_y|+|\overline{v}_n|\Delta}{\sigma} \\ &\leq \frac{V_{\max}\Delta}{\sigma} + \frac{|\overline{v}_n||\overline{p}_y|}{\sigma} + \frac{V_{\max}|\overline{p}_y|}{\sigma} + \frac{|\overline{v}_n|\Delta}{\sigma} \leq 2V_{\max} + 2|\overline{v}_n|.\end{aligned} \tag{8.43}$$

By inserting (8.43) into (8.42), it is straightforward to verify that

$$\|\boldsymbol{H}_\xi\|_2 \leq \mathcal{F}_1 + \mathcal{F}_2 \left\| \begin{bmatrix} \overline{p}_y \\ \overline{v}_n \end{bmatrix} \right\|_2, \tag{8.44}$$

where

$$\mathcal{F}_1 = \sqrt{2}\left(2V_{\max}\left(1+\frac{|X|}{\Delta}\right)+|X|\right), \tag{8.45}$$

$$\mathcal{F}_2 = 2\sqrt{2}\left(1+\frac{|X|}{\Delta}\right). \tag{8.46}$$

We are now ready to apply Theorem 8.1 to the cascaded system (8.36). We first consider the cascade of (8.36b) and (8.36c), for which it is straightforward to verify that Assumptions A1 and A3 of Theorem 8.1 are satisfied since the system (8.36c) and the nominal system of (8.36b) are both UGES (see Definition 8.4). Furthermore, Assumption A2 is trivially satisfied since $\|\boldsymbol{H}_\eta\|_2$ is bounded by the constant derived in (8.41). The cascaded system (8.36b), (8.36c) is therefore UGAS and, by Lemma 8.1, also globally $\mathcal{K}$-exponentially stable.

Next, we consider the cascade of (8.36a) and (8.36b), for which Assumption A1 of Theorem 8.1 is satisfied since, by Lemma 8.2, the nominal system of (8.36a) is UGAS with a quadratic Lyapunov function. Furthermore, it follows directly from (8.44) that Assumption A2 is satisfied. Finally, since the perturbing system (8.36b) is globally $\mathcal{K}$-exponentially stable, Assumption A3 is satisfied since the bound in Assumption A3 is easily shown to hold for any $\mathcal{K}$-exponentially stable system by integrating both sides of (8.4) from t_0 to ∞. The cascaded system (8.36a), (8.36b) is therefore UGAS and, by Lemma 8.1, also globally $\mathcal{K}$-exponentially stable since the nominal system of (8.36a) and the perturbing system (8.36b) are both globally $\mathcal{K}$-exponentially stable.

In summary, the complete cascaded system (8.36) is globally $\mathcal{K}$-exponentially stable. This means that $\overline{p}_y(t) \to 0$ and $\widetilde{\theta}(t) \to 0$, which, by (8.29), implies that $\theta(t) \to 0$, which means that control objective (8.12) is achieved. It subsequently follows from (8.13b) that $p_y(t) \to 0$, which means that control objective (8.11) is achieved. This completes the proof of Theorem 8.2.

Remark 8.9 *Any gait pattern controller that exponentially stabilizes the error variable (8.21), i.e. not just the joint controller proposed in (8.19) and (8.20), makes the complete cascaded system globally $\mathcal{K}$-exponentially stable. This is a nice feature of cascaded systems theory.*

8.3 Path Following Control of Snake Robots along Curved Paths

The straight line path following controller presented in Section 8.2.4 can be extended to path following of general curved paths by following an approach presented in Børhaug (2008) in the context of path following control of marine vessels. In this section, we describe how this extension can be achieved.

8.3.1 Comments on the Curved Path Following Controller

The stability proof regarding the convergence of the snake robot to the curved path will not be detailed in this thesis since the proof is developed by following similar steps as the approach presented in Børhaug (2008). Moreover, we do not consider curved path following control to be particularly relevant to our long-term goal of achieving snake robot locomotion in unstructured environments. In particular, we would like our control design efforts for snake robots on flat surfaces to be extendable to the case where the environment is unstructured and no longer flat. The idea of requiring a snake robot to follow a nice and smooth curved path in an unknown and unstructured environment (i.e. where external contact forces will influence the motion significantly) seems unrealistic. A more realistic and applicable approach for path following of general paths is, in our opinion, to characterize the general path in terms of waypoints interconnected by straight lines, where the straight line reference path between each waypoint basically tells the robot to take the shortest possible path to each waypoint. We will elaborate this approach in Section 8.4, and will further develop the approach in Chapter 13 in order to enable snake robot locomotion along general paths in an unstructured environment.

8.3.2 The Curved Path Following Controller

The desired curved path that the snake robot should follow is a continuously differentiable curve denoted by $\mathcal{C}$ (see Fig. 8.3). The idea behind the controller is to steer the snake robot towards a *virtual* particle that moves

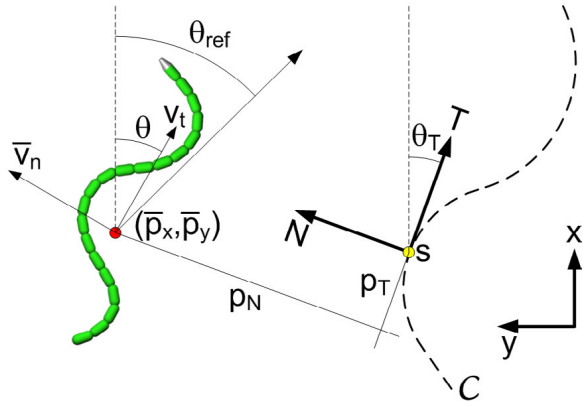


Figure 8.3: The guidance strategy for path following of general curved paths.

along the path. The distance travelled by the particle along the curve is denoted by s, which means that $\dot{s}$ is the instantaneous speed of the particle along the curve. Furthermore, we define a moving coordinate frame with axes denoted by T and N such that the origin of the frame coincides with the particle and the T axis is always tangential to the curve. This is called a *Serret-Frenet* coordinate frame (see e.g. Egeland and Gravdahl, 2002). As visualized in Fig. 8.3, the angle of the T axis with respect to the global x axis is denoted by θ_T and the position of the snake robot in the T-N frame is denoted by (p_T, p_N).

Since the goal is to make the snake robot converge to and follow the desired path $\mathcal{C}$, we state the control objective as

$$\lim_{t\to\infty} p_T(t) = 0, \quad \lim_{t\to\infty} p_N(t) = 0. \tag{8.47}$$

Based on the curved path following controller for marine vessels proposed in Børhaug (2008), we claim that control objective (8.47) is achieved by steering the heading θ of the snake robot according to the guidance law

$$\theta_{\text{ref}} = \theta_T - \arctan(\frac{\overline{v}_n}{v_t}) - \arctan(\frac{p_N}{\sqrt{\Delta^2 + p_T^2}}), \tag{8.48}$$

and updating the position of the virtual particle along the curve according

to

$$\dot{s} = U\frac{\sqrt{\Delta^2 + p_T^2} + p_T}{\sqrt{\Delta^2 + p_T^2 + p_N^2}},\ U = \sqrt{v_t^2 + \overline{v}_n^2}. \tag{8.49}$$

The curved path following controller is, in other words, defined by (8.17), (8.19), (8.20), and (8.25), where θ_{ref} is given by (8.48) and where s is updated according to (8.49). The proof of the achievement of control objective (8.47) can be developed based on the results in Børhaug (2008) by taking into account a few differences between the model of the snake robot and the model of the vessel considered in Børhaug (2008). The proof also requires us to make specific assumptions regarding the minimum forward velocity of the robot, the maximum forward acceleration of the robot, the maximum curvature of the path, and the look-ahead distance Δ.

8.4 Waypoint Guidance Control of Snake Robots

In this section, we employ the straight line path following controller presented in Section 8.2 in order to propose a guidance strategy for steering a snake robot between a set of reference locations, or *waypoints*, in the environment of the robot. Waypoint guidance is, in our opinion, an approach which is well suited for general motion control of snake robots. The waypoint guidance strategy proposed in the following represents an operator-friendly framework for steering a snake robot between arbitrary locations on a flat surface.

8.4.1 Description of the Approach

Future applications of snake robots will generally involve motion in challenging and unstructured environments where the aim is to bring sensors and/or tools to a single or several specified target location(s). In these situations, the exact path taken by the robot as it moves towards the target(s) is generally of less interest as long as the robot reaches the target(s) within a reasonable amount of time. Specifying the motion in terms of waypoints supports this target-oriented control approach. Waypoint guidance is a commonly used approach for control of e.g. marine surface vessels (see e.g. Fossen, 2002), but has, to our best knowledge, not been considered for motion control of snake robots.

In accordance with the target-oriented approach discussed above, we choose to interconnect the waypoints by straight lines and employ the path

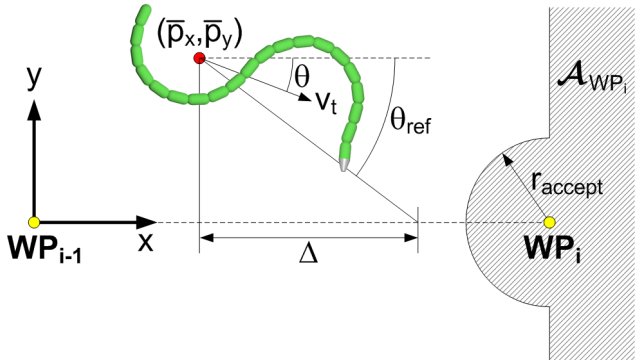


Figure 8.4: The waypoint guidance strategy.

following controller presented in Section 8.2 in order to steer the snake robot towards the straight line leading to the next waypoint. This approach is illustrated in Fig. 8.4. As argued in Section 8.3.1, the reason for considering straight lines instead of curved paths is our long-term goal of also employing the guidance strategy in unstructured environments. The idea of requiring a snake robot to follow a nice and smooth curved path in an unknown and unstructured environment seems unrealistic, while a straight line reference path between each waypoint basically tells the robot to take the shortest possible path to the next waypoint.

A common rule for switching between the waypoints is to proceed towards the next waypoint as soon as the position of the system enters inside an *acceptance circle* enclosing the current waypoint (see Fossen, 2002). In the present work, we propose that the acceptance circle is replaced by an *acceptance region* composed of an acceptance circle and also the right half plane of a coordinate system with origo in the current waypoint and x axis pointing away from the previous waypoint (see illustration in Fig. 8.4). With this definition, we are guaranteed that the robot will reach the acceptance region of the current waypoint no matter how the waypoints are defined. With only acceptance circles enclosing each waypoint, there would be the risk that the robot misses a waypoint which is placed too close to the previous waypoint, which would make the robot proceed indefinitely along the path away from the waypoint that was missed. Note that although the acceptance region is infinitely large, the path following controller presented in Section 8.2 guarantees rapid convergence to the straight path between two waypoints.

8.4.2 The Waypoint Guidance Strategy

In the following, we formalize the guidance strategy described in the previous subsection.

Definition 8.6 ***Waypoint.***
A waypoint *is a reference position along the path of the snake robot. There are k waypoints and the ith waypoint is denoted by WP_i, where $i \in \{1, \ldots, k\}$.*

Definition 8.7 ***Acceptance region.***
The acceptance region *of WP_i, denoted $\mathcal{A}_{WP_i}$, is the union of all points inside a circle centred in WP_i with radius r_{accept} and the right half plane of a coordinate system with origo in WP_i and x axis aligned with the vector from WP_{i-1} to WP_i.*

Definition 8.8 ***The waypoint guidance problem.***
Given a set of k waypoints $WP_1, \ldots, WP_k$, the waypoint guidance problem *is the task of steering the position of the snake robot into the acceptance region of each of the waypoints $WP_1, \ldots, WP_k$ in consecutive order.*

In accordance with the above definitions and the description in the previous subsection, we now present the waypoint guidance strategy for the snake robot.

Algorithm 8.1 ***The waypoint guidance strategy.***

1. Define the initial position of the snake robot as WP_0.
2. Repeat for all $i \in \{0, \ldots, k-1\}$:
   (a) Move the origin of the global frame to WP_i and orient the global x axis towards WP_{i+1}.
   (b) Conduct straight line path following according to the controller from Section 8.2 until $(p_x, p_y) \in \mathcal{A}_{WP_{i+1}}$.

The guidance strategy proposed in Algorithm 8.1 satisfies the following result:

Theorem 8.3 *The waypoint guidance problem presented in Definition 8.8 is solved by Algorithm 8.1 for a planar snake robot described by the model (8.15) under the conditions of Theorem 8.2.*

Proof. Given any waypoint WP_i that the snake robot is crawling towards, where $i \in \{1, \ldots, k\}$, we are ensured by Definition 8.7 that the desired straight path of the robot points into the acceptance region of WP_i. By Theorem 8.2, the snake robot will eventually reach the desired straight path and progress along the path indefinitely, which means that the position of the snake robot will eventually reach the acceptance region of WP_i. This completes the proof. ■

8.5 Simulation Study: The Performance of the Straight Line Path Following Controller

In this section, we present simulation results that illustrate the performance of the straight line path following controller proposed in Section 8.2.

8.5.1 Simulation Parameters

The model of the snake robot (6.24) and the path following controller defined by (8.17), (8.19), (8.20), (8.23), and (8.25) were implemented and simulated in *Matlab R2008b* on a laptop running *Windows XP*. The model dynamics was calculated using the *ode45* solver in Matlab with a relative and absolute error tolerance of 10^{-6}.

We considered a snake robot with $N = 10$ links of length $l = 0.14$ m and mass $m = 1$ kg. Furthermore, we chose the friction coefficients as $c_1 = 0.45$, $c_2 = 3$, $c_3 = 0.5$ and $c_4 = 20$, the controller gains as $k_\phi = 20$, $k_{v_\phi} = 5$, and $k_\theta = 0.05$, and calculated the coordinate transformation distance according to (8.14) as $\epsilon = -27$ cm. The gait parameters were $\alpha = 0.1$ m, $\omega = 70°/\text{s}$, and $\delta = 40°$, which by (8.18) corresponds to the average forward velocity $v_t^* = \frac{c_2}{2Nc_1}\alpha^2\omega k_\delta = 0.1$ m/s. By making the conjecture that the forward velocity will always be contained in the interval $v_t \in [V_{\min}, V_{\max}] = [0.5v_t^*, 2v_t^*] = [0.05 \text{ m/s}, 0.2 \text{ m/s}]$, the lower bound on the look-ahead distance Δ is given by (8.28) as $\Delta > 0.15$ m. During the simulations, we chose the look-ahead distance equal to the length of the snake robot, i.e. $\Delta = 1.4$ m, which is well above the estimated lower limit.

The derivatives $\dot{\phi}_o$, $\ddot{\phi}_o$, $\dot{\theta}_{\text{ref}}$, and $\ddot{\theta}_{\text{ref}}$, which are needed for the calculation of the control input in (8.20) and (8.25), were obtained by passing ϕ_o and θ_{ref} through a 3rd order low-pass filtering reference model (see Appendix C.2). The parameters of the reference model were set to $\omega = \pi/2$ and $\zeta = 1$.

The initial state of the snake robot was chosen as $\boldsymbol{\phi} = \mathbf{0}°$, $\theta = 90°$,

$p_x = 0$ m, $p_y = 1$ m, $\boldsymbol{v}_\phi = \mathbf{0}^\circ$/s, $v_\theta = 0^\circ$/s, $v_t = 0.1$ m/s, and $v_n = 0$ m/s, i.e. the snake robot was initially oriented along the global y axis and located 1 m away from the x axis with an initial forward velocity of 0.1 m/s, i.e. moving away from the desired path.

8.5.2 Simulation Results

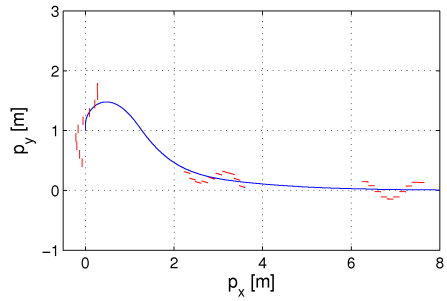
The simulation results are shown in Fig. 8.5. From Figures 8.5(a)-(b), we see that the position of the snake robot converges nicely to the desired path (i.e. the x axis). Fig. 8.5(a) shows the configuration of the snake robot at $t = 1$ s, $t = 30$ s, and $t = 70$ s. Note that Fig. 8.5(b) shows the cross-track error in terms of the y axis coordinate of the CM of the robot, not the transformed y axis coordinate given by (8.13b). The heading of the snake robot, shown in Fig. 8.5(c), also converges nicely to zero, i.e. to the direction of the desired path. As seen in Fig. 8.5(e), the forward velocity is always nonzero and positive, as required by Assumption 8.1, and converges to the velocity $v_t^* = \frac{c_2}{2Nc_1}\alpha^2\omega k_\delta = 0.1$ m/s, which was estimated above. Fig. 8.5(f) shows the joint coordinate of an arbitrarily chosen joint (joint 5) during the path following. The plot shows a very good tracking of the corresponding joint reference coordinates. In summary, the simulation results illustrate that the proposed path following controller successfully steers the snake robot towards and along the desired straight path.

8.6 Experimental Study: The Performance of the Straight Line Path Following Controller

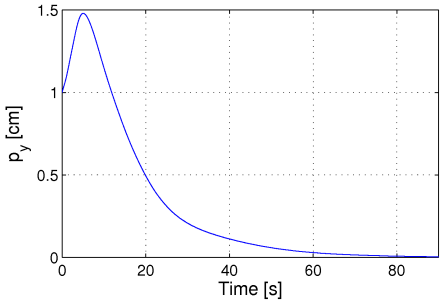
We have experimentally investigated the performance of the straight line path following controller proposed in Section 8.2 by use of the snake robot Wheeko, which was presented in Chapter 3. The experimental results are presented in the following.

8.6.1 Implementation Issues

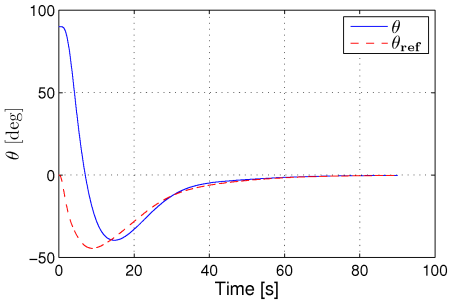
The joint coordinates of the simplified model (linear link displacements), on which the path following controller is based, are different from the joint coordinates of the physical snake robot (joint angles). However, it still makes sense to employ the path following controller to control the physical snake robot since, as implied by Property 4.8, the rotational link motion is what produces the linear displacements captured by the simplified model.



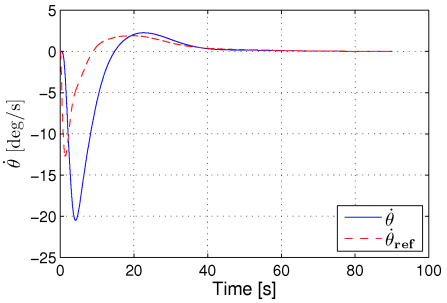
(a) The path of the snake robot.



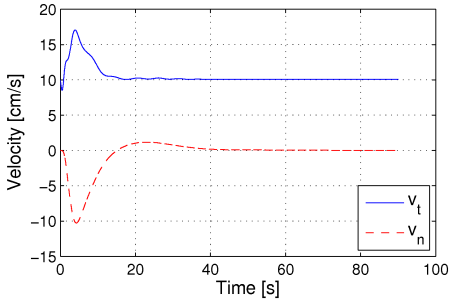
(b) Cross-track error, p_y.



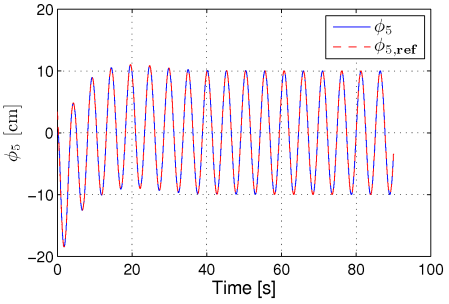
(c) Heading angle, θ.



(d) Angular velocity, $\dot{\theta}$.



(e) Velocity, v_t and v_n.



(f) Coordinate of joint 5, ϕ_5.

Figure 8.5: Simulation of straight line path following with the snake robot initially headed away from the desired path.

The simplified model describes the *qualitative* behaviour of a snake robot with revolute joints, and also approximates the *quantitative* behaviour of the robot for some choice of the ground friction coefficients c_1, c_2, c_3, and c_4. However, no definite mapping exists between the ground friction coefficients of a snake robot with revolute joints and the friction coefficients c_1 - c_4. In other words, the values of c_1 - c_4 that reflected the specific ground friction conditions of the experiments were not know. Since c_3 and c_4 appear in the equations of the path following controller, we chose to treat these coefficients as controller gains in order to implement the controller of the physical robot.

The unspecified values of c_1 - c_4 prevented us from determining the coordinate transformation distance ϵ in (8.14), which depends on c_2 and c_4. During the experiments, we therefore set this coordinate transformation distance to $\epsilon = 0$, i.e. we measured the cross-track error as $\overline{p}_y = p_y$. Note that the value of ϵ could anyhow be expected to be small based on the interpretation of this parameter given in Section 8.2.3, which means that measuring the cross-track error from the CM of the robot or from a point located a small distance ϵ away from the CM can be expected to produce similar motion. Note also that since the ϵ transformation is tangential to the robot, the value of ϵ has approximately no effect on the cross-track error when the heading of the snake robot with respect to the path is close to zero.

8.6.2 Implementation of the Path Following Controller of the Physical Snake Robot

The experimental setup, which consisted of the snake robot Wheeko and the camera-based position measurement system, is presented in Chapter 3. The path following controller of the snake robot was implemented on an external computer according to (8.17), (8.23), and (8.25). We did not implement the joint torque controller given by (8.19) and (8.20) since accurate torque control is not supported by the servo motors installed in the snake robot. The joint angles were instead controlled according to a proportional controller implemented in the microcontroller of each joint module. Note that we can experimentally validate Theorem 8.2 without implementing the joint controller in (8.19) and (8.20) since, as stated in Remark 8.9, the global $\mathcal{K}$-exponential stability of the complete system only requires that the error dynamics of the joints is exponentially stabilized.

The orientation $\overline{\theta}$ of the snake robot was estimated according to (2.2), i.e. as the average of the individual link angles. Furthermore, the forward

velocity $\overline{v}_t$ of the robot, which is needed to calculate the joint angle offset in (8.25), was estimated at 0.5 Hz as the displacement of the CM of the robot divided by the sampling interval (i.e. 2 s). The sampling interval was chosen to be large to obtain a reasonably accurate velocity estimate, but was sufficiently short for the experiments since the robot was moved at a slow pace. As explained in Section 8.6.1, the coordinate transformation distance in (8.14) was set to $\epsilon = 0$, i.e. we measured the cross-track error as $\overline{p}_y = p_y$.

The LOS angle θ_{ref} given by (8.23) was calculated with a look-ahead distance Δ equal to half the length of the snake robot, i.e. $\Delta = 0.7$ m. We conjecture that this value is well above the lower limit of Δ given by (8.28). The actual values of $V_{\min}$ and $V_{\max}$ are not known a priori, and as noted in Remark 8.6, specifying the bounds on Δ as a function of the gait pattern parameters α, ω, δ, and ϕ_o remains a topic of future work. To ensure a smooth control input, the LOS angle θ_{ref} was passed through a 3rd order low-pass filtering reference model (see Appendix C.2). The parameters of the reference model were set to $\omega = \pi/2$ and $\zeta = 1$. The output from this filter also provided the derivatives $\dot{\theta}_{\text{ref}}$ and $\ddot{\theta}_{\text{ref}}$, which are required in the calculation of ϕ_o in (8.25). The evolution of the reference values from the filter were calculated with a first-order numerical integration scheme.

The joint angle offset ϕ_o given by (8.25) was calculated with the gains set to $k_\theta = 1$, $c_3 = 0.5$ and $c_4 = 20$. The friction coefficients c_3 and c_4 were treated as controller gains, as explained in Section 8.6.1. We saturated the joint angle offset according to $\phi_o \in [-25^\circ, 25^\circ]$ in order to keep the joint reference angles within reasonable bounds with respect to the maximum allowable joint angles of the physical snake robot. This saturation also avoided the singularity in (8.25) at $v_t = 0$ (see Remark 8.4). Furthermore, to ensure that the joint angle offset was smooth despite of any steps in the estimate of the forward velocity $\overline{v}_t$, we filtered ϕ_o with a 1st order low-pass filter with cutoff frequency at 1.25 Hz.

The reference angles corresponding to the horizontal joint motion of the robot were calculated according to (8.17) with $N = 10$ links and with gait parameters $\alpha = 30^\circ$, $\omega = 50^\circ/\text{s}$, and $\delta = 36^\circ$. The reference angles corresponding to the vertical joint motion were set to zero to achieve a purely planar locomotion.

8.6.3 Experimental Results

The straight line path following controller was experimentally investigated from three different sets of initial conditions. In the first trial of the ex-

periment, the initial state of the snake robot was approximately $\boldsymbol{\phi} = \mathbf{0}^\circ$, $\theta = 0^\circ$, $p_x = 0$ m, $p_y = 1.3$ m, $\boldsymbol{v}_\phi = \mathbf{0}^\circ/\text{s}$, $v_\theta = 0^\circ/\text{s}$, $v_t = 0$ m/s, and $v_n = 0$ m/s, i.e. the snake robot was initially headed *along* the desired path (the x axis) and the initial distance from the CM to the desired path was 1.3 m. In the second trial, the initial state of the snake robot was approximately $\boldsymbol{\phi} = \mathbf{0}^\circ$, $\theta = 90^\circ$, $p_x = 0$ m, $p_y = 0.5$ m, $\boldsymbol{v}_\phi = \mathbf{0}^\circ/\text{s}$, $v_\theta = 0^\circ/\text{s}$, $v_t = 0$ m/s, and $v_n = 0$ m/s, i.e. the snake robot was initially headed *away* from the desired path (the x axis) and the initial distance from the CM to the desired path was 0.5 m. In the third and final trial, the initial state of the robot was approximately $\boldsymbol{\phi} = \mathbf{0}^\circ$, $\theta = -90^\circ$, $p_x = 0$ m, $p_y = 0.9$ m, $\boldsymbol{v}_\phi = \mathbf{0}^\circ/\text{s}$, $v_\theta = 0^\circ/\text{s}$, $v_t = 0$ m/s, and $v_n = 0$ m/s, i.e. the snake robot was initially headed *towards* the desired path (the x axis) and the initial distance from the CM to the desired path was 0.9 m.

The experimentally measured motion of the snake robot from the first trial is presented in Figures 8.6 and 8.7, from the second trial in Figures 8.8 and 8.9, and from the third trial in Figures 8.10 and 8.11. The desired path, i.e. the global x axis, is indicated with a black line on the floor in the pictures of the snake robot during the three trials.

The visualizations in Figures 8.6, 8.8, and 8.10 indicate that the snake robot converged nicely towards and along the desired path during all three trials. This claim is supported by the plots of the cross-track error in Figures 8.7(b), 8.9(b), and 8.11(b), respectively, which show that the cross-track error converges to and oscillates about zero. For a snake robot with revolute joints, it is difficult to achieve a purely non-oscillating motion of the CM, which was achieved in the simulation results based on the simplified model in Section 8.5. We therefore expected the cross-track error to oscillate about zero, as seen in the plots, rather than converge to zero.

Similar to the oscillatory behaviour of the CM, the heading $\overline{\theta}$ of the snake robot was also expected to oscillate. In particular, while θ provides an explicit representation of the heading in the simplified model, such a representation is not available for a snake robot with revolute joints, which forced us to estimate the heading according to $\overline{\theta}$, i.e. as the average of the link angles (see Remark 6.2). The oscillatory behaviour of $\overline{\theta}$ was thereby expected since the average of the link angles will not always be identically zero during forward locomotion. The heading during the trials is shown in Figures 8.7(c), 8.9(c), and 8.11(c), respectively, which clearly show that $\overline{\theta}$ oscillates nicely about the reference heading $\overline{\theta}_{\text{ref}}$. In all three trials, the heading converges to and oscillates about zero, i.e. the direction of the desired path.

The forward velocity of the robot during each trial is shown in Figures 8.7(d), 8.9(d), and 8.11(d), respectively. The variations in the velocity were primarily caused by the joint angle offset ϕ_o during turning motion, which sometimes interfered with the oscillatory body wave motion and caused the robot to lose momentum.

The joint angle of an arbitrarily chosen joint (joint 5) during each trial is shown in Figures 8.7(f), 8.9(f), and 8.11(f), respectively, which indicate that the snake robot tracked its joint reference coordinates very well.

In summary, the proposed path following controller successfully steered the snake robot towards and along the desired straight path during all three trials of the experiment.

8.7 Simulation Study: The Performance of the Waypoint Guidance Strategy

This section presents simulation results in order to investigate the performance of the waypoint guidance strategy proposed in Algorithm 8.1. In addition to simulation results from the simplified model of the snake robot given by (6.24), we also include simulation results from the complex model (2.46) to show that the applicability of the guidance strategy does not rely on the simplifications of the simplified model. Both models were implemented in *Matlab R2008b* and the dynamics was calculated using the *ode45* solver in Matlab.

8.7.1 Implementation of the Guidance Strategy with the Simplified Model

We considered a snake robot with $N = 10$ links of length $l = 0.14$ m, mass $m = 1$ kg, and moment of inertia $J = 0.0016$ kgm^2. All initial state values were set to zero. Furthermore, we chose the friction coefficients as $c_1 = 0.4$, $c_2 = 2.2$, $c_3 = 0.5$ and $c_4 = 20$.

The radius of the acceptance circle enclosing each waypoint was $r_{\text{accept}} = 0.5$ m. The path following controller was implemented according to (8.17), (8.19), (8.20), (8.23), and (8.25), and with the coordinate transformation distance in (8.14) set to $\epsilon = -19.8$ cm. The controller gains were $k_\phi = 20$, $k_{v_\phi} = 5$, and $k_\theta = 0.06$, and the gait parameters were $\alpha = 0.1$ m, $\omega = 70°/\text{s}$, and $\delta = 40°$. We chose the look-ahead distance as $\Delta = 1.4$ m, which corresponds to the length of the snake robot, and conjecture that this value is well above the lower bound given by (8.28).

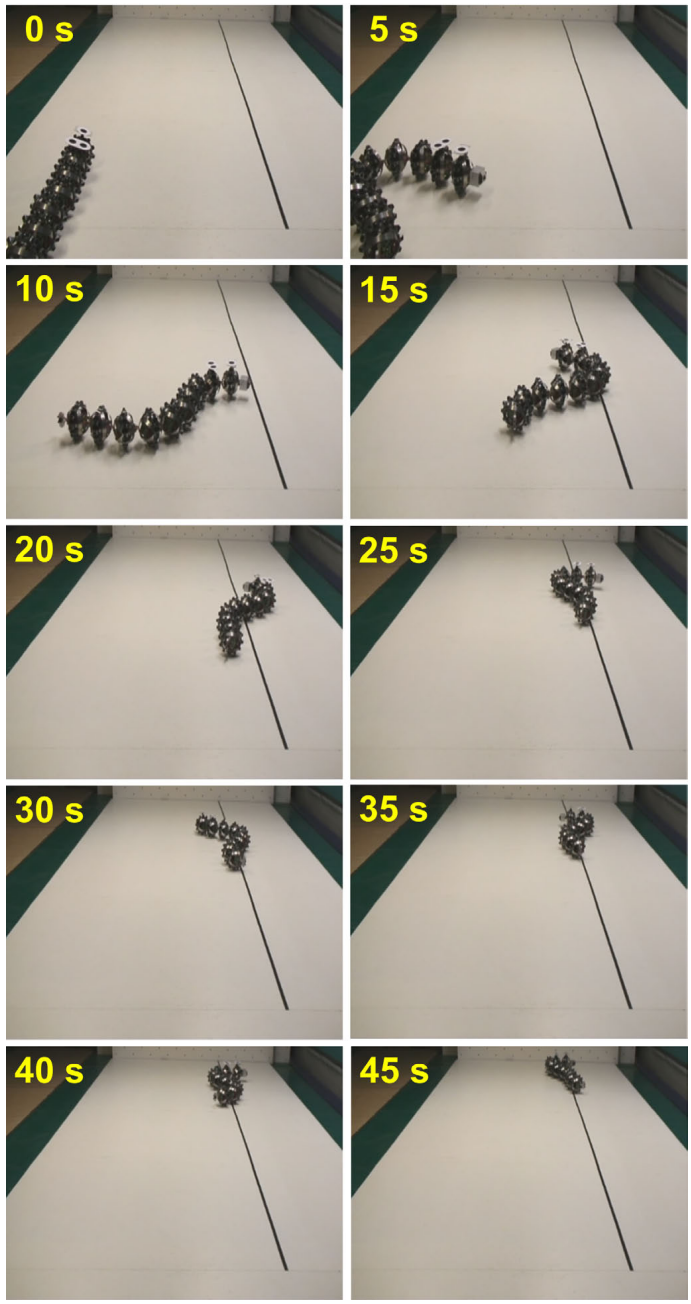
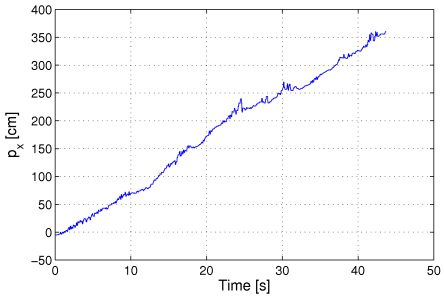
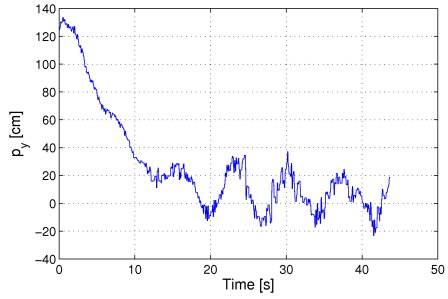


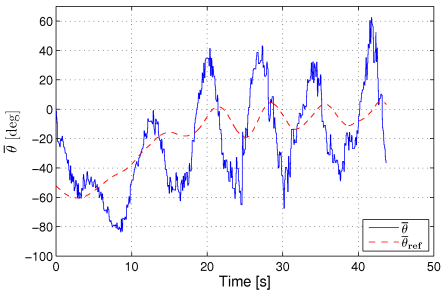
Figure 8.6: The motion of the snake robot during path following with initial heading along the desired path. The black line on the floor indicates the desired path, i.e. the global x axis.



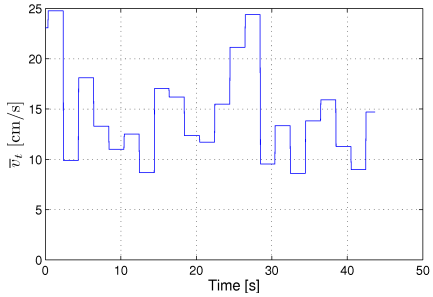
(a) Position along the path, p_x.



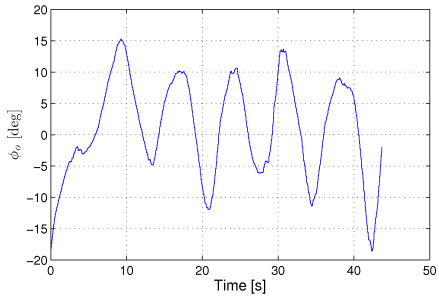
(b) Cross-track error, p_y.



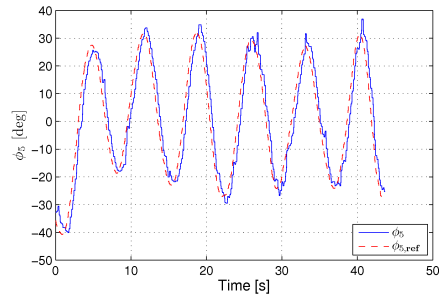
(c) Heading angle, $\overline{\theta}$.



(d) Forward velocity, $\overline{v}_t$.



(e) Joint angle offset, ϕ_o.



(f) Angle of joint 5, ϕ_5.

Figure 8.7: Straight line path following with the physical snake robot initially headed along the desired path.

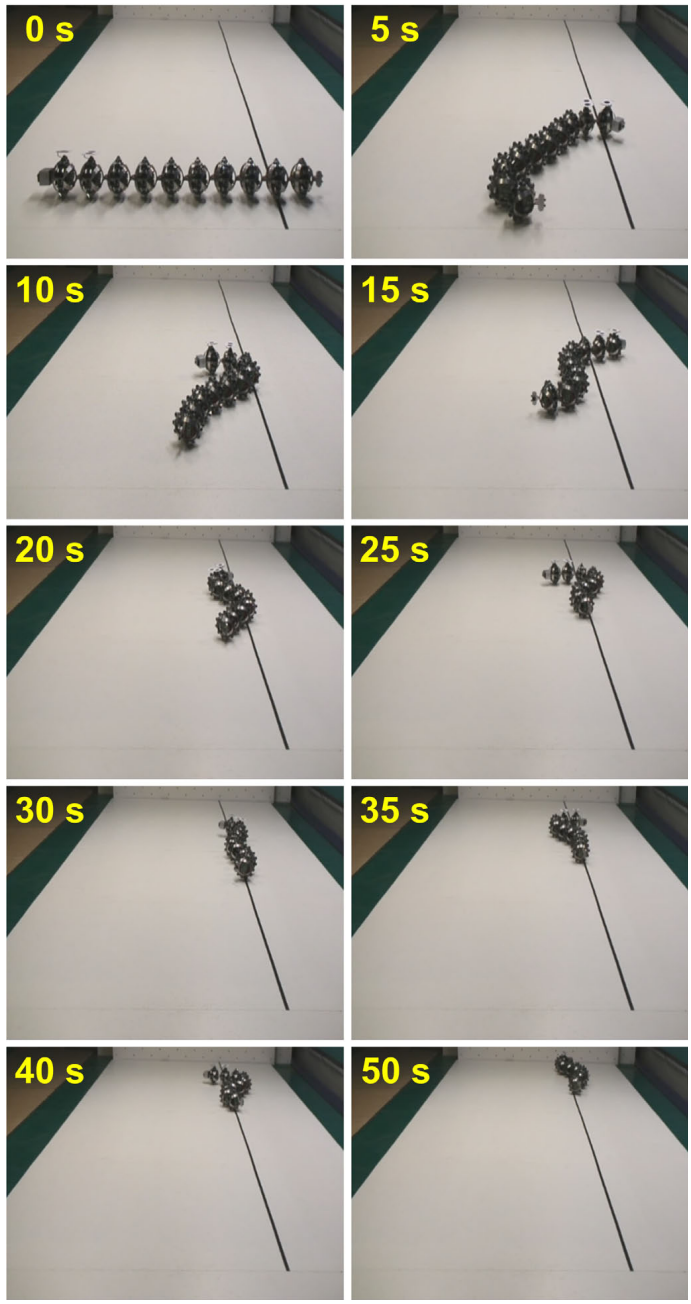
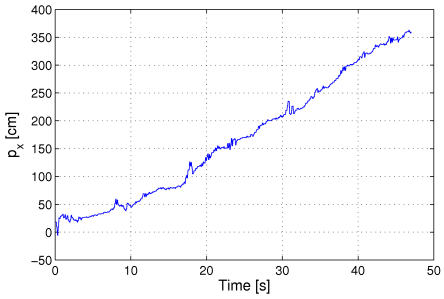
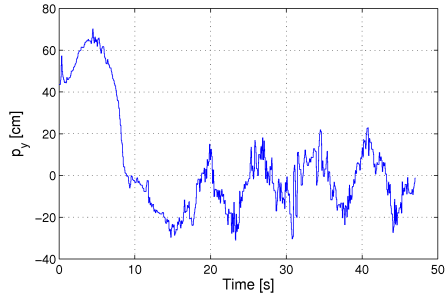


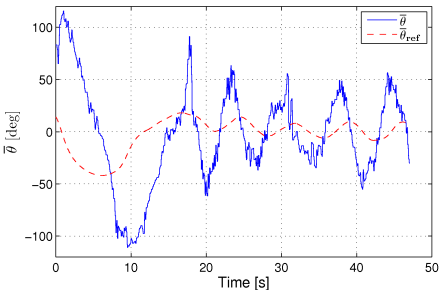
Figure 8.8: The motion of the snake robot during path following with initial heading away from the desired path. The black line on the floor indicates the desired path, i.e. the global x axis.



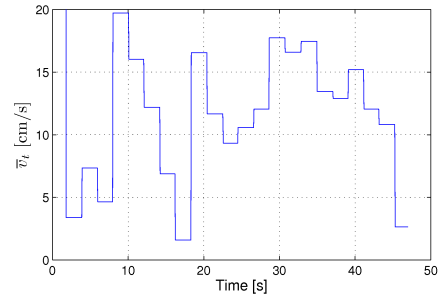
(a) Position along the path, p_x.



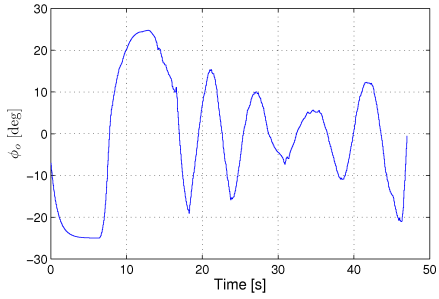
(b) Cross-track error, p_y.



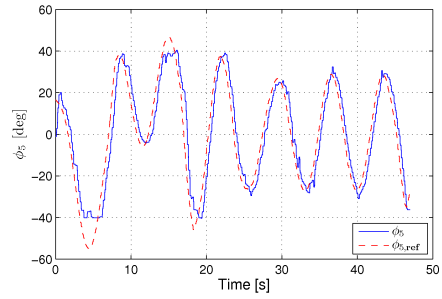
(c) Heading angle, $\overline{\theta}$.



(d) Forward velocity, $\overline{v}_t$.



(e) Joint angle offset, ϕ_o.



(f) Angle of joint 5, ϕ_5.

Figure 8.9: Straight line path following with the physical snake robot initially headed away from the desired path.

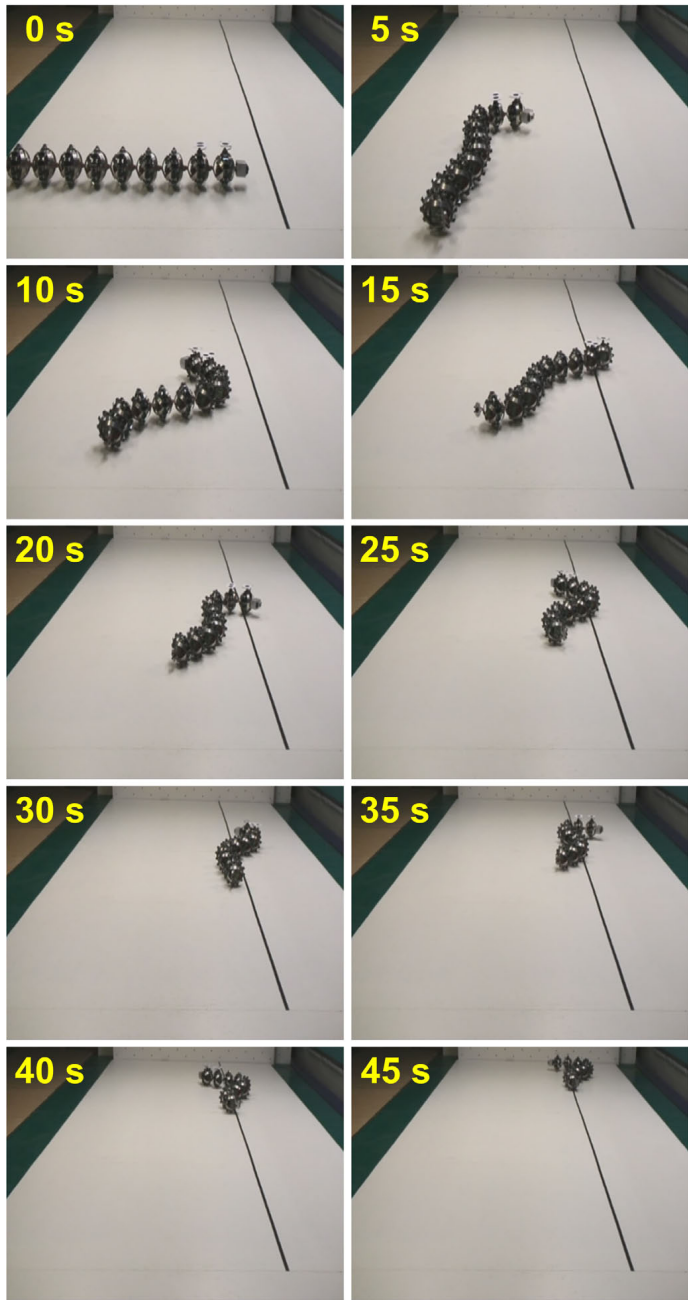
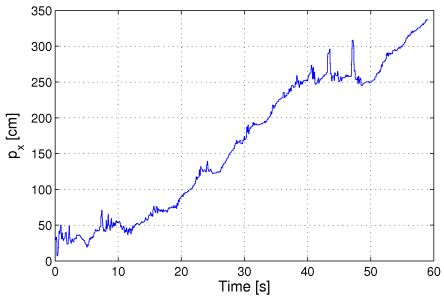
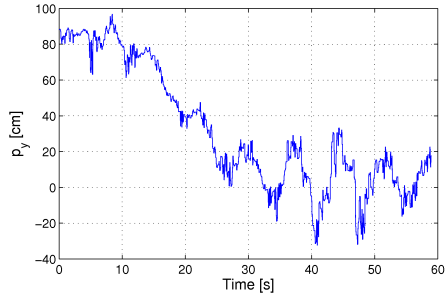


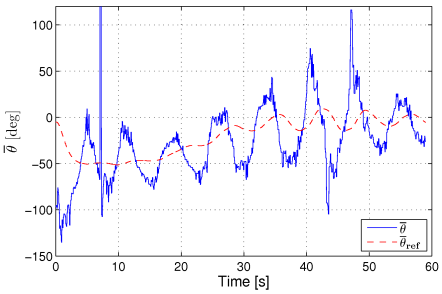
Figure 8.10: The motion of the snake robot during path following with initial heading towards the desired path. The black line on the floor indicates the desired path, i.e. the global x axis.



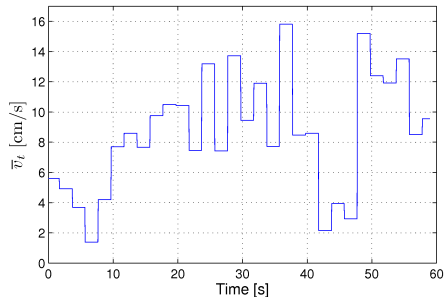
(a) Position along the path, p_x.



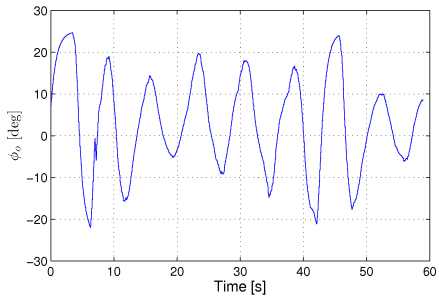
(b) Cross-track error, p_y.



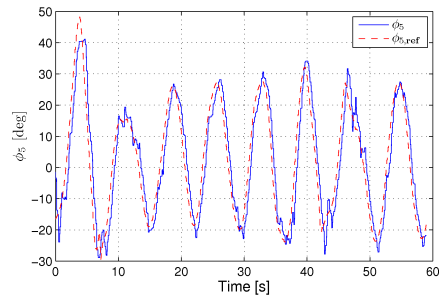
(c) Heading angle, $\overline{\theta}$.



(d) Forward velocity, $\overline{v}_t$.



(e) Joint angle offset, ϕ_o.



(f) Angle of joint 5, ϕ_5.

Figure 8.11: Straight line path following with the physical snake robot initially headed towards the desired path.

The derivatives of θ_{ref} and ϕ_o with respect to time, which are required in (8.20) and (8.25), were calculated by using a 3rd order low-pass filtering reference model (see Appendix C). The parameters of the reference model were set to $\omega = \pi/2$ and $\zeta = 1$. The joint angle offset was saturated according to $\phi_o \in [-0.08 \text{ m}, 0.08 \text{ m}]$ in order to avoid the singularity in (8.25) at $v_t = 0$.

8.7.2 Implementation of the Guidance Strategy with the Complex Model

The ground friction in the complex model was defined by the viscous friction forces in (2.24) with friction coefficients $c_t = 0.55$ and $c_n = 3$. The orientation $\overline{\theta}$ and the forward velocity $\overline{v}_t$ were calculated according to (2.2) and (2.5), respectively. Based on the argumentation given for the physical snake robot in Section 8.6.2, the coordinate transformation distance ϵ in (8.14) was set to zero in the complex model (i.e. we measured the cross-track error as $\overline{p}_y = p_y$).

The shape variables of the simplified model (i.e. the transversal distance between the links) and the complex model (i.e. the joint angles) are different. The controller parameters of the complex model were therefore scaled to account for this difference. In particular, it is shown in Fig. 4.4 in Section 4.9 that lateral undulation with $\alpha = 30^\circ$ and $\delta = 40^\circ$ will displace the links transversally with amplitude of about 0.1 m. We therefore scaled the controller parameters of the complex model with respect to the parameters employed in the simplified model according to the scaling factor $30^\circ/0.1$ m. In other words, we set $\alpha = 30^\circ$, $k_\theta = 0.06\frac{30\pi/180}{0.1} = 0.3$, and saturated the joint offset according to $\phi_o \in [-0.08\frac{30^\circ}{0.1}, 0.08\frac{30^\circ}{0.1}] = [-25^\circ, 25^\circ]$. The remaining controller parameters of the complex model were set equal to the controller parameters of the simplified model.

8.7.3 Simulation Results

We defined $k = 7$ waypoints with global frame coordinates $(3, 0)$, $(3, 3)$, $(6, 3)$, $(6, 6)$, $(0, 6)$, $(2, 3)$, and $(0, 0)$, respectively. Fig. 8.12 shows the motion of the CM of the snake robot from the simplified model (solid line) and the complex model (dashed line), where each waypoint is indicated with a black square. The figure also shows the shape and position of the robot in green at $t = 20$ s, $t = 90$ s, and $t = 180$ s for the simplified model, and at $t = 55$ s, $t = 125$ s, and $t = 235$ s for the complex model. Furthermore, Fig. 8.13 shows the cross-track error (in terms of the y axis

coordinate of the CM of the robot), the heading angle, and the forward velocity from the two models. The vertical lines in the plots indicate time instants where the guidance strategy switches to the next waypoint. We see that the state of the robot experiences a jump at each waypoint switch since, by Algorithm 8.1, the global frame is redefined at a waypoint switch.

As seen in Fig. 8.12, the snake robot has a nice and smooth motion towards each waypoint. The plotted paths, in particular the path near the waypoint at coordinate $(0, 6)$, indicate that the snake robot is able to turn more rapidly in the complex model compared to the simplified model. The qualitative behaviour of the two models are, however, similar. Fig. 8.13(a)-(b) shows that the cross-track error converges nicely to zero after each waypoint switch. The heading of the snake robot, shown in Fig. 8.13(c)-(d), also converges nicely to zero, i.e. to the direction of the desired path. In summary, the simulation results illustrate that the proposed waypoint guidance strategy successfully steers the snake robot towards each of the specified waypoints.

8.8 Chapter Summary

This chapter is summarized as follows:

- We have proposed a path following controller that enables snake robots to track straight paths.
- Using cascaded systems theory, we have proven that the proposed path following controller $\mathcal{K}$-exponentially stabilizes the snake robot to any desired straight path (see Theorem 8.2).
- The proof relies on the assumption that the forward velocity of the robot is contained in some nonzero and positive interval. Specifying the bounds of this interval as a function of the gait pattern parameters remains a topic of future work.
- We have investigated the performance of the path following controller through simulations and through experiments with the snake robot Wheeko, where the proposed controller was shown to successfully steer the snake robot towards and along the desired straight path.
- We have described how the straight line path following controller can be extended to path following of general curved paths by employing an approach previously proposed in the marine control literature for path following control of marine vessels.

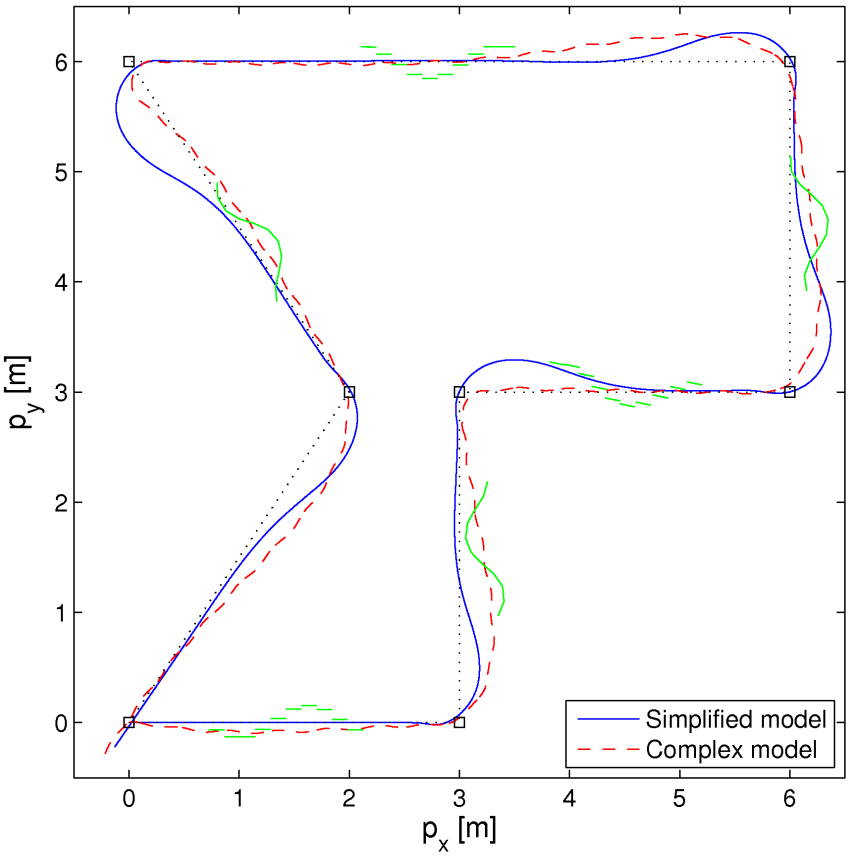
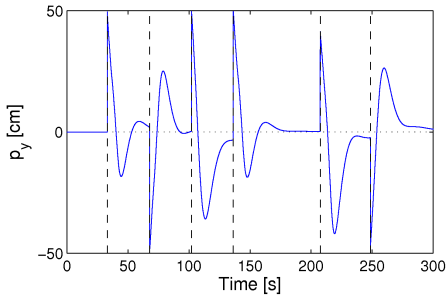
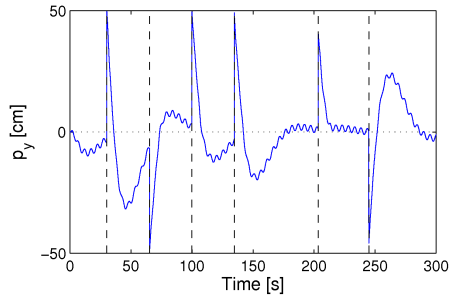


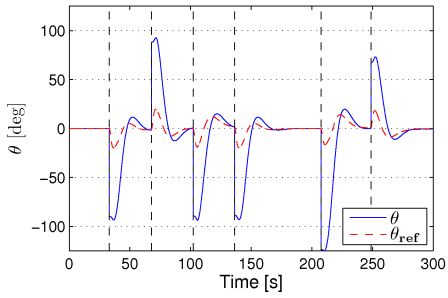
Figure 8.12: The simulated path of the snake robot from the simplified model (solid line) and the complex model (dashed line) during waypoint guidance.



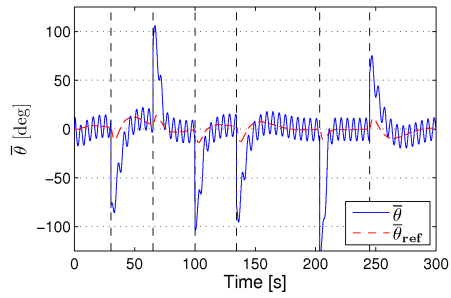
(a) Cross-track error, p_y (simplified model).



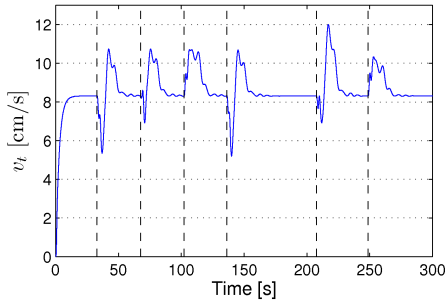
(b) Cross-track error, p_y (complex model).



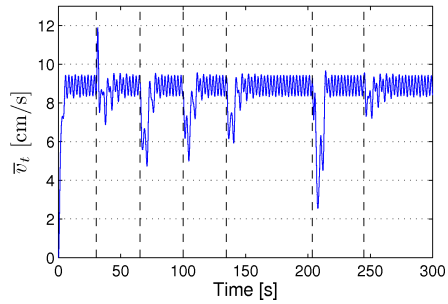
(c) Heading angle, θ (simplified model).



(d) Heading angle, $\overline{\theta}$ (complex model).



(e) Forward velocity, v_t (simplified model).



(f) Forward velocity, $\overline{v}_t$ (complex model).

Figure 8.13: Simulation of the waypoint guidance strategy with the simplified (left) and the complex (right) model of the snake robot.

- We have proposed a waypoint guidance strategy for steering a snake robot along a path defined by waypoints interconnected by straight lines (see Algorithm 8.1).
- We have proven that the waypoint guidance strategy is guaranteed to steer the position of the snake robot into the acceptance region of each waypoint (see Theorem 8.3).
- We have presented simulation results that illustrated the successful performance of the waypoint guidance strategy.

Part II

Snake Robot Locomotion in Unstructured Environments

Chapter 9

Introduction to Part II

The long-term goal of our research activities on snake robot locomotion is to enable these mechanisms to move intelligently and efficiently in unknown and unstructured environments (see Fig. 9.1). The second part of this thesis is therefore devoted to snake robot locomotion in environments that are no longer assumed to be flat, which is more in line with practical applications of snake robots. The second part of the thesis is also in line with current trends in robotic research, which aim at making robots more 'aware' of their environment (e.g. for grasping and object manipulation purposes) and enabling them to work in unknown and unstructured environments (see EUROP/CARE, 2009). The results presented in Part II of this thesis have been developed on the basis of the following fundamental hypothesis:

Hypothesis 9.1 *Intelligent and efficient snake robot locomotion in unknown and unstructured environments requires that the snake robot can* sense *its environment and* adapt *its body shape and movements accordingly.*

We can provide both *theoretical* and *empirical* justifications for Hypothesis 9.1. A theoretical justification for the hypothesis follows from the analysis in Chapter 4, which enables us to conclude that the fundamental control principle of snake robot locomotion is to produce body shape changes that induce external contact forces whose sum points in the desired direction of motion. Since controlled body shape changes for inducing desired contact forces is equivalent to environment *adaptation*, and since environment adaptation is not possible without *sensing* the environment in some way, we have established a theoretical justification for Hypothesis 9.1.

An empirical justification for Hypothesis 9.1 follows from observations of biological snakes in nature. As described in Bauchot (1994), the sensory

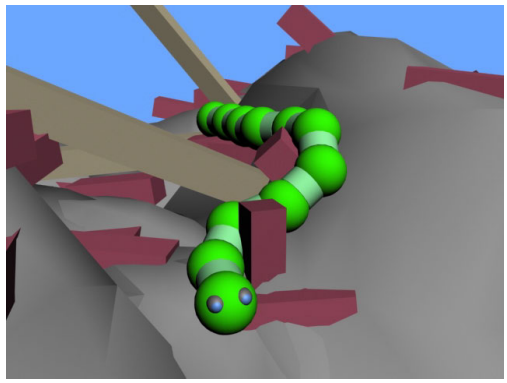


Figure 9.1: Snake robot locomotion in an unstructured environment.

information transmitted by the skin of a snake influences the shape to which the body adapts in a given situation. The body adaptation of snakes with respect to external objects is studied analytically and empirically in Gray (1946); Hirose (1993); Moon and Gans (1998). Environment sensing and adaptation is, in other words, inherently present in the motion of any snake, which serves as an empirical justification of Hypothesis 9.1.

Note that environment sensing and adaptation was not necessary in the control design efforts of Part I of this thesis since the surface beneath the snake robot was assumed to be *flat*, which allowed us to employ predetermined gait patterns in open-loop without sensing the external forces.

It is implied by Hypothesis 9.1 that we recognize the interaction between the snake robot and its environment as the propelling force. We also made this recognition in Part I of this thesis. However, while ground friction represents the propelling forces during flat surface locomotion, the propelling forces in Part II will be contact forces from external objects (or obstacles) in the environment of the snake robot. At this point in the discussion, we are actually at the very core of the principle underlying snake robot locomotion. In particular, by considering Hypothesis 9.1 at a glance, one realizes that an interesting and unique feature of snake robot locomotion compared to other forms of robotic mobility is that irregularities on the ground are actually beneficial for the propulsion since they provide push points for the robot. The word *obstacle* will be used in the following chapters to denote an object or an irregular surface in the path of the snake robot that can be utilized for propulsion. This may seem like a contradiction since it is, in fact, not an obstacle from the point of view of the snake robot. However,

the characterization is valid in the sense of mobile robotics in general. The use of this denotation therefore helps emphasize one of the fundamental differences between snake locomotion and other traditional means of mobility, such as wheeled, tracked, and legged mobility. While in traditional mobile robotics, the aim is typically to avoid obstacles, a snake robot should rather seek out and make contact with obstacles since they represent push points that can be utilized for more efficient propulsion. Hence, for snake robots, the aim is not *obstacle avoidance*, but rather *obstacle utilization*. To fully embrace this concept, the term *obstacle-aided locomotion* was previously introduced by our research group in Transeth *et al.* (2008*a*).

We can now summarize the material in Part II of this thesis as modelling and control strategies for obstacle-aided locomotion. In accordance with the scope of the thesis described in Section 1.4, we will maintain a planar perspective throughout the remaining chapters, i.e. we consider the motion of the snake robot to be purely horizontal on a surface with vertical obstacles that induce horizontal contact forces on the robot. The reason for only considering planar motion is, as explained in Section 1.4, that we believe the essential control principles of snake robot locomotion are contained in a planar perspective. In particular, since the fully three-dimensional motion of a snake robot consists of motion components in a horizontal and vertical plane, respectively, we conjecture that control laws that fulfil some control objective in a planar perspective can be extended to fulfil a similar control objective in a fully three-dimensional perspective. Moreover, it makes no sense to attack the problem of three-dimensional motion in an unstructured environment before the simpler case of planar obstacle-aided locomotion is well understood.

We end this introductory text with a note on the literature review presented in Section 1.3. An interesting observation that can be made from this literature review (one might even call this a paradox) is that the majority of previous research on snake robots has focused on flat surface motion even though the main advantage of snake robots are their potential ability to move in unstructured environments. As described in Section 1.3, only a few published works consider snake robot locomotion in situations where the surface is no longer assumed to be flat. We therefore hope that the material presented in the second part of this thesis addresses a small part of the, in our opinion, large research gap that must be closed before we will ever see useful snake robots outside the laboratory.

Chapter 10

A Hybrid Model of Snake Robot Locomotion in Unstructured Environments

We begin the second part of this thesis by extending the model of the snake robot presented in (2.34) to include contact forces from external obstacles in the environment around the robot. Since the interaction with an obstacle represents a discrete event that only occurs when a link of the robot comes into contact with an obstacle, the snake robot will be subjected to both *continuous* and *discontinuous* dynamics in this environment. We will therefore describe the dynamics of the snake robot in terms of a *hybrid model* by employing the hybrid modelling framework described in Goebel *et al.* (2009).

An important difference between models of continuous dynamical systems, such as the model in (2.34), and models of hybrid dynamical systems, is that while most continuous models always exhibit a unique solution to the evolution of the state vector, a hybrid model may have a single solution, several solutions, or no solution at all. For the hybrid model of the snake robot, we will handle this existence and uniqueness issue by formulating the equations governing the obstacle contact forces as a *linear complementarity problem* (LCP). This formulation enables us to apply existing general results concerning existence and uniqueness of solutions to LCPs (see Cottle *et al.*, 1992) to the model of the snake robot.

A long-term goal of the model proposed in this chapter is to facilitate development of model-based control laws for obstacle-aided locomotion with provable stability properties. We will therefore make several simplifying

assumptions during the modelling process so that the environment interaction model maintains a simple and analytical form. In particular, we will model the interaction with obstacles by introducing a *unilateral velocity constraint* on each contacted link of the snake robot. This approach simplifies the equations of motion since the shape of the obstacles does not have to be considered explicitly.

In order to illustrate the validity of the proposed modelling approach, this chapter includes a simulation study where simulation results from the proposed model of the snake robot are shown to agree well with simulation results from a more extensive model of obstacle-aided locomotion previously presented in Transeth *et al.* (2008*a*).

Contributions of this Chapter: The contribution of this chapter is the *hybrid model* of a planar snake robot interacting with obstacles in its environment. In contrast to the hybrid model presented in Transeth *et al.* (2008*a*), which is based on a *timestepping* method that approximates trajectories of the hybrid system without tracking events, the hybrid model presented in this chapter is based on *event-tracking* (Schaft and Schumacher, 2000), where discrete events are tracked. We believe the model formulation with this approach is better suited for analysis and synthesis of model-based controllers. Modelling obstacle interaction by introducing a *unilateral velocity constraint* on each contacted link of the snake robot is a novel approach. In particular, the conventional approach for modelling the obstacle interaction would be to assume that the obstacle constraint force points in the normal direction of the *obstacle* (see Brogliato, 1999). With the approach described in this chapter, the shape of the obstacles does not have to be considered explicitly as we instead calculate constraint forces with respect to the normal direction of the *contacted links*, which simplifies the equations of motion.

Organization of this Chapter: Section 10.1 gives a general presentation of the hybrid modelling framework and also the *linear complementarity problem* (LCP). The model of the snake robot without obstacles, that was presented in Chapter 2, is reformulated in a slightly different form in Section 10.2, and an overview of the contact modelling approach is given in Section 10.3. The organization of the subsequent sections is illustrated in Fig. 10.1 to increase the readability of this chapter. In particular, Section 10.4 describes the mechanism for determining if a link is in contact with an obstacle, Section 10.5

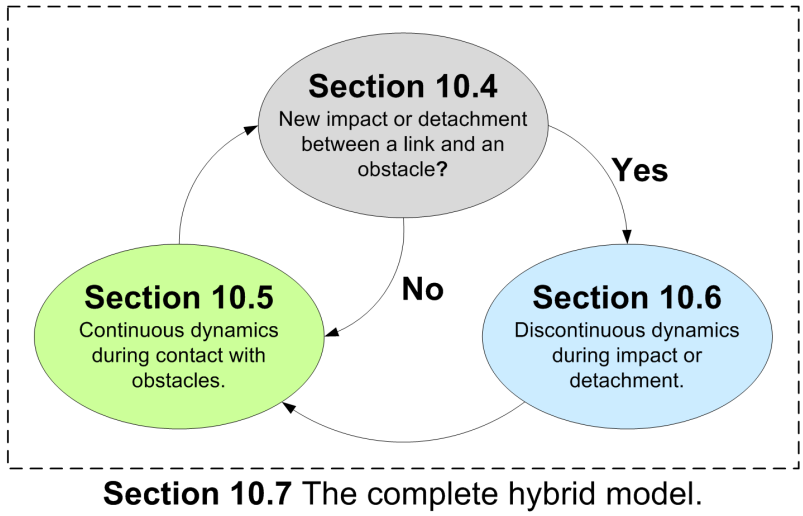


Figure 10.1: The organization of this chapter.

presents the model of the continuous constrained dynamics of the robot during contact with a fixed set of obstacles, Section 10.6 presents the model of the discontinuous dynamics of the robot when the set of links in contact with obstacles changes, and Section 10.7 gives a unified presentation of the snake robot dynamics in terms of the hybrid modelling framework. Simulation results that compare the hybrid model with the model of obstacle-aided locomotion previously proposed in Transeth *et al.* (2008*a*) are presented Section 10.8. Finally, the chapter is summarized in Section 10.9.

Publications: The material presented in this chapter is based on the journal paper Liljebäck *et al.* (2010*h*) and the conference papers Liljebäck *et al.* (2009*a*) and Liljebäck *et al.* (2010*g*).

10.1 Hybrid Dynamical Systems and Complementarity Systems

This section presents the framework employed in order to formulate the *hybrid model* of the snake robot, and also the *linear complementarity problem* (LCP), which is instrumental in the modelling of obstacle contact forces in this chapter.

10.1.1 Modelling of Hybrid Dynamical Systems

A *hybrid dynamical system* is a dynamical system that exhibits both continuous and discontinuous state evolution. A snake robot interacting with obstacles is a hybrid system since the impacts between the snake and the obstacles represent discrete events.

Several modelling frameworks for hybrid systems exist, some of which are presented in Schaft and Schumacher (2000). In this work, we have chosen to employ the modelling framework described in Goebel *et al.* (2009) since this framework captures a wide variety of hybrid phenomena, and it also facilitates stability analysis of hybrid systems.

In accordance with Goebel *et al.* (2009), a hybrid system has a state vector $\boldsymbol{x} \in \mathbb{R}^n$ that can both *flow* (evolve continuously) and *jump* (evolve discontinuously). The data that determine the evolution of $\boldsymbol{x}$ is given by the four elements $(\boldsymbol{C}, \boldsymbol{F}, \boldsymbol{D}, \boldsymbol{G})$, where $\boldsymbol{C}$ denotes the *flow set*, $\boldsymbol{F}$ denotes the *flow map*, $\boldsymbol{D}$ denotes the *jump set*, and $\boldsymbol{G}$ denotes the *jump map* of the hybrid system. Whenever the state $\boldsymbol{x}$ belongs to the flow set $\boldsymbol{C}$, it flows (or evolves continuously) according to $\boldsymbol{F}$. During flows, the system acts as an ordinary continuous dynamical system. However, when $\boldsymbol{x}$ belongs to the jump set $\boldsymbol{D}$, it generally jumps according to $\boldsymbol{G}$ to a new value $\boldsymbol{x}^+$ (superscript $+$ and $-$ denote 'the next value' and 'the previous value', respectively). Hence, the general form of a hybrid dynamical system is given by

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}) \quad \text{for all} \quad \boldsymbol{x} \in \boldsymbol{C}, \\ \boldsymbol{x}^+ &= \boldsymbol{G}(\boldsymbol{x}) \quad \text{for all} \quad \boldsymbol{x} \in \boldsymbol{D}, \end{aligned} \tag{10.1}$$

where we have also included a control input, $\boldsymbol{u} \in \mathbb{R}^m$.

Existence and *uniqueness* of solutions is a very important issue when modelling hybrid systems. From a given initial state, $\boldsymbol{x}_0$, a hybrid system may have a single solution, several solutions, or no solution at all. For a general hybrid system, there are no easily verifiable necessary and sufficient conditions for existence and uniqueness of solutions. However, such conditions exist for special classes of hybrid systems, such as for *complementarity systems* (see Section 10.1.2).

A hybrid system is simulated by letting the state vector $\boldsymbol{x}$ flow according to the flow map $\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})$ as long as $\boldsymbol{x} \in \boldsymbol{C}$. Whenever $\boldsymbol{x} \in \boldsymbol{D}$, the state vector jumps to the new value $\boldsymbol{x}^+$ according to the jump map $\boldsymbol{G}(\boldsymbol{x})$, and the simulation of the flow map is *restarted* from the new initial value given by $\boldsymbol{x}^+$. This approach for simulating hybrid systems is called *event-tracking* (Schaft and Schumacher, 2000) since the discrete events of the model are tracked.

Remark 10.1 ***A note regarding the notation in this chapter.*** *We will employ boolean operators in the formulation of the hybrid model of the snake robot to handle the discrete nature of the obstacle contact forces. In particular,* $\wedge$ *denotes a logical AND operation,* $\vee$ *denotes a logical OR operation,* $\cap$ *denotes the intersection of two sets, and* $\cup$ *denotes the union of two sets. For ease of notation, we will also often use* , *to denote the logical AND operation. For example, the set* $S = \{x | x > a, x < b\}$ *contains all values of* x *that are greater than* a *AND less than* b*.*

10.1.2 Complementarity Systems

A hybrid system is called a *complementarity system* if the flow of the system states is constrained by a set of *complementarity conditions* (Schaft and Schumacher, 2000). A complementarity condition between two scalar variables requires that both variables are nonnegative and that their product is always zero (i.e. one variable is always zero). In mathematical terms, the complementarity condition between two scalar variables x and y can be written $x \geq 0 \wedge y \geq 0 \wedge xy = 0$. Two vectors $\boldsymbol{x} \in \mathbb{R}^m$ and $\boldsymbol{y} \in \mathbb{R}^m$ are said to be *complementary* if, for all i, the pair of variables (x_i, y_i) is subject to a complementarity condition. We will see in Section 10.5 that we can formulate complementarity conditions for the links of the snake robot that are in contact with an obstacle.

The constraint equations of a complementarity system can often be formulated as a *linear complementarity problem* (LCP). A LCP asks whether there exist two complementary vectors $\boldsymbol{x} \in \mathbb{R}^m$ and $\boldsymbol{y} \in \mathbb{R}^m$ such that

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{a} + \boldsymbol{A}\boldsymbol{x}, \\ \boldsymbol{x} \geq \boldsymbol{0}, \boldsymbol{y} &\geq \boldsymbol{0}, \boldsymbol{x}^T \boldsymbol{y} = 0, \end{aligned} \tag{10.2}$$

for a given vector $\boldsymbol{a} \in \mathbb{R}^m$ and a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times m}$. The constraint equations of the snake robot are given in this form in Section 10.5. The following result is proved in Cottle *et al.* (1992):

Theorem 10.1 *The LCP in (10.2) is uniquely solvable for all data vectors* $\boldsymbol{a}$ *if and only if* $\boldsymbol{A}$ *is a P-matrix.*

A P-matrix is a matrix whose *principal minors* are all positive. A principal minor of the matrix $\boldsymbol{A}$ is the determinant of a square submatrix of $\boldsymbol{A}$ consisting of the same set of rows and columns. A real *symmetrical* matrix is a P-matrix if and only if it is *positive definite*. For a real symmetrical

matrix, one can therefore apply the standard criteria for positive definiteness in order to check if the matrix is a P-matrix. If the matrix is not symmetrical, one can e.g. apply the recursive algorithm proposed in Tsatsomeros and Li (2000), which is $O\left(2^m\right)$, in order to check if the matrix is a P-matrix.

Several algorithms exist for solving the LCP in (10.2). A famous approach is the so-called *Lemke's algorithm* (see e.g. Cottle *et al.*, 1992), which basically uses trial and error to find the non-zero elements of $\boldsymbol{x}$ and $\boldsymbol{y}$, but with clever rules for changing the non-zero elements between trials.

10.2 The Dynamics of the Snake Robot without Obstacles

The hybrid model will be developed by extending the unconstrained (no obstacles) model of the snake robot presented in (2.34). In this section, we restate this model with two modifications. The first modification, which concerns the ground friction model, is explained in Section 10.2.1. The second modification, which involves a slight reformulation of the model to a form which is more suitable for inclusion of obstacle contact forces, is described in Section 10.2.2.

10.2.1 The Ground Friction Model

Since the goal of Part II of this thesis is to study snake robot locomotion propelled by obstacle contact forces, we do not want the ground friction forces to contribute to the propulsion since it would then be difficult to know wether the robot is propelled by ground friction or by obstacle contact forces. Consequently, Theorem 4.4 from Chapter 4 suggests that the ground friction on the robot should be *isotropic*. We have considered viscous ground friction in Part I of this thesis based on the argument stated in Section 2.4.1, namely that the motion of the snake robot is qualitatively similar with anisotropic viscous friction as with anisotropic Coulomb friction. With isotropic ground friction, however, this argument is no longer relevant. We therefore choose to employ a Coulomb friction model in the following since, as stated in Section 2.4.1, a Coulomb friction model is more accurate from a physical point of view than a viscous friction model. Moreover, to further increase the accuracy of the ground friction model, we choose to also model the friction torque induced on a link due to the rotation of the link.

From the above discussion, we assume that each link of the snake robot is subjected to an isotropic Coulomb ground friction force acting on the CM of the link, and also a friction torque acting about the link CM. We define the global frame friction force on link $i \in \{1, \ldots, N\}$ in the form of (2.15) as

$$\boldsymbol{f}_{R,i} = \begin{bmatrix} f_{R,x,i} \\ f_{R,y,i} \end{bmatrix} = \begin{cases} -\mu m g \frac{\boldsymbol{v}_i}{|\boldsymbol{v}_i|} & \text{when} \quad |\boldsymbol{v}_i| > 0 \\ \boldsymbol{0}_{2\times 1} & \text{when} \quad |\boldsymbol{v}_i| = 0 \end{cases}, \tag{10.3}$$

where $\mu \geq 0$ is the Coulomb friction coefficient, g is the gravitational acceleration constant, and $\boldsymbol{v}_i = [\dot{x}_i, \dot{y}_i]^T$ is the velocity of link i. We now define the scalar value $\widehat{v}_i \in \mathbb{R}$ given by

$$\widehat{v}_i = \begin{cases} \frac{1}{|\boldsymbol{v}_i|} & \text{when} \quad |\boldsymbol{v}_i| > 0 \\ 0 & \text{when} \quad |\boldsymbol{v}_i| = 0 \end{cases}, \tag{10.4}$$

and also the diagonal matrix $\boldsymbol{\Gamma} = \operatorname{diag}(\widehat{v}_1, \widehat{v}_2, \ldots, \widehat{v}_N) \in \mathbb{R}^{N\times N}$, which enable us to express the friction forces on all links in the form of (2.16) as

$$\boldsymbol{f}_R = \begin{bmatrix} \boldsymbol{f}_{R,x} \\ \boldsymbol{f}_{R,y} \end{bmatrix} = -\mu m g \begin{bmatrix} \boldsymbol{\Gamma}\dot{\boldsymbol{X}} \\ \boldsymbol{\Gamma}\dot{\boldsymbol{Y}} \end{bmatrix} \in \mathbb{R}^{2N}. \tag{10.5}$$

The friction torque about the CM of link i is denoted by $\tau_{R,i}$ and is produced by the friction forces acting normal to the link during link rotation. As illustrated in Fig. 10.2, the friction force $df_{R,i}$ on an infinitesimal mass element dm of link i due to the link rotation $\dot{\theta}_i$ produces a friction torque $d\tau_{R,i}$ about the CM of the link, which is given by

$$d\tau_{R,i} = s df_{R,i} = s\left(-\mu g \cdot \operatorname{sgn}\left(s\dot{\theta}_i\right) \cdot dm\right), \tag{10.6}$$

where s is the distance from the CM of link i to the mass element dm. Using the relation $dm = \frac{m}{2l} ds$, the total friction torque on link i can be calculated as

$$\tau_{R,i} = \int_{-l}^{l} d\tau_{R,i} = -\frac{1}{2}\mu m g l \cdot \operatorname{sgn}\left(\dot{\theta}_i\right). \tag{10.7}$$

The global frame friction torque on all links can be expressed in matrix form as

$$\boldsymbol{\tau}_R = -\frac{1}{2}\mu m g l \cdot \operatorname{sgn}\left(\dot{\boldsymbol{\theta}}\right), \tag{10.8}$$

where $\boldsymbol{\tau}_R = [\tau_{R,1}, \ldots, \tau_{R,N}]^T \in \mathbb{R}^N$.

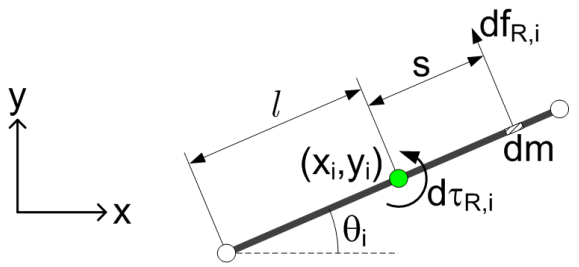


Figure 10.2: The ground friction torque acting about the CM of each link.

10.2.2 The Equations of Motion without Obstacles

It follows directly from the equations of motion in (2.32) that the model of a snake robot influenced by isotropic Coulomb ground friction forces on a flat surface can be written as

$$\boldsymbol{M}_\theta \ddot{\boldsymbol{\theta}} = l\boldsymbol{S}_\theta \boldsymbol{K} \boldsymbol{f}_{R,x} - l\boldsymbol{C}_\theta \boldsymbol{K} \boldsymbol{f}_{R,y} - \boldsymbol{W}\dot{\boldsymbol{\theta}}^2 + \boldsymbol{\tau}_R + \boldsymbol{D}^T \boldsymbol{u}, \tag{10.9}$$

$$Nm\ddot{\boldsymbol{p}} = Nm \begin{bmatrix} \ddot{p}_x \\ \ddot{p}_y \end{bmatrix} = \begin{bmatrix} \boldsymbol{e}^T \boldsymbol{f}_{R,x} \\ \boldsymbol{e}^T \boldsymbol{f}_{R,y} \end{bmatrix} = \boldsymbol{E}^T \boldsymbol{f}_R, \tag{10.10}$$

where $\boldsymbol{\theta} \in \mathbb{R}^N$ and $\boldsymbol{p} \in \mathbb{R}^2$ represent the $N+2$ generalized coordinates of the system, $\boldsymbol{f}_R$ contains the Coulomb friction forces defined in (10.5), $\boldsymbol{\tau}_R$ contains the friction torques defined in (10.8), and

$$\boldsymbol{M}_\theta = J\boldsymbol{I}_N + ml^2 \boldsymbol{S}_\theta \boldsymbol{V} \boldsymbol{S}_\theta + ml^2 \boldsymbol{C}_\theta \boldsymbol{V} \boldsymbol{C}_\theta, \tag{10.11}$$

$$\boldsymbol{W} = ml^2 \boldsymbol{S}_\theta \boldsymbol{V} \boldsymbol{C}_\theta - ml^2 \boldsymbol{C}_\theta \boldsymbol{V} \boldsymbol{S}_\theta, \tag{10.12}$$

$$\boldsymbol{V} = \boldsymbol{A}^T \left(\boldsymbol{D}\boldsymbol{D}^T\right)^{-1} \boldsymbol{A}, \tag{10.13}$$

$$\boldsymbol{K} = \boldsymbol{A}^T \left(\boldsymbol{D}\boldsymbol{D}^T\right)^{-1} \boldsymbol{D}. \tag{10.14}$$

By introducing the configuration vector of the system

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{p} \end{bmatrix} \in \mathbb{R}^{N+2}, \tag{10.15}$$

the model of the snake robot can be written compactly as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} = \boldsymbol{f}_u(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{u}), \tag{10.16}$$

where subscript 'u' denotes *unconstrained* since the motion is not constrained by obstacles, and where

$$\boldsymbol{M}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{M}_\theta & \boldsymbol{0}_{N\times 1} & \boldsymbol{0}_{N\times 1} \\ \boldsymbol{0}_{1\times N} & Nm & 0 \\ \boldsymbol{0}_{1\times N} & 0 & Nm \end{bmatrix} \in \mathbb{R}^{(N+2)\times(N+2)}, \tag{10.17}$$

$$\boldsymbol{f}_u(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{u}) = \begin{bmatrix} l\boldsymbol{S}_\theta \boldsymbol{K} & -l\boldsymbol{C}_\theta \boldsymbol{K} \\ \boldsymbol{e}^T & \boldsymbol{0}_{1\times N} \\ \boldsymbol{0}_{1\times N} & \boldsymbol{e}^T \end{bmatrix} \boldsymbol{f}_R + \begin{bmatrix} -\boldsymbol{W}\dot{\boldsymbol{\theta}}^2 + \boldsymbol{\tau}_R + \boldsymbol{D}^T \boldsymbol{u} \\ 0 \\ 0 \end{bmatrix} \tag{10.18}$$

10.3 Overview of the Contact Modelling Approach

The unconstrained dynamics of the snake robot given by (10.16) will be extended in the following sections in order to include contact forces from external obstacles in the environment of the robot (see section organization in Fig. 10.1). We now describe the contact modelling approach in more detail.

The planar environment of the snake robot consists of an arbitrary number of external obstacles with *circular* shape. We consider circular obstacles to simplify the process of detecting overlap between a link and an obstacle, as described in Section 10.4. Note that this assumption is not very restrictive since most objects can locally be approximated by circular shapes. The friction coefficient between the snake robot and any obstacle is denoted by $\mu_o \geq 0$. Furthermore, we assume that the shortest distance between the edges of any two obstacles is greater than the link length $2l$ to prevent contact on both sides of a link.

The interaction between a snake robot link and an obstacle is modelled by introducing a *unilateral velocity constraint* for the link when it comes into contact with the obstacle. The constraint is *unilateral* (acts in one lateral direction only) since the constraint shall allow sideways motion of the link *away* from the obstacle, but prevent any sideways motion *towards* (and thereby into) the obstacle. As illustrated in Fig. 10.3, the interaction model is somewhat similar to assuming that each contacted link is equipped with tangentially mounted passive wheels with no-slip conditions. There is, however, an important difference between the two situations illustrated in Fig. 10.3 since the constraints introduced by the passive wheels act in both lateral directions of the links (i.e. they are *bilateral*), while the constraints introduced by the obstacles are *unilateral*. Section 10.5.1 presents the equations describing the unilateral constraints for all contacted links.

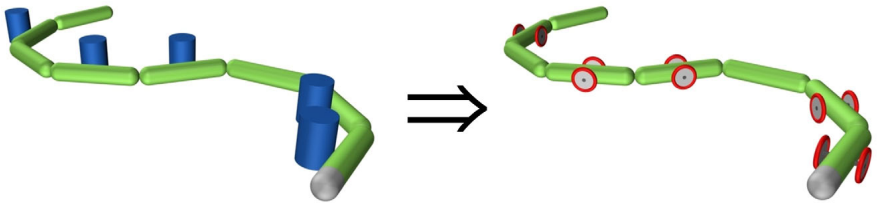


Figure 10.3: The interaction model is similar to assuming that each contacted link is equipped with passive wheels with no-slip conditions.

Remark 10.2 *It was noted at the very end of Section 4.3.2 that links with anisotropic ground friction properties (where the normal direction friction coefficient is much larger than the tangential direction coefficient) are similar to links equipped with tangentially mounted passive wheels with no-slip conditions. The comparison in Fig. 10.3 is therefore particularly interesting since it implies that the phenomenon that propels a snake robot forward due to the anisotropic ground friction properties of the links is similar to the phenomenon that propels a snake robot forward due to the interaction of the links with external obstacles. We will use this observation in Chapter 12 as an argument for continuing to consider lateral undulation motion also in environments with obstacles.*

The obstacle contact force on link $i \in \{1, \ldots, N\}$ consists of two orthogonal components, which are illustrated in Fig. 10.4. The first component is the *constraint force*, $\boldsymbol{f}_{c,i} \in \mathbb{R}^2$, acting in the normal direction of link i and away from the obstacle (parallel to the local y axis of link i). The second component is the obstacle *friction force*, $\boldsymbol{f}_{\mu,i} \in \mathbb{R}^2$, acting in the tangential direction of link i and in the opposite direction of the tangential link velocity (parallel to the local x axis of link i).

We assume that an obstacle contact force (i.e. the *constraint force* and the *friction force* from the obstacle) acts on the CM of a link only. Furthermore, we disregard any contact torque about the CM of the link. This simplifies the equations of motion considerably and does not have any significant influence on the overall motion of the robot when the length of the links is small.

During sustained contact with a fixed set of obstacles, we will show in Section 10.5 how the problem of calculating the resulting obstacle contact forces can be formulated as a LCP, which was introduced in Section 10.1.2. By solving the LCP, we calculate the forces on the CM of the contacted

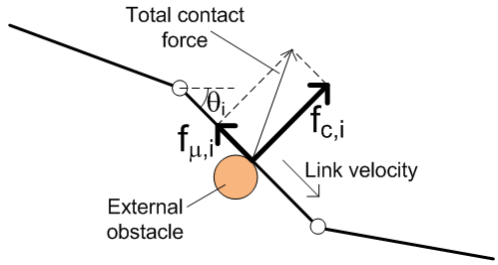


Figure 10.4: The obstacle contact force on link i consists of the normal direction constraint force, $f_{c,i}$, which acts normal to the link, and the friction force, $f_{\mu,i}$, which acts in the tangential link direction.

links that are needed to satisfy the unilateral velocity constraints imposed on each contacted link. These contact forces are then added to the equations of motion in (10.16) in order to cancel out the applied forces acting against the constraints. This represents the dynamics of the snake robot during continuous constrained motion.

When a link (that was previously not in contact with an obstacle) comes into contact with an obstacle, an *impact* occurs. We assume all impacts to be completely *inelastic*, meaning that the normal direction velocity of the link is completely absorbed during the impact. Furthermore, we assume all impacts to be *instantaneous*. During an impact, the contacted link is subjected to an *impulsive* constraint force in the normal direction of the link which instantaneously changes the normal direction link velocity in order to prevent it from continuing into the obstacle. This represents the discontinuous impact dynamics of the snake robot. We assume that the configuration $\boldsymbol{q}$ of the snake robot, which was defined in (10.15), is unaltered during an impact. We also assume that the obstacle friction forces cannot display impulsive behaviour, which means that we will disregard obstacle friction forces during an impact. Similar to the calculation of the continuous constraint forces, we will show in Section 10.6 that the problem of calculating the impulsive constraint forces during an impact can be formulated as a LCP. It will be seen in Section 10.7 that an impact triggers a *jump* in the state of the hybrid snake robot model.

The above description of the contact modelling approach is summarized by the following set of assumptions:

Assumption 10.1 *All obstacles have a circular shape.*

Assumption 10.2 *The distance between the edges of any two obstacles is greater than the link length* $2l$.

Assumption 10.3 *The friction coefficient between the snake robot and any obstacle is* $\mu_o \geq 0$.

Assumption 10.4 *An obstacle contact force acts on the CM of a link only. The link length is small so that contact torques about the link CM are negligible.*

Assumption 10.5 *Impacts between the snake robot and the obstacles are completely inelastic.*

Assumption 10.6 *All impacts are instantaneous in time and all impact forces are impulsive.*

Assumption 10.7 *During an impact, the configuration* $\boldsymbol{q}$ *of the snake robot remains unaltered, while the velocity* $\dot{\boldsymbol{q}}$ *will generally experience a jump.*

Assumption 10.8 *Obstacle friction forces are negligible during an impact.*

Remark 10.3 *The common approach when modelling mechanical systems with unilateral constraints is to calculate the direction of a constraint force with respect to the normal direction of the* constraint surface *(see Brogliato, 1999), i.e. the normal direction of the obstacles in this case. With the approach taken in this chapter, the shape of the obstacles does not have to be considered explicitly as we instead calculate the constraint forces with respect to the normal direction of the* contacted links*. This simplifies the equations of motion. Note that these two approaches produce similar constraint directions when the end point of a link is not in contact with an obstacle. To verify this, consider a snake robot link in contact with a circular obstacle. Since the link is tangent to the obstacle, the normal direction of the link and the obstacle must be equal. When the end point of a link, i.e. a joint, is in contact with an obstacle, however, both links attached to the joint are in contact with the obstacle. The approach in this chapter will then produce a normal constraint force on both links attached to the joint.*

Remark 10.4 *A consequence of modelling obstacle contact by a unilateral force on the contacted link, is that there is nothing preventing the foremost link (the head) of the snake robot from penetrating an obstacle head-on along its tangential direction. Furthermore, a consequence of Assumption 10.4 is*

that a link in theory can rotate 'into' an obstacle while its CM has zero normal direction velocity. These two consequences are results of the goal of keeping the mathematical model as simple as possible, but are not critical in practice. In particular, head-on collisions with the head of the snake robot can be avoided through the control strategy, e.g. by assuming that the head is equipped with distance sensors that enable the robot to actively avoid head-on collisions with obstacles. Link rotations 'into' an obstacle may only occur to a very small extent during obstacle-aided locomotion since this is mostly a forward gliding type of motion.

10.4 Detection of Obstacle Impacts and Detachments

The planar environment of the snake robot consists of k circular obstacles indexed by $j \in \{1, \ldots, k\}$. The global coordinates of the centre of obstacle j is denoted by $\left(x_{O_j}, y_{O_j}\right)$. The set $\boldsymbol{O}_j$ of points occupied by obstacle j is given by

$$\boldsymbol{O}_j = \left\{(x,y) \,|\, \left(x - x_{O_j}\right)^2 + \left(y - y_{O_j}\right)^2 \leq R_{O_j}^2 \right\}, \tag{10.19}$$

where R_{O_j} is the radius of obstacle j. By studying Fig. 2.1, it is easily seen that the set $\boldsymbol{L}_i$ of points occupied by link $i \in \{1, \ldots, N\}$ is given by

$$\boldsymbol{L}_i = \{(x,y) \,|x = x_i + s\cos\theta_i, y = y_i + s\sin\theta_i, s \in [-l, l]\}, \tag{10.20}$$

where (x_i,y_i) and θ_i are the CM coordinates and angle of link i, respectively. A collision between link i and obstacle j occurs whenever $\boldsymbol{L}_i \cap \boldsymbol{O}_j \neq \emptyset$, where $\emptyset$ denotes an empty set. There is no collision if $\boldsymbol{L}_i \cap \boldsymbol{O}_j = \emptyset$.

We now introduce a vector of *contact parameters*, $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]^T \in \mathbb{R}^N$. The contact parameter of link i, denoted by $\alpha_i \in \{-1, 0, 1\}$, is a discrete state value that determines if the link is in contact with an obstacle and also on which side of the link there is contact. As shown in Fig. 10.5, $\alpha_i = 1$ when the obstacle constraint force points along the positive link y axis, while $\alpha_i = -1$ when the obstacle constraint force points along the negative link y axis. We set $\alpha_i = 0$ when link i is *not* in contact with an obstacle.Whenever link i impacts an obstacle, we update the contact parameter α_i according to $\alpha_i = \widetilde{\alpha}\,(i)$, where the scalar function $\widetilde{\alpha}\,(i)$ is defined by

$$\widetilde{\alpha}\,(i) = -\operatorname{sgn}\left([0,1]\left(\boldsymbol{R}_{\text{link},i}^{\text{global}}\right)^T \left(\min_{j\in\{1,\ldots,k\}} \boldsymbol{r}_{L_i,O_j}\right)\right), \tag{10.21}$$

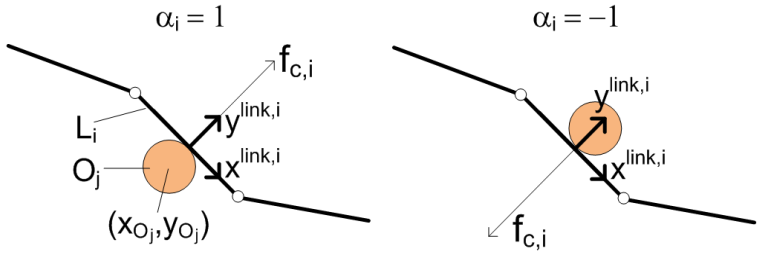


Figure 10.5: The value of the contact parameter of link i is $\alpha_i = 1$ when the constraint force points along the positive link y axis, and $\alpha_i = -1$ when the constraint force points along the negative link y axis. We set $\alpha_i = 0$ when there is no contact.

where $\boldsymbol{r}_{L_i,O_j} \in \mathbb{R}^2$ is the vector from link i to obstacle j, and $\boldsymbol{R}^{\text{global}}_{\text{link},i}$ was defined in (2.3) as the rotation matrix from the global frame to the local frame of link i. In (10.21), the global frame vector from link i to the *closest* obstacle is first found by use of the min operator, which is assumed to find the vector with the smallest *Euclidean norm*. Subsequently, this vector is transformed to the frame of link i using the rotation matrix $\boldsymbol{R}^{\text{global}}_{\text{link},i}$. Finally, the y component of this vector is extracted using the vector $[0, 1]$. The sign of the y component determines on which side of the link the obstacle is located.

10.5 The Continuous Dynamics of the Snake Robot during Constrained Motion

In this section, we present the model of the continuous constrained dynamics of the snake robot during sustained contact with a fixed set of obstacles. The section consists of three parts. In particular, Section 10.5.1 presents the equations describing the unilateral velocity constraints for the links that are in contact with an obstacle, while Section 10.5.2 presents the resulting equations of motion of the snake robot under the assumption that the obstacles are frictionless ($\mu_o = 0$). The equations of motion where obstacle friction is present ($\mu_o > 0$) are derived in Section 10.5.3. Note that the discontinuous dynamics occurring when the set of contacted links changes is treated in Section 10.6.

10.5.1 The Unilateral Constraints from the Obstacles

With reference to Fig. 10.5, the unilateral velocity constraint imposed on link i during contact with an obstacle may be compactly expressed as

$$\alpha_i v_{n,i} \geq 0, \tag{10.22}$$

where α_i is the contact parameter of link i and $v_{n,i}$ is the normal direction velocity of link i, i.e. the velocity of the CM of link i in the direction of the local link y axis. This constraint prevents sideways link motion towards (and thereby into) the obstacle. Using (2.3), the velocity constraint is given in the global frame as

$$\alpha_i \left(-\dot{x}_i \sin\theta_i + \dot{y}_i \cos\theta_i\right) \geq 0, \tag{10.23}$$

where $(\dot{x}_i, \dot{y}_i)$ and θ_i are the CM velocity and angle of link i, respectively. Hence, using notation from Section 2.3, the unilateral velocity constraints for all links may be expressed in matrix form as

$$\mathrm{diag}(\boldsymbol{\alpha}) \left(-\boldsymbol{S}_\theta \dot{\boldsymbol{X}} + \boldsymbol{C}_\theta \dot{\boldsymbol{Y}}\right) \geq \boldsymbol{0}. \tag{10.24}$$

By inserting (2.12) into (10.24) and rearranging we get

$$\mathrm{diag}(\boldsymbol{\alpha}) \boldsymbol{C}(\boldsymbol{q})\, \dot{\boldsymbol{q}} \geq \boldsymbol{0}, \tag{10.25}$$

where $\boldsymbol{q} \in \mathbb{R}^{N+2}$ was defined in (10.15) and $\boldsymbol{C}(\boldsymbol{q}) \in \mathbb{R}^{N \times (N+2)}$ is given by

$$\boldsymbol{C}(\boldsymbol{q}) = \left[-l\left(\boldsymbol{S}_\theta \boldsymbol{K}^T \boldsymbol{S}_\theta + \boldsymbol{C}_\theta \boldsymbol{K}^T \boldsymbol{C}_\theta\right), -\sin\boldsymbol{\theta}, \cos\boldsymbol{\theta}\right]. \tag{10.26}$$

We denote the number of contacted links by $m \in \{0, 1, \ldots, N\}$. In order to easily select the velocity constraints from (10.25) that correspond to contacted links, we define a *selection matrix* $\boldsymbol{S}_c(\boldsymbol{\alpha}) \in \mathbb{R}^{m \times N}$, which simply contains the m rows from the matrix $\mathrm{diag}(\boldsymbol{\alpha}) \in \mathbb{R}^{N \times N}$ that contain a nonzero element. With this selection matrix, we can write the velocity constraints for all links that are in contact with an obstacle as

$$\overline{\boldsymbol{C}}(\boldsymbol{q}, \boldsymbol{\alpha})\, \dot{\boldsymbol{q}} \geq \boldsymbol{0}, \tag{10.27}$$

where $\overline{\boldsymbol{C}}(\boldsymbol{q}, \boldsymbol{\alpha}) = \boldsymbol{S}_c(\boldsymbol{\alpha})\, \boldsymbol{C}(\boldsymbol{q}) \in \mathbb{R}^{m \times (N+2)}$.

The calculation of the obstacle contact forces in the next subsection requires the time derivative of the matrix $\overline{\boldsymbol{C}}(\boldsymbol{q}, \boldsymbol{\alpha})$, which is given by

$$\dot{\overline{\boldsymbol{C}}}(\boldsymbol{q}, \boldsymbol{\alpha}) = \boldsymbol{S}_c(\boldsymbol{\alpha}) \left[l\left(\boldsymbol{S}_\theta \widetilde{\boldsymbol{K}} \boldsymbol{C}_\theta - \boldsymbol{C}_\theta \widetilde{\boldsymbol{K}} \boldsymbol{S}_\theta\right), -\boldsymbol{C}_\theta \dot{\boldsymbol{\theta}}, -\boldsymbol{S}_\theta \dot{\boldsymbol{\theta}}\right], \tag{10.28}$$

where $\widetilde{\boldsymbol{K}} = \text{diag}(\dot{\boldsymbol{\theta}})\boldsymbol{K}^T - \boldsymbol{K}^T\,\text{diag}(\dot{\boldsymbol{\theta}})$. Note that this derivative is only valid over intervals where the set of contacted links remains constant. The derivative of $\overline{\boldsymbol{C}}\,(\boldsymbol{q}, \boldsymbol{\alpha})$ is not defined for time instants where an element of $\boldsymbol{\alpha}$ is changed.

The following rank property of the constraint matrix $\overline{\boldsymbol{C}}$ is important in order to uniquely determine the contact forces acting on the snake robot (see Section 10.5.2).

Property 10.1 *The matrix $\overline{\boldsymbol{C}}$ has full rank (*$\text{rank}\left(\overline{\boldsymbol{C}}\right) = m$*) for all $(\boldsymbol{q}, \boldsymbol{\alpha})$.*

Remark 10.5 *Due to the complexity of the elements in $\overline{\boldsymbol{C}}$, it is difficult to present a purely mathematical proof that Property 10.1 holds. However, we can argue from a physical perspective that this property must hold. Assume that* $\text{rank}\left(\overline{\boldsymbol{C}}\right) < m$*. This implies linear dependence between some of the m rows of $\overline{\boldsymbol{C}}$, i.e. there must exist a row of $\overline{\boldsymbol{C}}$, denoted $\overline{\boldsymbol{C}}_i$, such that*

$$\overline{\boldsymbol{C}}_i = \sum_{j \in \{1,\cdots,m\} \backslash \{i\}} k_j \overline{\boldsymbol{C}}_j, \tag{10.29}$$

where $k_j \in \mathbb{R}$. The scalar $\overline{\boldsymbol{C}}_i \dot{\boldsymbol{q}}$ is the magnitude of the normal direction velocity of link i, denoted by $|v_{n,i}|$. Multiplying (10.29) by $\dot{\boldsymbol{q}}$ therefore gives

$$|v_{n,i}| = \overline{\boldsymbol{C}}_i \dot{\boldsymbol{q}} = \sum_{j \in \{1,\cdots,m\} \backslash \{i\}} k_j \left|v_{n,j}\right|, \tag{10.30}$$

which states that the normal direction velocity of link i can be written as a linear combination of the normal direction velocities of all other contacted links of the snake robot. From a physical perspective, such a dependence could never occur unless all links are parallel since the couplings between the link velocities are given in terms of both normal and *tangential link velocities. In particular, (2.7) in Section 2.3 implies that the velocity of link i can be written in terms of the velocities of link $i-1$ and link $i+1$. Unless link $i-1$, i, and $i+1$ are parallel, this is a relationship involving both the normal* and *tangential velocities of link $i-1$ and link $i+1$. This contradicts (10.30) since the relationship in (10.30) only contains normal direction velocities. This leaves the case of parallel links ($\theta_1 = \theta_2 = \cdots = \theta_N$) as the only way for (10.30) to be true. A straightforward calculation of $\overline{\boldsymbol{C}}$ in e.g.* Matlab Symbolic Toolbox *shows that $\overline{\boldsymbol{C}}$ always has full rank when the link angles are equal. We can therefore conclude that $\overline{\boldsymbol{C}}$ never drops rank.*

This subsection can be summarized as follows. At any given time instant, the snake robot is in contact with m obstacles. The interaction between the robot and these m obstacles is modelled by imposing the unilateral velocity constraints in (10.27) on the m contacted links.

10.5.2 The Constrained Dynamics of the Snake Robot without Obstacle Friction

We will now use the unilateral velocity constraints in (10.27) to derive the resulting equations of motion of the snake robot. We assume that the m contact points between the links and the obstacles have *already* been established, i.e. we consider the continuous contact dynamics of the snake robot over a time interval where the set of contacted links remains fixed. We first consider the frictionless case in this subsection, followed by contact forces *with* friction in the next subsection.

Let us first assume that the m velocity constraints on the snake robot in (10.27) are *bilateral*, i.e. that they are given by

$$\overline{\boldsymbol{C}}\dot{\boldsymbol{q}} = \boldsymbol{0}, \tag{10.31}$$

These are called *Pfaffian* constraints and are modelled by adding a term to the equations of motion in (10.16) as follows (see Goldstein *et al.*, 2002):

$$\boldsymbol{M}\ddot{\boldsymbol{q}} = \boldsymbol{f}_u + \overline{\boldsymbol{C}}^T\boldsymbol{\lambda}. \tag{10.32}$$

The term $\overline{\boldsymbol{C}}^T\boldsymbol{\lambda}$ ensures compliance with the imposed velocity constraints, where $\boldsymbol{\lambda} \in \mathbb{R}^m$ is a vector of scalars known as *Lagrange multipliers* (Goldstein *et al.*, 2002). The Lagrange multipliers are important because multiplier λ_j equals the magnitude of the constraint force that ensures compliance with the jth constraint. This means that if the jth constraint in (10.31) corresponds to the velocity constraint on link i, then λ_j equals the magnitude of the constraint force $\boldsymbol{f}_{c,i}$ acting on link i.

We now argue that (10.32) also represent the equations of motion of the snake robot when the velocity constraints are *unilateral*, as in (10.27). This is quite obvious since the influence of a *unilateral* constraint on the snake robot when this constraint is active (i.e. when the unilateral constraint is preventing sideways motion of a link) must necessarily be identical to the influence that the corresponding *bilateral* constraint would have. In other words, the nature of the constraint (i.e. unilateral or bilateral) is not apparent when the constraint is active since it is only active in one direction at a time. The only difference between the bilateral and unilateral

case concerns the calculation of the constraint forces given by $\boldsymbol{\lambda}$. In the bilateral case, one may calculate $\boldsymbol{\lambda}$ directly by differentiating (10.31) with respect to time, inserting (10.32), and solving for $\boldsymbol{\lambda}$. In the unilateral case, however, the constraint forces must be calculated so that they comply with the directional requirements of the constraints, i.e. we require that $\boldsymbol{\lambda} \geq \mathbf{0}$. This means that the constraint forces can only point *away* from the obstacles.

We will now handle this directional requirement by employing the theory of *linear complementarity problems* (LCPs) introduced in Section 10.1.2. This approach is based on the work in Lötstedt (1982). The key observation is that the normal direction velocity of a contacted link and the corresponding constraint force are subjected to a *complementarity condition*. If the normal direction velocity is non-zero (i.e. the link is moving away from the obstacle), then the corresponding constraint force must be zero. Likewise, the normal direction velocity must be zero if the corresponding constraint force is non-zero. This complementarity condition also applies to the normal direction *acceleration* of a contacted link and the corresponding constraint force.

From the above discussion, the equations of motion of the snake robot that include *unilateral* constraint forces from *frictionless* obstacles are given by

$$\boldsymbol{M}\ddot{\boldsymbol{q}} = \boldsymbol{f}_u + \overline{\boldsymbol{C}}^T\boldsymbol{\lambda}, \tag{10.33}$$

$$\overline{\boldsymbol{C}}\dot{\boldsymbol{q}} \geq \mathbf{0}, \quad \boldsymbol{\lambda} \geq \mathbf{0}, \quad \boldsymbol{\lambda}^T\overline{\boldsymbol{C}}\dot{\boldsymbol{q}} = 0. \tag{10.34}$$

The vector $\overline{\boldsymbol{C}}\dot{\boldsymbol{q}} \in \mathbb{R}^m$ contains the normal direction velocity of each contacted link in the direction *away* from each obstacle. The normal direction *acceleration* of each contacted link in the direction *away* from each obstacle, denoted by $\overline{\boldsymbol{a}}_n \in \mathbb{R}^m$, is given by

$$\overline{\boldsymbol{a}}_n = \frac{d}{dt}\left(\overline{\boldsymbol{C}}\dot{\boldsymbol{q}}\right) = \overline{\boldsymbol{C}}\ddot{\boldsymbol{q}} + \dot{\overline{\boldsymbol{C}}}\dot{\boldsymbol{q}} \geq \mathbf{0}. \tag{10.35}$$

By solving (10.33) for $\ddot{\boldsymbol{q}}$ and inserting into (10.35), we finally arrive at the following model of the continuous contact dynamics of the snake robot with frictionless obstacles:

$$\boldsymbol{M}\ddot{\boldsymbol{q}} = \boldsymbol{f}_u + \overline{\boldsymbol{C}}^T\boldsymbol{\lambda}, \tag{10.36}$$

$$\begin{gathered}\overline{\boldsymbol{a}}_n = \overline{\boldsymbol{C}}\boldsymbol{M}^{-1}\boldsymbol{f}_u + \dot{\overline{\boldsymbol{C}}}\dot{\boldsymbol{q}} + \overline{\boldsymbol{C}}\boldsymbol{M}^{-1}\overline{\boldsymbol{C}}^T\boldsymbol{\lambda}, \\ \overline{\boldsymbol{a}}_n \geq \mathbf{0}, \quad \boldsymbol{\lambda} \geq \mathbf{0}, \quad \boldsymbol{\lambda}^T\overline{\boldsymbol{a}}_n = 0.\end{gathered} \tag{10.37}$$

Equation (10.37) is in the form of the general LCP given in (10.2) with

$\boldsymbol{A} = \overline{\boldsymbol{C}}\boldsymbol{M}^{-1}\overline{\boldsymbol{C}}^T$ and $\boldsymbol{a} = \overline{\boldsymbol{C}}\boldsymbol{M}^{-1}\boldsymbol{f}_u + \dot{\overline{\boldsymbol{C}}}\dot{\boldsymbol{q}}$. In order to calculate the dynamics of the snake robot at any given time instant, this LCP must be solved for the unknowns $\overline{\boldsymbol{a}}_n$ and $\boldsymbol{\lambda}$ subject to the complementarity conditions. The calculated $\boldsymbol{\lambda}$ gives the constraint forces from the obstacles and is plugged into (10.36) in order to calculate $\ddot{\boldsymbol{q}}$.

In order to determine the existence and uniqueness properties of the LCP in (10.37), we will need the following result, which is proved in Bernstein (2009) (Proposition 8.1.2, item *xiii*):

Proposition 10.1 *Let $\boldsymbol{M} \in \mathbb{R}^{N\times N}$ be a* symmetrical *and* positive definite *matrix ($\boldsymbol{M} > 0$), and $\boldsymbol{C} \in \mathbb{R}^{m\times N}$ be a matrix of full rank (*$\operatorname{rank}(\boldsymbol{C}) = m$*). Then $\boldsymbol{C}\boldsymbol{M}\boldsymbol{C}^T > 0$.*

We can now state the following result concerning the existence and uniqueness properties of the LCP in (10.37):

Theorem 10.2 *The LCP in (10.37) always has a unique solution $(\overline{\boldsymbol{a}}_n, \boldsymbol{\lambda})$.*

Proof. From Theorem 10.1, the proof is complete if we can show that $\boldsymbol{A} = \overline{\boldsymbol{C}}\boldsymbol{M}^{-1}\overline{\boldsymbol{C}}^T$ is a P-matrix. Since $\overline{\boldsymbol{C}}$ has full rank (by Property 10.1) and $\boldsymbol{M} = \boldsymbol{M}^T > 0$ (the inertia matrix is always symmetrical and positive definite), we have from Proposition 10.1 that $\boldsymbol{A} = \overline{\boldsymbol{C}}\boldsymbol{M}^{-1}\overline{\boldsymbol{C}}^T > 0$. Since $\boldsymbol{A}$ is symmetrical and positive definite, it must also be a P-matrix. This completes the proof. ■

Remark 10.6 *The LCP in (10.37) can be regarded as the problem of, at a given time instant, determining which obstacle contacts that will persist onto the next time instant, and which will not. A link contact will persist onto the next time instant if the corresponding value of $\boldsymbol{\lambda}$ is non-zero. If, however, the value of $\overline{\boldsymbol{a}}_n$ for a link contact is non-zero, then the link will detach from the obstacle.*

10.5.3 The Constrained Dynamics of the Snake Robot with Obstacle Friction

We employ a Coulomb friction model in order to describe the gliding friction force between the links and the obstacles. In accordance with Fig. 10.4, we define the obstacle friction force on link i as

$$\boldsymbol{f}_{\mu,i} = \begin{bmatrix} \boldsymbol{f}_{\mu,x,i} \\ \boldsymbol{f}_{\mu,y,i} \end{bmatrix} = -\mu_o \begin{bmatrix} \cos\theta_i \\ \sin\theta_i \end{bmatrix} \operatorname{sgn}(v_{t,i}) \left|\boldsymbol{f}_{c,i}\right|, \tag{10.38}$$

where $\mu_o \geq 0$ is the Coulomb friction coefficient of the obstacles, $\left|\boldsymbol{f}_{c,i}\right|$ is the magnitude of the obstacle constraint force acting on link i, and $v_{t,i}$ is the tangential direction velocity of link i, i.e. the velocity of the CM of link i in the direction of the local link x axis. The obstacle friction forces on all the links can be expressed as

$$\boldsymbol{f}_{\mu} = \begin{bmatrix} \boldsymbol{f}_{\mu,x} \\ \boldsymbol{f}_{\mu,y} \end{bmatrix} = -\mu_o \begin{bmatrix} \boldsymbol{C}_\theta \\ \boldsymbol{S}_\theta \end{bmatrix} \operatorname{diag}\left(\operatorname{sgn}\left(\boldsymbol{v}_t\right)\right) \left|\boldsymbol{f}_c\right|, \tag{10.39}$$

where $\boldsymbol{f}_{\mu,x} = \left[\boldsymbol{f}_{\mu,x,1}, \ldots, \boldsymbol{f}_{\mu,x,N}\right]^T \in \mathbb{R}^N$ and $\boldsymbol{f}_{\mu,y} = \left[\boldsymbol{f}_{\mu,y,1}, \ldots, \boldsymbol{f}_{\mu,y,N}\right]^T \in \mathbb{R}^N$ contain the obstacle friction forces on the links in the global x and y direction, respectively, $\left|\boldsymbol{f}_c\right| = \left[\left|\boldsymbol{f}_{c,1}\right|, \ldots, \left|\boldsymbol{f}_{c,N}\right|\right]^T \in \mathbb{R}^N$ contains the magnitude of the constraint force on each link, and $\boldsymbol{v}_t = [v_{t,1}, \ldots, v_{t,N}]^T \in \mathbb{R}^N$ contains the tangential link velocities in the local x direction of each link.

The mapping between the friction forces and the acceleration of the configuration vector $\ddot{\boldsymbol{q}}$ is identical to the mapping between the ground friction forces $\boldsymbol{f}_R$ and $\ddot{\boldsymbol{q}}$ given in (10.18) since $\boldsymbol{f}_R$ and $\boldsymbol{f}_\mu$ both act on the CM of the links. By using the easily verifiable relation $\left|\boldsymbol{f}_c\right| = \left|\boldsymbol{S}_c\left(\boldsymbol{\alpha}\right)\right|^T \boldsymbol{\lambda}$, where $\boldsymbol{S}_c\left(\boldsymbol{\alpha}\right)$ is the selection matrix introduced in (10.27) and $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers introduced in (10.32), we may write the accelerations due to the obstacle friction forces, temporarily denoted $\ddot{\boldsymbol{q}}_{f_\mu}$, as

$$\ddot{\boldsymbol{q}}_{f_\mu} = -\mu_o \boldsymbol{\Lambda} \boldsymbol{\lambda}, \tag{10.40}$$

where $\boldsymbol{\Lambda} \in \mathbb{R}^{(N+2)\times m}$ is given by

$$\boldsymbol{\Lambda}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\alpha}) = \begin{bmatrix} l\boldsymbol{S}_\theta \boldsymbol{K} & -l\boldsymbol{C}_\theta \boldsymbol{K} \\ \boldsymbol{e}^T & \boldsymbol{0}_{1\times N} \\ \boldsymbol{0}_{1\times N} & \boldsymbol{e}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{C}_\theta \\ \boldsymbol{S}_\theta \end{bmatrix} \operatorname{diag}\left(\operatorname{sgn}\left(\boldsymbol{v}_t\right)\right) \left|\boldsymbol{S}_c\left(\boldsymbol{\alpha}\right)\right|^T. \tag{10.41}$$

By adding (10.40) to the equations of motion in (10.36) and following the exact same approach that led to the LCP in (10.37), we get the following equations describing the continuous contact dynamics of the snake robot that include obstacle friction forces:

$$\boldsymbol{M}\ddot{\boldsymbol{q}} = \boldsymbol{f}_u + \left(\overline{\boldsymbol{C}}^T - \mu_o \boldsymbol{\Lambda}\right) \boldsymbol{\lambda}, \tag{10.42}$$

$$\begin{gathered} \overline{\boldsymbol{a}}_n = \overline{\boldsymbol{C}} \boldsymbol{M}^{-1} \boldsymbol{f}_u + \dot{\overline{\boldsymbol{C}}} \dot{\boldsymbol{q}} + \overline{\boldsymbol{C}} \boldsymbol{M}^{-1} \left(\overline{\boldsymbol{C}}^T - \mu_o \boldsymbol{\Lambda}\right) \boldsymbol{\lambda}, \\ \overline{\boldsymbol{a}}_n \geq \boldsymbol{0}, \quad \boldsymbol{\lambda} \geq \boldsymbol{0}, \quad \boldsymbol{\lambda}^T \overline{\boldsymbol{a}}_n = 0. \end{gathered} \tag{10.43}$$

We again identify (10.43) as a LCP of the general form given in (10.2) with $\boldsymbol{A} = \overline{\boldsymbol{C}}\boldsymbol{M}^{-1}(\overline{\boldsymbol{C}}^T - \mu_o\boldsymbol{\Lambda})$ and $\boldsymbol{a} = \overline{\boldsymbol{C}}\boldsymbol{M}^{-1}\boldsymbol{f}_u + \dot{\overline{\boldsymbol{C}}}\dot{\boldsymbol{q}}$. When obstacle friction is present ($\mu_o > 0$), we can no longer guarantee existence and uniqueness of the solution to the LCP in (10.43) since it is no longer evident that the matrix $\boldsymbol{A}$ is a P-matrix ($\boldsymbol{A}$ is no longer symmetrical, which complicates the P-matrix check). This existence and uniqueness issue is a general and well-known problem for acceleration LCPs that include Coulomb friction (see e.g. Brogliato, 1999; Lötstedt, 1981; Mason and Wang, 1988; Song *et al.*, 2000; Trinkle *et al.*, 1997). Due to the complexity of determining if $\boldsymbol{A}$ is a P-matrix, we are unable to provide an analytical upper bound of μ_o that must be satisfied to guarantee existence and uniqueness of the solution to the LCP in (10.43). However, we can still state the following result:

Theorem 10.3 *For a given $(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\alpha})$, there exists a $\mu_o^* > 0$ such that the LCP in (10.43) has a unique solution $(\overline{\boldsymbol{a}}_n, \boldsymbol{\lambda})$ for $\mu_o \in [0, \mu_o^*)$.*

Proof. Recall from Section 10.1.2 that $\boldsymbol{A} = \overline{\boldsymbol{C}}\boldsymbol{M}^{-1}\left(\overline{\boldsymbol{C}}^T - \mu_o\boldsymbol{\Lambda}\right)$ is a P-matrix if all principal minors of $\boldsymbol{A}$ are positive. We know from Theorem 10.2 that the LCP in (10.43) *always* has a unique solution for $\mu_o = 0$ since $\boldsymbol{A}$ is a P-matrix in this case. All principal minors of $\boldsymbol{A}$ must therefore be positive for $\mu_o = 0$. Assume now that we increase μ_o until a principal minor of $\boldsymbol{A}$ becomes zero, and denote the corresponding value of the friction coefficient by $\mu_o^* > 0$. It is then evident that the P-matrix property of $\boldsymbol{A}$ must be preserved for $\mu_o < \mu_o^*$, i.e. existence and uniqueness of the solution to the LCP in (10.43) must hold for $\mu_o < \mu_o^*$. This completes the proof. ■

Remark 10.7 *During our numerical treatments of the LCP in (10.43) so far, we have not yet encountered a single instance where $\boldsymbol{A}$ has failed to be a P-matrix. We therefore conjecture that μ_o must have an unrealistically high value in order for $\boldsymbol{A}$ to no longer be a P-matrix, and that the LCP in (10.43) will always be uniquely solvable during our simulations of the snake robot.*

Remark 10.8 *As implied by Theorem 10.1, the matrix $\boldsymbol{A}$ being a P-matrix is only required to guarantee existence and uniqueness of the solution to a LCP for* all *data vectors $\boldsymbol{a}$. The LCP in (10.43) may therefore have a unique solution for a given $(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\alpha})$ even if $\boldsymbol{A}$ is not a P-matrix (Cottle* et al.*, 1992).*

10.6 The Discontinuous Dynamics of the Snake Robot during Obstacle Impacts and Detachments

In this section, we present the model of the discontinuous dynamics of the snake robot occurring when the set of links in contact with an obstacle changes. We first present the dynamics during obstacle impacts in Section 10.6.1, followed by the dynamics during obstacle detachments in Section 10.6.2.

10.6.1 The Discontinuous Dynamics of the Snake Robot during Obstacle Impacts

By Assumption 10.5, an *inelastic impact* occurs when a link (that was previously not in contact with an obstacle) comes into contact with an obstacle. By Assumption 10.6, the impact is instantaneous in time and the resulting impact forces are impulsive, resulting in a discontinuous *jump* in the velocity of the snake robot. Following an approach in Brogliato (1999), we model the impact as

$$\boldsymbol{M}\left(\boldsymbol{q}^{+}\right)\dot{\boldsymbol{q}}^{+}-\boldsymbol{M}\left(\boldsymbol{q}^{-}\right)\dot{\boldsymbol{q}}^{-}=\boldsymbol{F}_{\text{impulse}}, \tag{10.44}$$

where $\boldsymbol{F}_{\text{impulse}} \in \mathbb{R}^{N+2}$ denotes the generalized impulsive impact forces and $\boldsymbol{q}^{-}$, $\dot{\boldsymbol{q}}^{-}$, $\boldsymbol{q}^{+}$, and $\dot{\boldsymbol{q}}^{+}$ denote the generalized coordinates and velocities immediately *before* and *after* the impact, respectively. This superscript notation is commonly used when modelling hybrid systems, as described in Section 10.1.1. By Assumption 10.7, the configuration of the snake robot is unaltered during an impact ($\boldsymbol{q}^{+}=\boldsymbol{q}^{-}$). This means that $\boldsymbol{M}\left(\boldsymbol{q}^{-}\right)=\boldsymbol{M}\left(\boldsymbol{q}^{+}\right)=\boldsymbol{M}\left(\boldsymbol{q}\right)$. By Assumption 10.8, the impact forces are frictionless. Following the same argumentation that led to the expression in (10.36), we can now rewrite (10.44) as

$$\boldsymbol{M}\left(\boldsymbol{q}\right)\left(\dot{\boldsymbol{q}}^{+}-\dot{\boldsymbol{q}}^{-}\right)=\overline{\boldsymbol{C}}^{T}\left(\boldsymbol{q},\boldsymbol{\alpha}^{+}\right)\boldsymbol{\lambda}, \tag{10.45}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{m}$ is a vector of *impulsive constraint forces*. Note that the constraint matrix $\overline{\boldsymbol{C}}$, which was defined in (10.27), depends on the contact parameter vector *after* the impact, i.e. $\boldsymbol{\alpha}^{+}$. This is because the contact parameter of the impacted link is zero immediately *before* the impact (i.e. if link i impacts an obstacle, then $\alpha_i^{-}=0$). In order to include this link in the impact dynamics, we must calculate $\overline{\boldsymbol{C}}$ based on the value of α_i *after*

the impact, i.e. $\alpha_i^+ = \widetilde{\alpha}(i)$, where $\widetilde{\alpha}(i)$ is given by (10.21). Note also that $\overline{\boldsymbol{C}}(\boldsymbol{q}, \boldsymbol{\alpha}^-) \in \mathbb{R}^{(m-1)\times(N+2)}$, while $\overline{\boldsymbol{C}}(\boldsymbol{q}, \boldsymbol{\alpha}^+) \in \mathbb{R}^{m\times(N+2)}$.

We will now calculate the impulsive constraint forces $\boldsymbol{\lambda}$ and the post-impact velocity $\dot{\boldsymbol{q}}^+$ by following an approach presented in Schaft and Schumacher (2000). The post-impact velocity and the impulsive constraint forces are naturally subjected to the same complementarity conditions as given in (10.34). We therefore have that

$$\overline{\boldsymbol{C}}(\boldsymbol{q}, \boldsymbol{\alpha}^+)\,\dot{\boldsymbol{q}}^+ \geq \boldsymbol{0}, \quad \boldsymbol{\lambda} \geq \boldsymbol{0}, \quad \boldsymbol{\lambda}^T \overline{\boldsymbol{C}}(\boldsymbol{q}, \boldsymbol{\alpha}^+)\,\dot{\boldsymbol{q}}^+ = 0. \tag{10.46}$$

Solving (10.45) for $\dot{\boldsymbol{q}}^+$ and premultiplying by $\overline{\boldsymbol{C}}(\boldsymbol{q}, \boldsymbol{\alpha}^+)$ gives

$$\overline{\boldsymbol{C}}\dot{\boldsymbol{q}}^+ = \overline{\boldsymbol{C}}\dot{\boldsymbol{q}}^- + \overline{\boldsymbol{C}}\boldsymbol{M}^{-1}\overline{\boldsymbol{C}}^T\boldsymbol{\lambda}. \tag{10.47}$$

Denoting the normal direction velocities of each of the contacted links (in the direction *away* from each obstacle) by the vector $\overline{\boldsymbol{v}}_n \in \mathbb{R}^m$, we may combine (10.46) and (10.47) into the following LCP describing the impact dynamics of the snake robot:

$$\begin{array}{c} \overline{\boldsymbol{v}}_n^+ = \overline{\boldsymbol{v}}_n^- + \overline{\boldsymbol{C}}\boldsymbol{M}^{-1}\overline{\boldsymbol{C}}^T\boldsymbol{\lambda}, \\ \overline{\boldsymbol{v}}_n^+ \geq \boldsymbol{0}, \quad \boldsymbol{\lambda} \geq \boldsymbol{0}, \quad \boldsymbol{\lambda}^T\overline{\boldsymbol{v}}_n^+ = 0. \end{array} \tag{10.48}$$

The LCP in (10.48) is in the general form of the LCP given in (10.2) with $\boldsymbol{A} = \overline{\boldsymbol{C}}\boldsymbol{M}^{-1}\overline{\boldsymbol{C}}^T$ and $\boldsymbol{a} = \overline{\boldsymbol{v}}_n^-$, and must be solved for the unknowns $\overline{\boldsymbol{v}}_n^+$ and $\boldsymbol{\lambda}$. Subsequently, the post-impact velocity is found by solving (10.45) for $\dot{\boldsymbol{q}}^+$ and inserting the calculated $\boldsymbol{\lambda}$. The following result concerns the existence and uniqueness properties of the LCP in (10.48):

Theorem 10.4 *The LCP in (10.48) always has a unique solution* $(\overline{\boldsymbol{v}}_n^+, \boldsymbol{\lambda})$.

Proof. The proof is identical to the proof of Theorem 10.2. ■

This subsection is now summarized. The discontinuous impact dynamics of the snake robot when link i impacts an obstacle and the state immediately before the impact is $\left(\boldsymbol{q}^-, \dot{\boldsymbol{q}}^-, \boldsymbol{\alpha}^-\right)$, is given by

$$\boldsymbol{q}^+ = \boldsymbol{q}^-, \tag{10.49a}$$

$$\dot{\boldsymbol{q}}^+ = \dot{\boldsymbol{q}}^- + \boldsymbol{M}^{-1}(\boldsymbol{q})\,\overline{\boldsymbol{C}}^T\left(\boldsymbol{q}, \boldsymbol{\alpha}^+\right)\boldsymbol{\lambda}, \tag{10.49b}$$

$$\alpha_j^+ = \begin{cases} \widetilde{\alpha}(i) & \text{when} \quad j = i \\ \alpha_j^- & \text{when} \quad j \neq i \end{cases}, \tag{10.49c}$$

where $j \in \{1, \ldots, N\}$, $\widetilde{\alpha}(i)$ is given by (10.21), and $\boldsymbol{\lambda}$ is calculated from the LCP in (10.48).

10.6.2 The Discontinuous Dynamics of the Snake Robot during Obstacle Detachments

If, at any time instant, the normal direction acceleration vector $\overline{\boldsymbol{a}}_n$ in the solution to the LCP in (10.43) contains a non-zero element, then the link corresponding to this non-zero element will accelerate away from the obstacle, meaning that the link will *detach* from the obstacle. The detachment dynamics is trivial compared to the impact dynamics since it only involves setting the contact parameter of the detached link to zero. In other words, whenever $\alpha_i \neq 0$ and the collision detection mechanism described in Section 10.4 detects that link i no longer overlaps with an obstacle, then the state of the snake robot is updated according to the discontinuous detachment dynamics given by

$$\boldsymbol{q}^+ = \boldsymbol{q}^-, \tag{10.50a}$$

$$\dot{\boldsymbol{q}}^+ = \dot{\boldsymbol{q}}^-, \tag{10.50b}$$

$$\alpha_j^+ = \begin{cases} 0 & \text{when} \quad j = i \\ \alpha_j^- & \text{when} \quad j \neq i \end{cases}, \tag{10.50c}$$

where $j \in \{1, \ldots, N\}$.

10.7 The Complete Hybrid Model of the Snake Robot in an Obstacle Environment

In this section, we employ the framework of *hybrid dynamical systems*, which was described in Section 10.1.1, in order to encapsulate the *continuous* dynamics from Section 10.5 and the *discontinuous* dynamics from Section 10.6 into a single *hybrid* model, or a *hybrid plant*. We will denote the hybrid model as a *plant* to distinguish it from a hybrid *controller* that we will present in Chapter 12. In the following subsections, we first present the *jump set* $\boldsymbol{D}_p$, *jump map* $\boldsymbol{G}_p$, *flow set* $\boldsymbol{C}_p$, and *flow map* $\boldsymbol{F}_p$ of the plant, followed by a summary of the hybrid plant in the last subsection. We define the state vector of the hybrid plant as

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \\ \boldsymbol{\alpha} \end{bmatrix} \in \mathbb{R}^{3N+4}. \tag{10.51}$$

10.7.1 The Jump Set

A *jump* in the state vector $\boldsymbol{x}$ of the plant occurs when a link impacts an obstacle (jump in $\dot{\boldsymbol{q}}$ and $\boldsymbol{\alpha}$) or when a link detaches from an obstacle (jump in $\boldsymbol{\alpha}$). By employing the notation from Section 10.4, the *jump set* corresponding to an *impact* between link i and an obstacle may be expressed as

$$\boldsymbol{D}_{L_i}^{\text{Impact}} = \{\boldsymbol{x} | \boldsymbol{L}_i \cap \boldsymbol{O}_j \neq \emptyset, j \in \{1, \ldots, k\}, \widetilde{\alpha}(i) \boldsymbol{C}_i \dot{\boldsymbol{q}} < 0\}, \tag{10.52}$$

where $\widetilde{\alpha}(i)$ is given by (10.21) and $\boldsymbol{C}_i$ denotes the ith row of the matrix $\boldsymbol{C}(\boldsymbol{q})$ in (10.26). We use $\widetilde{\alpha}(i)$ instead of α_i in (10.52) because $\alpha_i = 0$ before the impact has taken place. The *jump set* corresponding to link i *detaching* from an obstacle may be expressed as

$$\boldsymbol{D}_{L_i}^{\text{Detach}} = \{\boldsymbol{x} | \boldsymbol{L}_i \cap \boldsymbol{O}_j = \emptyset, j \in \{1, \cdots, k\}, \alpha_i \neq 0\}. \tag{10.53}$$

The jump sets comprising the *impacts* and the *detachments* of all the links, respectively, are given by

$$\boldsymbol{D}^{\text{Impact}} = \bigcup_{i \in \{1, \cdots, N\}} \boldsymbol{D}_{L_i}^{\text{Impact}}, \quad \boldsymbol{D}^{\text{Detach}} = \bigcup_{i \in \{1, \cdots, N\}} \boldsymbol{D}_{L_i}^{\text{Detach}}. \tag{10.54}$$

The complete jump set of the hybrid plant may now be compactly expressed as

$$\boldsymbol{D}_p = \boldsymbol{D}^{\text{Impact}} \cup \boldsymbol{D}^{\text{Detach}}. \tag{10.55}$$

10.7.2 The Jump Map

The *jump map* corresponding to the *impact* between link i and an obstacle is presented in (10.49), while the *jump map* corresponding to link i *detaching* from an obstacle is presented in (10.50). By combining (10.49) and (10.50), the complete jump map of the plant can be expressed as

$$\boldsymbol{x}^+ = \boldsymbol{G}_p(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{q}^+ \\ \dot{\boldsymbol{q}}^+ \\ \boldsymbol{\alpha}^+ \end{bmatrix} \quad \text{for all } \boldsymbol{x} \in \boldsymbol{D}_p, \tag{10.56}$$

where

$$\boldsymbol{q}^{+} = \boldsymbol{q}^{-}, \tag{10.57a}$$

$$\dot{\boldsymbol{q}}^{+} = \begin{cases} \dot{\boldsymbol{q}}^{-} + \boldsymbol{M}^{-1}(\boldsymbol{q})\,\overline{\boldsymbol{C}}^{T}(\boldsymbol{q},\boldsymbol{\alpha}^{+})\,\boldsymbol{\lambda} & \text{when} \quad \boldsymbol{x} \in \boldsymbol{D}^{\text{Impact}} \\ \dot{\boldsymbol{q}}^{-} & \text{when} \quad \boldsymbol{x} \notin \boldsymbol{D}^{\text{Impact}} \end{cases} \tag{10.57b}$$

$$\alpha_i^{+} = \begin{cases} \widetilde{\alpha}(i) & \text{when} \quad \boldsymbol{x} \in \boldsymbol{D}_{L_i}^{\text{Impact}} \\ 0 & \text{when} \quad \boldsymbol{x} \in \boldsymbol{D}_{L_i}^{\text{Detach}} \\ \alpha_i^{-} & \text{when} \quad \boldsymbol{x} \notin \left(\boldsymbol{D}_{L_i}^{\text{Impact}} \cup \boldsymbol{D}_{L_i}^{\text{Detach}}\right) \end{cases}. \tag{10.57c}$$

The value of $\widetilde{\alpha}(i)$ is given by (10.21) and $\boldsymbol{\lambda}$ is calculated from the LCP in (10.48).

10.7.3 The Flow Set

We define the *flow set* of the plant so that the state vector $\boldsymbol{x}$ always flows as long as the *jump set* is empty. The flow set is therefore given as

$$\boldsymbol{C}_p = \{\boldsymbol{x} | \boldsymbol{x} \notin \boldsymbol{D}_p\}. \tag{10.58}$$

10.7.4 The Flow Map

The *flow map* of $\boldsymbol{q}$ is simply $\dot{\boldsymbol{q}}$ and the *flow map* of $\dot{\boldsymbol{q}}$ is given by (10.42). The contact vector $\boldsymbol{\alpha}$ remains unchanged between jumps of $\boldsymbol{x}$, which means that the *flow map* of $\boldsymbol{\alpha}$ is the zero vector. The complete flow map of the plant is given by

$$\dot{\boldsymbol{x}} = \boldsymbol{F}_p(\boldsymbol{x},\boldsymbol{u}) = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \ddot{\boldsymbol{q}} \\ \boldsymbol{0}_{N\times 1} \end{bmatrix} \quad \text{for all } \boldsymbol{x} \in \boldsymbol{C}_p, \tag{10.59}$$

where

$$\ddot{\boldsymbol{q}} = \boldsymbol{M}^{-1}\left(\boldsymbol{f}_u(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{u}) + \left(\overline{\boldsymbol{C}}^{T} - \mu_o \boldsymbol{\Lambda}\right)\boldsymbol{\lambda}\right), \tag{10.60}$$

and $\boldsymbol{\lambda}$ is calculated from the LCP in (10.43).

10.7.5 Summary of the Complete Hybrid Plant

In accordance with Section 10.1.1, the complete hybrid model of the plant is written

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{F}_p(\boldsymbol{x},\boldsymbol{u}) \quad \text{for all} \quad \boldsymbol{x} \in \boldsymbol{C}_p, \\ \boldsymbol{x}^{+} &= \boldsymbol{G}_p(\boldsymbol{x}) \quad \text{for all} \quad \boldsymbol{x} \in \boldsymbol{D}_p. \end{aligned} \tag{10.61}$$

The evolution of the state vector of the plant has the following existence and uniqueness properties:

Theorem 10.5 *Given a control input* $\boldsymbol{u}$*, the evolution of the state* $\boldsymbol{x}$ *of the hybrid plant in (10.61) from any initial state can always be uniquely determined when the obstacles are frictionless* ($\mu_o = 0$)*. With obstacle friction, there exists a* $\mu_o^* > 0$ *such that existence and uniqueness of the evolution of* $\boldsymbol{x}$ *is guaranteed for* $\mu_o \in [0, \mu_o^*)$*, but not guaranteed for* $\mu_o \geq \mu_o^*$*.*

Proof. From (10.58), the *flow* and *jump set* are mutually exclusive, so we can always uniquely determine whether $\boldsymbol{x}$ should flow or jump. By Theorem 10.4, the *jump map* of $\boldsymbol{x}$ is always unique. By Theorem 10.2, the *flow map* of $\boldsymbol{x}$ is always unique with frictionless obstacles. By Theorem 10.3, there exists a $\mu_o^* > 0$ such that the *flow map* of $\boldsymbol{x}$ is always unique when $\mu_o \in [0, \mu_o^*)$. This completes the proof. ■

10.8 Simulation Study: Comparison of the Hybrid Model with Previous Experimental and Simulation Results

In order to investigate the validity of the hybrid plant proposed in (10.61), we have compared simulation results from this hybrid plant with experimental and simulation results presented in Transeth *et al.* (2008*a*). The simulator of the hybrid plant in (10.61) was implemented in *Matlab R2008b* on a laptop running *Windows XP*. The continuous dynamics of the plant was calculated with the *ode45* solver in Matlab with a relative and absolute error tolerance of 10^{-3}.

The work in Transeth *et al.* (2008*a*), which was described in Section 1.3, presents experimental results of obstacle-aided locomotion and also simulation results from a model developed based on the framework of nonsmooth dynamics. The joints of both the physical and the simulated snake robot in Transeth *et al.* (2008*a*) were controlled according to the gait pattern lateral undulation that was defined in (4.32). In particular, joint i of the snake robot was controlled according to the reference

$$\phi_{i,\mathrm{ref}} = \alpha \sin\left(\omega t + (i-1)\,\delta\right), \tag{10.62}$$

with $i \in \{1, \ldots, N-1\}$, $\alpha = 40°$, $\omega = 80°/s$, and $\delta = 50°$. Furthermore, the parameters characterizing the snake robot in Transeth *et al.* (2008*a*) were $N = 11$, $l = 0.061$ m, $m = 0.682$ kg, and $J = 0.0013$ kgm^2. Obstacles

of diameter 25 cm were placed in accordance with the expected motion of the robot (see Transeth *et al.* (2008*a*) for details). We implemented the plant in (10.61) in accordance with the above parameters and set the control input $\boldsymbol{u}$ according to the joint controller in (4.33) to control the joints of the robot according to (10.62) with $k_p = 20$ and $k_d = 5$.

The experimental and simulation results from Transeth *et al.* (2008*a*) are reprinted in Fig. 10.6, while the corresponding simulation results from the hybrid plant proposed in this chapter are presented in Fig. 10.7. We see that there is a close resemblance between the results from Transeth *et al.* (2008*a*) and the simulation results based on the plant in (10.61). The y direction amplitude in Fig. 10.6(b) and Fig. 10.7(b) are slightly different because the model in Transeth *et al.* (2008*a*) also considers the width of each link, while the model in this chapter assumes the width to be infinitesimal. In our opinion, the simulation results support the conjecture that, despite its simplifying assumptions, the model proposed in this chapter captures the essential part of the dynamics of a snake robot interacting with obstacles.

10.9 Chapter Summary

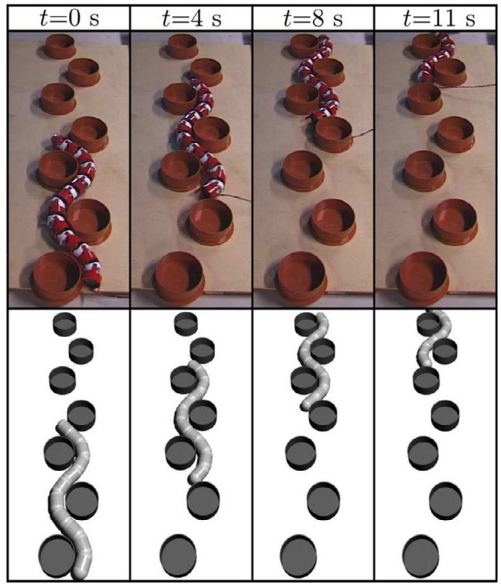
This chapter is summarized as follows:

- We have proposed an isotropic Coulomb ground friction model that also considers the ground friction torque on the links. The unconstrained (no obstacles) model of the snake robot with this new friction model is given in (10.16).

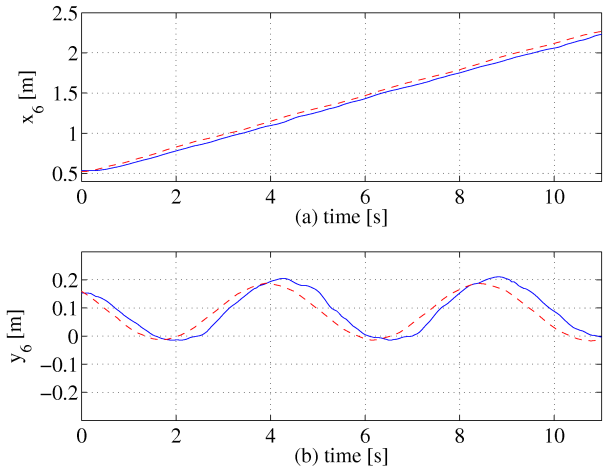
- We have extended the unconstrained model in (10.16) to include contact forces from external obstacles in the environment around the robot. In particular:

  - Under Assumption 10.1 - Assumption 10.8 presented in Section 10.3, the motion of the snake robot is constrained according to the velocity constraints in (10.27) when one or several snake robot links are in contact with obstacles.

  - The continuous constrained dynamics of the snake robot during sustained contact with a fixed set of obstacles is described by (10.36) when the obstacles are frictionless, and by (10.42) when obstacle friction is present.

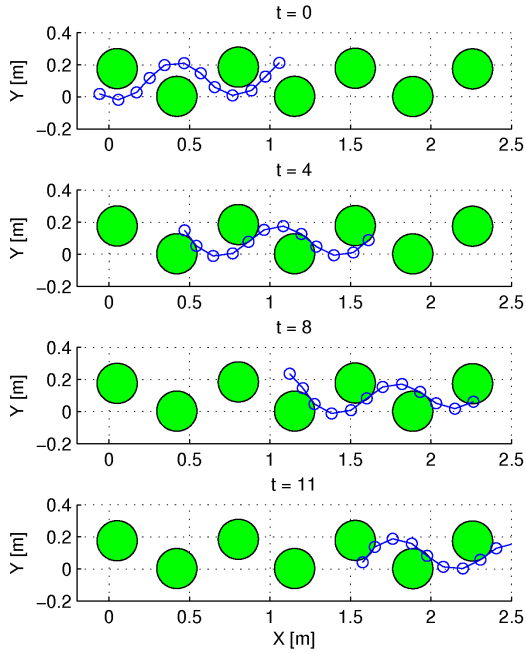


(a) The experimental (top) and simulated (bottom) robot motion.

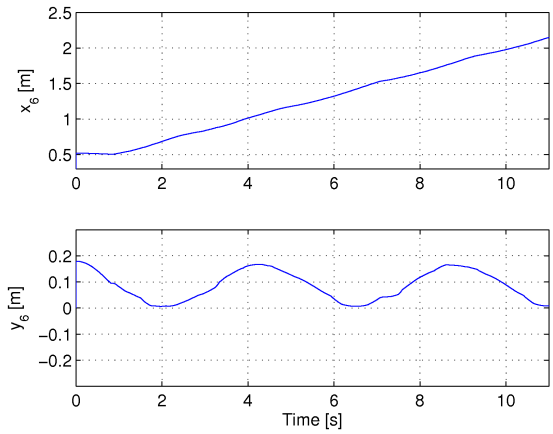


(b) The experimentally measured (solid line) and simulated (dashed line) position of link 6.

Figure 10.6: A reprint of Figures 14 and 15 from Transeth *et al.* (2008*a*), which show experimentally measured and simulated lateral undulation in an environment with obstacles.



(a) The simulated motion of the robot.



(b) The simulated position of link 6.

Figure 10.7: Simulation of lateral undulation in an environment with obstacles based on the hybrid model presented in this chapter. The simulated scenario is similar to the scenario shown in Fig. 10.6.

- Calculation of the constrained dynamics *without* obstacle friction requires us to solve the LCP in (10.37). By Theorem 10.2, this LCP always has a unique solution.
- Calculation of the constrained dynamics *with* obstacle friction requires us to solve the LCP in (10.43). By Theorem 10.3, there always exists a non-zero scalar $\mu_o^* > 0$ such that this LCP has a unique solution when the obstacle friction coefficient μ_o satisfies $0 \leq \mu_o < \mu_o^*$. We were unable to derive an analytical expression for μ_o^*.
- The discontinuous *impact* dynamics of the snake robot occurring when a link (that was previously not in contact with an obstacle) comes into contact with an obstacle is described by (10.49), which requires us to solve the LCP in (10.48). By Theorem 10.4, this LCP always has a unique solution.
- The discontinuous *detachment* dynamics of the snake robot occurring when a link detaches from an obstacle is described by (10.50).
- The complete *hybrid model* of a planar snake robot in an environment with obstacles is given by (10.61).

Chapter 11

Development of a Mechanical Snake Robot for Obstacle-Aided Locomotion

In this chapter, we describe the development of the snake robot *Kulko*, which is shown in Fig. 11.1. While the snake robot *Wheeko*, described in Chapter 3, was used as the experimental platform in Part I of this thesis, Kulko serves as the experimental platform in Part II. The joint modules of Kulko are covered by contact force sensors to allow the robot to sense its environment, and spherical shells that give the robot a smooth outer surface, thereby allowing slithering (gliding) motion in unstructured environments. In the following, we will detail the design and implementation of the robot, and present experimental results that validate the function of the contact force measurement system.

Contributions of this Chapter: Previous snake robot design efforts have given very limited attention to the exterior gliding surface of such robots, and to methods for enabling snake robots to sense their environment (see literature review in Section 1.3). The contribution of this chapter is therefore the design of a spherical-shaped joint mechanism for a snake robot that 1) allows the joint modules to be covered by shells, thereby giving the robot a smooth outer surface independently of how the joints are flexed, and that 2) allows contact force sensors to be installed underneath the shells, thereby enabling the robot to sense its environment. To our best knowledge, this is the first reported snake robot that can measure the magnitude of external forces applied

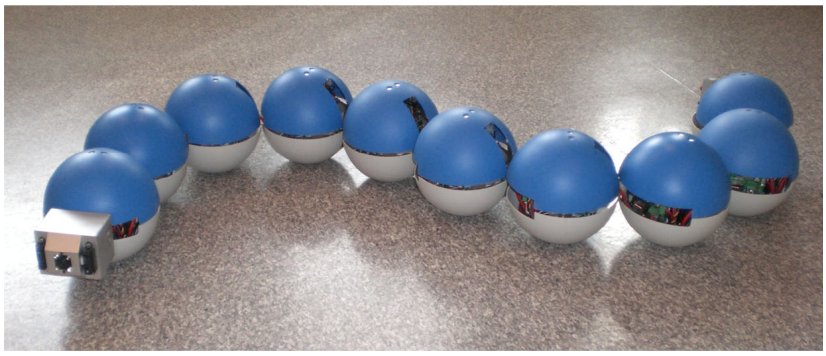


Figure 11.1: The snake robot *Kulko* developed for locomotion in unstructured environments.

along its body. This chapter also proposes an alternative strategy for contact force sensing, which has the advantage that the sensor system can be well protected inside the snake robot.

Organization of this Chapter: We provide an overview of the snake robot design in Section 11.1, followed by a description of the exterior gliding surface, the force measurement system, and the power and control system in Sections 11.2, 11.3, and 11.4, respectively. Section 11.5 presents an experimental investigation of the contact force measurement system, while Section 11.6 describes the setup of the motion control experiments that have been carried out with the robot. An alternative approach to contact force sensing is outlined in Section 11.7. Finally, the chapter is summarized in Section 11.8.

Publications: The material in this chapter is based on the journal papers Liljebäck *et al.* (2010*e*), Liljebäck *et al.* (2010*k*), and Liljebäck *et al.* (2011*e*), and on the conference papers Liljebäck *et al.* (2009*d*) and Liljebäck *et al.* (2010*c*).

11.1 Overview of the Snake Robot Design

As we stated in Hypothesis 9.1, snake robot locomotion in an unstructured environment requires that the snake robot can *sense* its environment, which can be achieved by equipping the robot with contact force sensing capabilities along its body. A force sensing system for a snake robot is challenging

since the robot is articulated. In particular, the force sensing capabilities of the robot should be maintained independently of how the joints are flexed, which represents a significant design challenge.

Enabling a snake robot to glide forward in an unstructured environment requires that the body of the robot is sufficiently smooth, i.e. free of obstructive features. In particular, irregularities along the body may potentially induce large friction forces on the robot that obstruct the gliding motion. Obtaining a sufficiently smooth surface combined with contact force sensing along the articulated body is challenging.

The idea behind the design of Kulko, which was conceived with the above challenges in mind, is to encapsulate each joint module by a spherical shell that gives the joint a smooth outer surface independently of how the joint is flexed. Contact force sensing is thereby achieved by mounting force sensors underneath each spherical shell. As shown in Fig. 11.1, the complete snake robot consists of a serial connection of 10 identical ball-shaped joint modules. The smooth exterior surface and the force sensing capabilities of the robot are maintained independently of how the joints are flexed.

Since the scope of this thesis is *planar* snake robot locomotion (as explained in Section 1.4), Kulko was primarily developed to study obstacle-aided locomotion on a *horizontal* surface with vertical obstacles, which corresponds to the environment captured by the hybrid model proposed in Chapter 10. As a result, the contact force sensor system of Kulko was implemented to primarily measure horizontal contact forces, even though the design can be modified to measure contact forces of arbitrary direction.

In order to study obstacle-aided locomotion, we required Kulko to be propelled solely by contact forces from external obstacles. To this end, we deliberately designed the snake robot to have *isotropic* ground friction properties by making the shells completely smooth. Had we designed the robot with *anisotropic* friction properties (e.g. by making shells with grooves to produce larger friction forces in the normal direction of the body), the efficiency of the motion would have been improved according to the analysis in Section 4.3. However, we then would have been unable to know if the robot was propelled by obstacle contact forces or by the propulsive ground friction forces produced due to the anisotropic friction property.

In the following sections, we describe the various components of Kulko in more detail. The joint actuation mechanism of Kulko is identical to the actuation mechanism of Wheeko, and is therefore described in Chapter 3.

Figure 11.2: Left: The upper and lower hemispherical shell of a joint module. Right: The smooth gliding surface along the snake robot.

11.2 The Exterior Gliding Surface

The smooth exterior gliding surface of Kulko is obtained by covering each joint module by two hemispherical shells, as shown in Fig. 11.2. Each hemispherical shell is 1.5 mm thick, weighs 42 g, and has an outer diameter of 140 mm. The shells were moulded from a plastic material.

As illustrated to the left in Fig. 11.3, four aluminium plates (indicated with a red colour) are bent around the joint in order to support the shells and also to allow for contact force measurements. Each shell is attached to the joint mechanism by two screws, as shown to the right in Fig. 11.3. The locations of the attachment screws define the *top* and *bottom* of the snake robot, respectively. The splice between the two hemispherical shells lies in the horizontal plane. The shells have a slit on each side corresponding to the range of motion of the connection points to the two neighbouring joints.

Note that an even more smooth exterior surface can be obtained by installing a thin hollow cylinder of e.g. a plastic material between each joint module. This approach is illustrated in Fig. 11.4, but has not been implemented on Kulko since it was not necessary for the experiments reported in this thesis.

11.3 The Contact Force Measurement System

11.3.1 Assumptions Underlying the Sensor System

As explained in Section 11.1, the main goal of Kulko is to demonstrate obstacle-aided locomotion on *horizontal* surfaces with vertical obstacles. The design described in the following therefore assumes that all contact

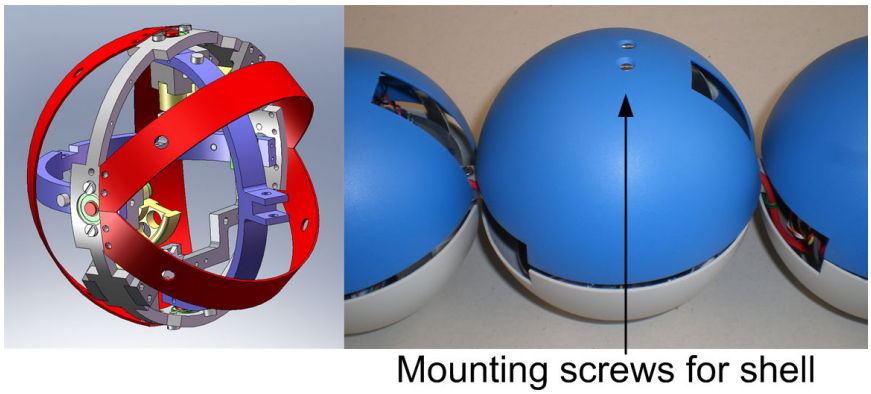


Figure 11.3: Left: The four curved aluminium plates (with red colour) used for mounting force sensors. Right: The pair of screws for attaching the shell to the joint mechanism.

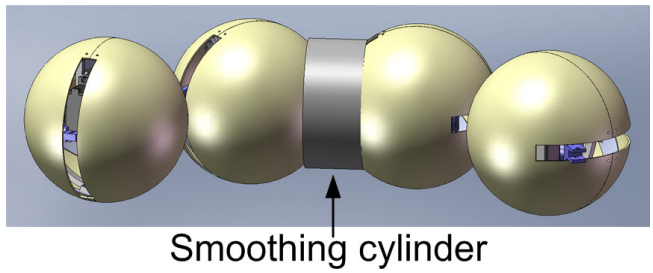


Figure 11.4: Installing a thin cylinder between each joint module will further smoothen the exterior surface of the snake robot.

forces are applied at the sides of the joint and not at the top or bottom. This assumption affects the placement of the contact force sensors. Note that the design can be modified to measure contact forces of arbitrary direction. However, these modifications are not considered in this thesis.

We require the contact force measurement system to provide information about contact forces with respect to the *macroscopic* shape of the snake robot. Information about the specific location of an applied contact force within a single joint module is not believed to be of significant interest during obstacle-aided locomotion since the location of a force within a single joint module only has a minor effect on the motion compared to the location of the force with respect to the overall shape of the robot. This means that the sensor system is only required to determine the *magnitude* of a contact force and also at which *side* of a joint module it is applied, but not the specific location where the force is applied on the outer shell. It should be noted that information about the force location within a joint module could be extracted from the force measurements by relating the magnitude of the measured forces to the relative placement of each sensor.

Since the location of the contact force with respect to the shell is not determined by the sensor system, it will not be possible to determine the exact direction of the contact forces. However, we conjecture that it will be adequate to approximate the direction of any contact force as being *normal to the macroscopic shape* of the snake robot at the location where the force is applied. This approach is in line with the contact modelling approach of the hybrid model presented in Chapter 10.

11.3.2 The Sensor System Setup

A set of force sensing resistors (FSRs) are used to measure the external contact forces applied to each joint module. A FSR is a polymer thick film device that exhibits a decrease in electrical resistance when the force applied to the active surface area of the sensor increases. Due to effects such as hysteresis, a FSR is not suited for precision measurements. However, we conjecture that obstacle-aided locomotion with a snake robot does not require very precise force measurements, which, combined with their low cost and ease of use, make FSRs suitable as a force sensor for snake robots.

The FSR chosen for Kulko has a diameter (active sensor area) of 13 mm and is shown to the left in Fig. 11.5. The right of Fig. 11.5 shows the placement of the FSRs on the curved aluminium plates covering each joint. A small cotton pad (3 mm thick) is placed over each FSR in order to distribute the applied force across the entire active area of the sensor.

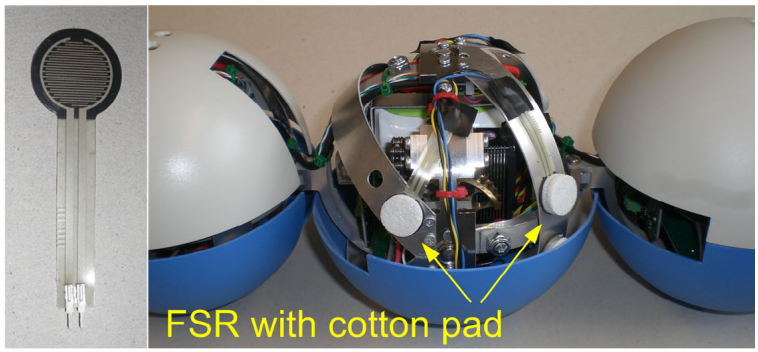


Figure 11.5: Left: FSR (force sensing resistor) used to measure contact forces. Right: FSRs covered by cotton pads mounted to a joint module.

Four FSRs are placed at each side of a joint module in order to be able to measure horizontal contact forces, as explained in Section 11.3.1. There are, in other words, eight FSRs mounted to each joint module. The exact placement of the FSRs around the joint is not critical since, as explained in the next subsection, the magnitude of the contact force is estimated by simply summing the contact forces measured by each FSR.

Note that the hemispherical shells enclosing the sensors are not completely rigid, i.e. the shells are, to some extent, deformable. This, combined with the deformability of the cotton pads placed over each FSR, means that there is compliance between the sensors and the location of an applied force.

The controller board for the joint, which is described in Section 11.4, contains a set of identical voltage divider circuits for measuring the resistance through the FSRs. The circuit diagram for the voltage divider circuit is shown in Fig. 11.6. The voltage V_{ADC}, where ADC denotes *analog to digital converter*, is the FSR measurement signal and is given as a function of the variable resistance R_{FSR} across the FSR.

11.3.3 Calculation of Contact Forces

The force vs. resistance characteristic of a FSR is extremely nonlinear. However, as shown in Fig. 11.7, there is a near linear relationship between the conductance (1/resistance) of a FSR and the force applied to it. The measurements in the figure are indicated by '*' and were carried out by placing an FSR on a digital scale. We used the scale to measure the force applied to the FSR while simultaneously measuring the electrical resistance

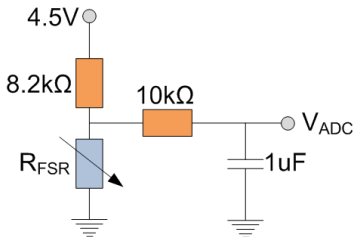


Figure 11.6: The voltage divider circuit used to measure the resistance through the FSR.

through the FSR. A linear curve approximation to these measurements is plotted with a solid line in Fig. 11.7. The linear curve approximates the relationship between the force F_{FSR} applied to a FSR as a function of its conductance G_{FSR} and resistance R_{FSR}. Based on the measurements, the expression for this linear curve was estimated as

$$F_{\mathrm{FSR}} = 18.9 \cdot G_{\mathrm{FSR}} = \frac{18.9}{R_{\mathrm{FSR}}}. \tag{11.1}$$

A simple mapping may now be derived between the FSR measurement voltage in Fig. 11.6, V_{ADC}, and the estimated applied force, F_{FSR}. The measurement voltage is given by

$$V_{\mathrm{ADC}} = \frac{R_{\mathrm{FSR}}}{R_{\mathrm{FSR}} + 8.2} 4.5. \tag{11.2}$$

Solving (11.1) for R_{FSR}, inserting into (11.2), and solving for F_{FSR} give

$$F_{\mathrm{FSR}} = \frac{4.5 - V_{\mathrm{ADC}}}{8.2 V_{\mathrm{ADC}}} 18.9. \tag{11.3}$$

As explained in Section 11.3.2, each side of the joint mechanism is equipped with four FSRs. Since the spherical shell covering the joint mechanism is only in contact with the internal structure of the joint through the FSR measuring points, the magnitude of an external contact force applied to the joint may be estimated by simply summing the forces measured at each FSR. Note that the attachment of the shells causes the shells to induce a constant pressure on the force sensors even when there are no external forces acting on the shells. This produces a constant force offset which we subtract from the force measurements, and which we calculate as the

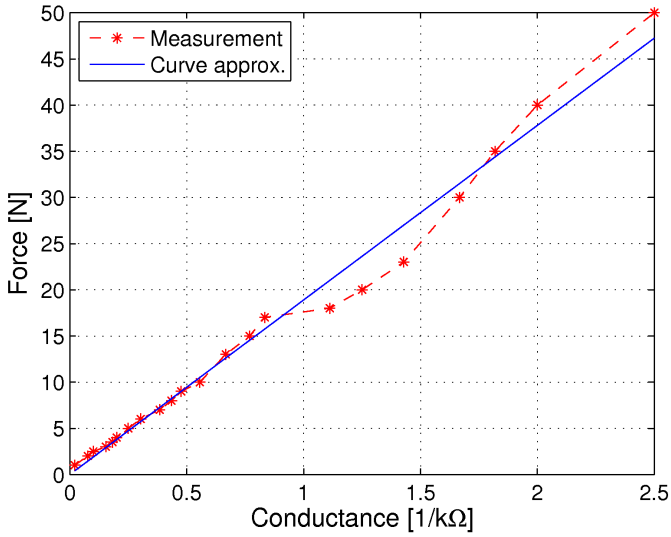


Figure 11.7: The measured conductance (1/R) of the FSR as a function of applied force (measurements indicated by '*'). The solid line shows the linear curve approximation to these measurements.

average force during the first second after the snake robot is powered up. We denote the four FSR measurements on the left side of the joint by $F_{\text{FSR,left},1}, .., F_{\text{FSR,left},4}$, the measurements on the right side of the joint by $F_{\text{FSR,right},1}, .., F_{\text{FSR,right},4}$, and the force offset on the left and right side of the joint by $F_{\text{left,offset}}$ and $F_{\text{right,offset}}$, respectively. Consequently, we can estimate the total external forces, F_{left} and F_{right}, applied to the left and right side of the joint, respectively, as

$$F_{\text{left}} = \left(\sum_{i=1}^{4} F_{\text{FSR,left},i}\right) - F_{\text{left,offset}}, \tag{11.4a}$$

$$F_{\text{right}} = \left(\sum_{i=1}^{4} F_{\text{FSR,right},i}\right) - F_{\text{right,offset}}. \tag{11.4b}$$

11.4 The Power and Control System

Motion control and supply of power to the components of each joint module are handled by three custom-designed circuit boards installed in each joint

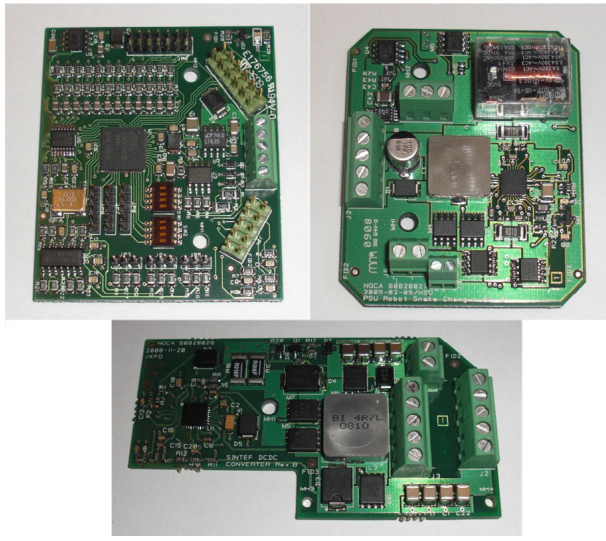


Figure 11.8: The three custom-designed circuit boards located in each joint module. Top left: Microcontroller card that controls the joint mechanism. Top right: Battery charger card. Bottom: Motor power supply card.

module. These circuit boards are shown in Fig. 11.8 and are described in the following subsections.

11.4.1 The Power System

Fig. 11.9 illustrates the flow of power to the various components of a joint module. Each joint is powered by two serially connected Lithium Ion batteries from *A123Systems* of the type ANR26650M1. The batteries produce a supply voltage of about 6.6V at a capacity of 2.3Ah. The batteries were chosen due to their ability to deliver high currents (rated at 70A continuous discharge current) and also their short charge time (rated at 15 min charge time at 10A charge current). In particular, the two Hitec servo motors driving each joint (see Section 3.1) draw high current pulses each time the motor direction changes rapidly. Had the power system not been able to deliver such high currents, then the system voltage would drop and reset the microcontroller card each time the motor direction changes rapidly.

Each battery is charged by an individual battery charger card shown at the top right in Fig. 11.8. The charging is automatically initiated by applying an external voltage to the external power connectors located at

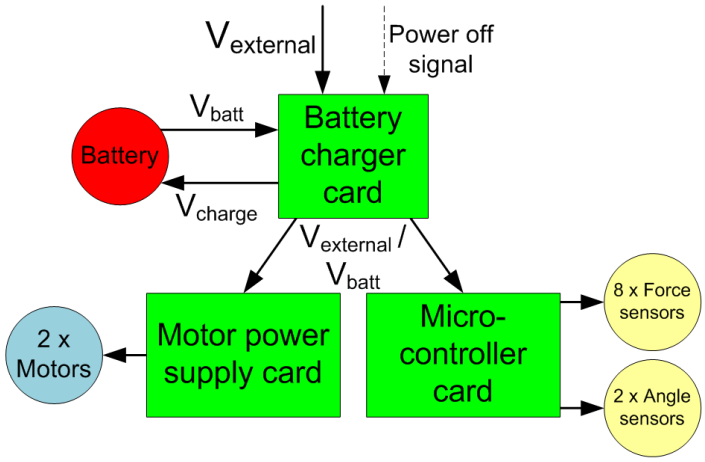


Figure 11.9: The flow of power to the components of a joint module.

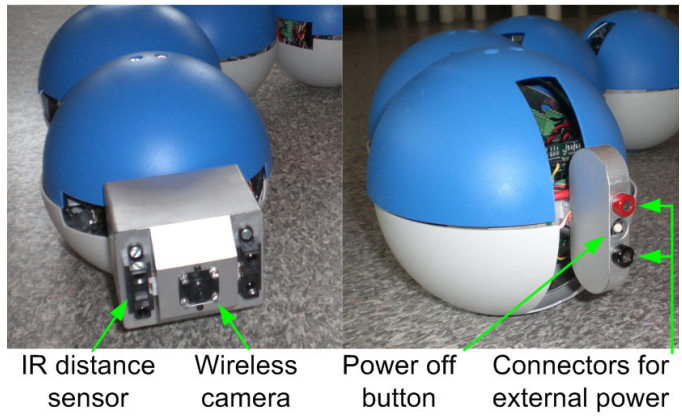


Figure 11.10: Left: The head of the snake robot. Right: The tail of the snake robot.

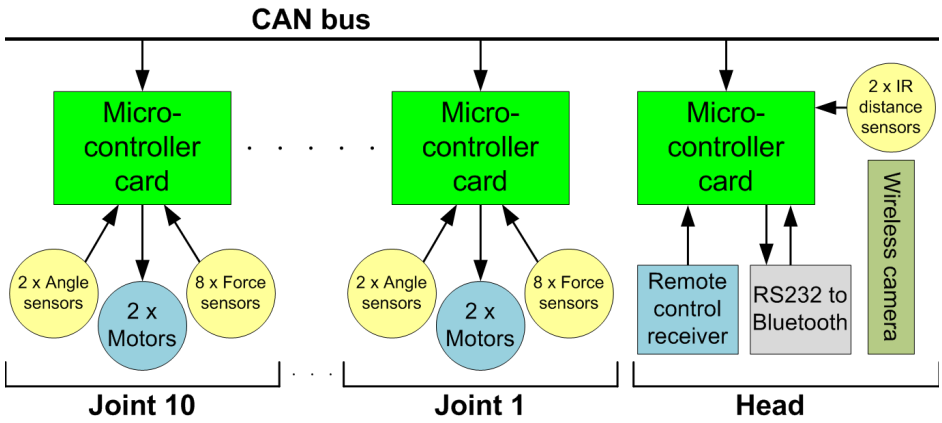


Figure 11.11: The data flow between the modules of the snake robot.

the tail of the snake robot (see the right of Fig. 11.10). Each battery is connected in series with a relay, which is controlled by a power off button also located at the tail of the snake robot.

The motor power supply card, shown at the bottom of Fig. 11.8, supplies power to the two servo motors driving each joint. This card converts the voltage supplied by the battery charger card to the motor voltage (6 V). The voltage supplied by the battery charger card is either the battery voltage (when external power is disconnected) or the external voltage.

11.4.2 The Control System

The data flow between the components of the snake robot is illustrated in Fig. 11.11. Motion control of each joint module is handled by the microcontroller card shown to the top left in Fig. 11.8, which is based on the *Atmel* microcontroller AT90CAN128. This card continuously reads angular measurements from the two magnetic encoders (see Fig. 3.3 in Section 3.1) and also contact force sensor data from the FSRs (see Fig. 11.5). This card also generates PWM pulses that control the two servo motors driving each joint module. The card has a CAN bus interface for communicating with the other modules of the snake robot.

The brain (or head) of the snake robot, which is shown to the left in Fig. 11.10, contains the same microcontroller card that controls the motion of the joints. The brain card is responsible for sending joint reference angles to all joint modules over the CAN bus. The joint reference angles

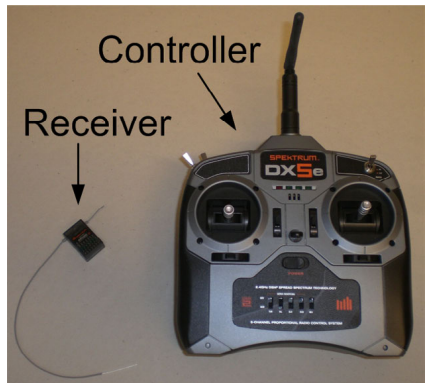


Figure 11.12: The remote controller and the receiver used for demonstrating the snake robot.

are calculated on an external computer in accordance with any desired control strategy and sent to the brain card via a wireless connection based on Bluetooth. The refresh rate for the two reference angles of each joint module is about 20 Hz.

For simple demonstration purposes (not for experimental purposes), the snake robot can also be manually controlled with a commercially available radio transmitter (the DX5e developed by *Spektrum*), which is shown in Fig. 11.12. The receiver of the radio controller is connected to the brain card, which calculates joint reference angles based on the input from the radio controller. The mapping from radio control input to the resulting joint reference angles will not be detailed here as it is not relevant to the experiments reported in this thesis.

As shown to the left in Fig. 11.10, the head of the snake robot is equipped with a small wireless camera and two IR distance sensors (*Sharp* GP2D120). These sensors have not been employed in the experiments reported in this thesis, but are intended for future experiments where the goal is to prevent the head from colliding with obstacles in its path.

11.5 The Performance of the Snake Robot

In this section, we present experimental results that validate the function of the contact force measurement system. We also show the ability of the snake robot to display different motion patterns.

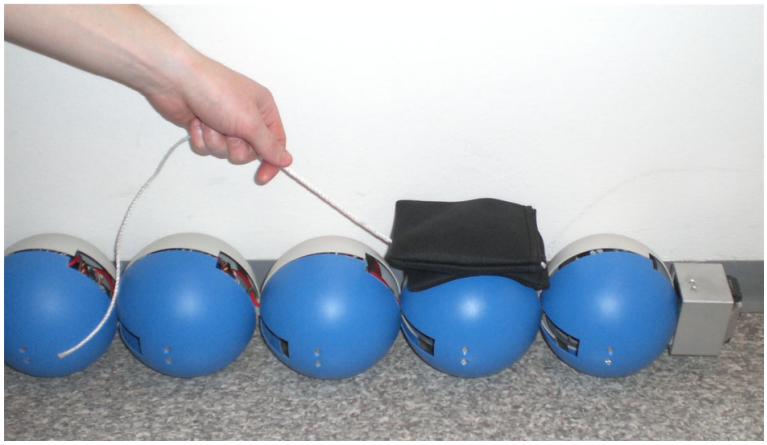


Figure 11.13: The experimental investigation of the contact force measurement system of Kulko.

11.5.1 Experimental Validation of the Contact Force Measurement System

The force measurement system of Kulko was experimentally investigated as shown in Fig. 11.13. In particular, we placed the robot against a wall (to prevent it from rolling over) so that the left side of the robot was facing upwards. Loads with different weights were then dragged backwards along the part of the snake robot facing upwards (i.e. the left side), while each joint module estimated the applied contact forces according to (11.4) at a sampling frequency of 10 Hz. The joint modules reported the measured forces over the CAN bus to the brain module, which redirected these measurements to an external computer over the wireless Bluetooth connection.

Three different loads weighing 1350 g, 2750 g, and 4300 g, respectively, were dragged from the head and backwards along the snake robot. The resulting force measurements at joint 4, joint 5, and joint 6 (joint 1 is the foremost module) are shown in Fig. 11.14(a)-(c), respectively. In theory, the amplitude of each force curve should be 13.5 N, 27.5 N, and 43 N, respectively, for the three different loads. Despite some deviations, the measured forces agree well with the weight of the loads. As described in Section 11.3.2, a FSR is not suitable for precision measurements, and consequently, some deviations were expected. However, we conjecture that obstacle-aided locomotion with a snake robot primarily requires the ability to detect a contact force and also, to some extent, assess the magnitude

of this force. The experimental results indicate that the proposed sensor setup is able to meet these requirements.

Note that an ideal sensor system would produce a linear horizontal curve corresponding to the weight of the load being dragged along the snake body. Since the plots of the measured forces are instead given as peaks, it is clear that the sensor system does not measure forces *between* the joints very well. However, we do not consider this to be a critical issue in order to demonstrate obstacle-aided locomotion, especially not if the obstacles are large compared to the size of each joint module.

11.5.2 Demonstration of Motion Patterns

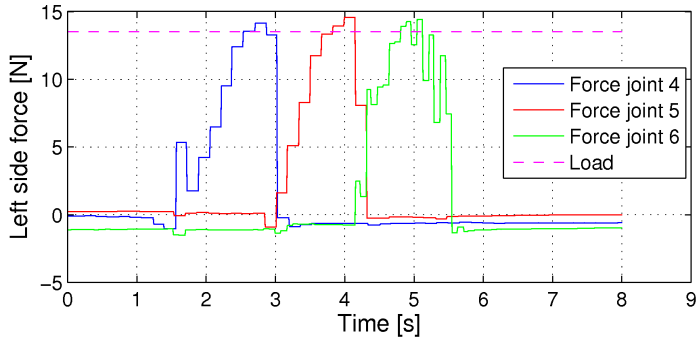
Some of the motion capabilities of Kulko are demonstrated in Fig. 11.15. In Fig. 11.15(a), the snake robot conducts sidewinding across a flat surface. This is a sideways motion produced by propagating both horizontal and vertical body waves backwards along the snake (see e.g. Transeth *et al.*, 2008*b*). In Fig. 11.15(b), the snake robot conducts lateral rolling, which is a rolling motion produced by continuously creating a U-shape with the snake body that tips over to one side (see e.g. Mori and Hirose, 2002).

In summary, we conjecture that the motion capabilities of the snake robot are satisfactory and adequate in order to demonstrate obstacle-aided locomotion.

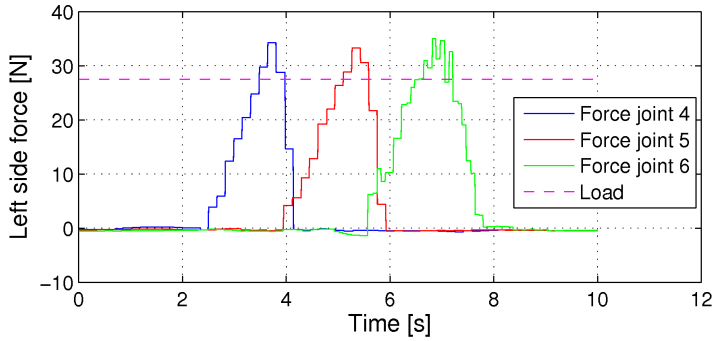
11.6 The Experimental Setup of the Snake Robot

The motion control experiments carried out with Kulko are reported in Chapter 12 and Chapter 13. These experiments were carried out on a black horizontal surface measuring about 100 cm in width and 200 cm in length. As shown in Fig. 11.16, circular obstacles were placed around the robot. The location of each obstacle could easily be changed by means of a grid of mounting holes in the floor. The friction coefficient between the floor and the robot, denoted by μ, and the friction coefficient between the obstacles and the robot, denoted by μ_o, were found experimentally to lie between 0.2 and 0.3. Note that the physical environment of the snake robot corresponds to the environment captured by the hybrid model proposed in Chapter 10.

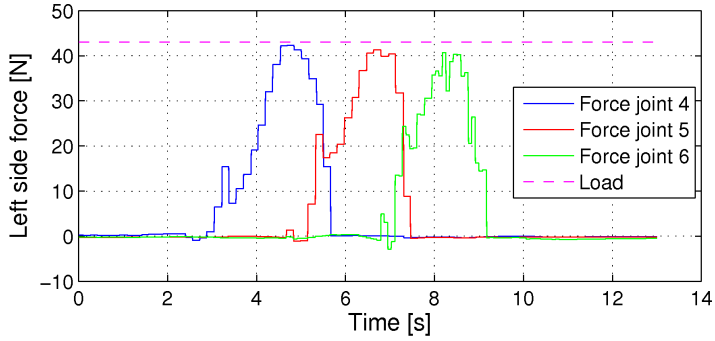
The horizontal position of the snake robot during the experiments was measured by mounting black circular markers to the snake robot, as shown to the right in Fig. 11.16, and then tracking these markers by use of the camera system shown to the left in Fig. 11.16. This camera system is



(a) Forces measured with a load weighing 1350 g.

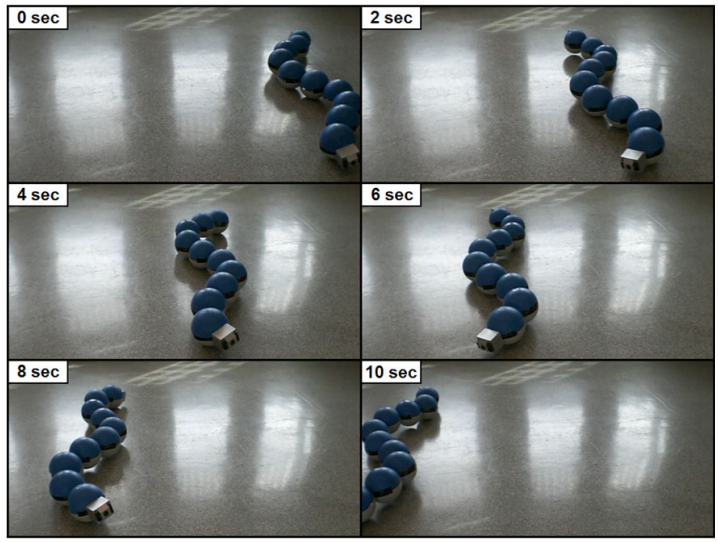


(b) Forces measured with a load weighing 2750 g.

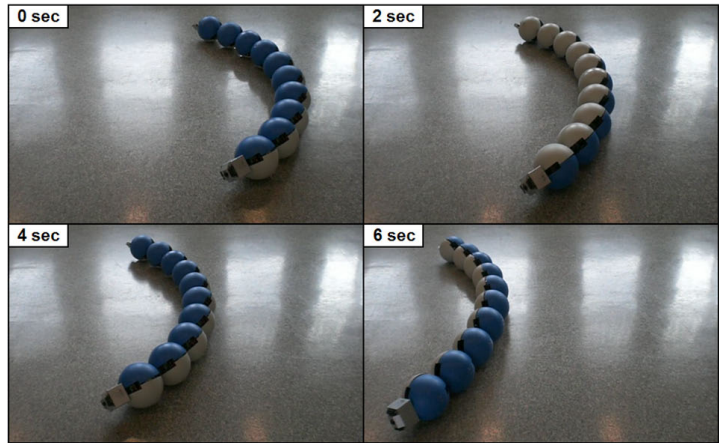


(c) Forces measured with a load weighing 4300 g.

Figure 11.14: Forces measured by joint 4 - 6 when three different loads were dragged along the snake robot.



(a) Kulko sidewinding across the floor.



(b) Kulko conducting lateral rolling across the floor.

Figure 11.15: Demonstration of the motion capabilities of Kulko.

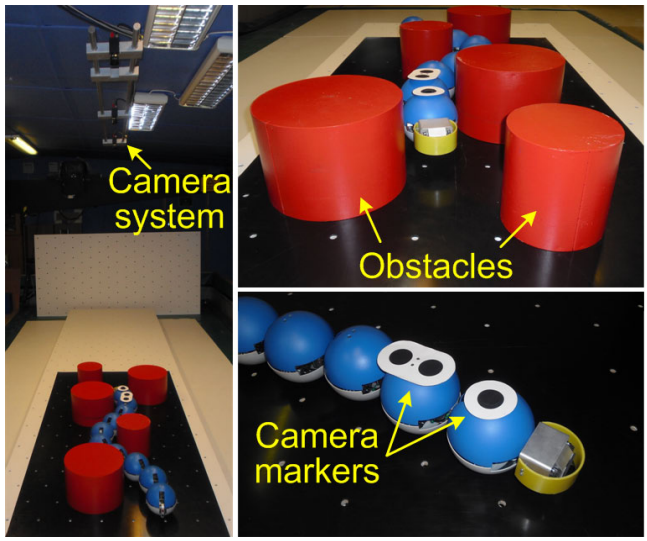


Figure 11.16: The experimental setup. Three cameras mounted in the ceiling measured the position of the snake robot in a course with obstacles.

identical to the camera system used in the experimental setup of Wheeko, and is described in more detail in Section 3.4.

11.7 An Alternative Approach for Measuring External Contact Forces

We end this chapter by proposing an alternative strategy for environment sensing based on force measurements. With respect to Kulko, this alternative approach leads to a simpler instrumentation system. The idea behind the approach was, however, conceived *after* the development of Kulko. An experimental validation of the approach is therefore a topic of future work.

The idea is simply to calculate the external forces on each link of the robot based on measurements of the joint constraint forces that occur at the connection between the links. As illustrated in Fig. 11.17, a major advantage of this approach is that external forces on each link can be determined based on force measurements conducted *inside* the robot. In particular, the top of Fig. 11.17 illustrates force sensing based on direct measurement of the external forces acting on each link, which basically requires that the sensing area covers the link completely. Kulko is based on this force sensing

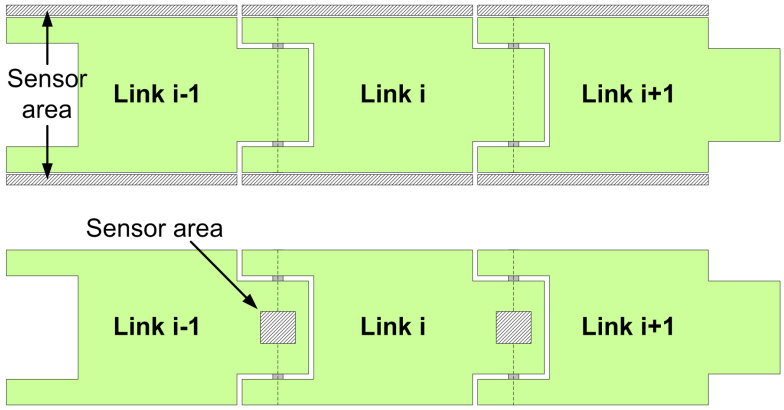


Figure 11.17: Top: Sensor area required for direct measurement of external forces. Bottom: Sensor area required for calculating external forces based on internally measured joint constraint forces.

principle. Force sensing based on measured joint constraint forces, on the other hand, only requires instrumentation at the locations of the joints, and all the instrumentation can be well protected inside the joint modules. This alternative approach, which is illustrated in the bottom of Fig. 11.17, simplifies the development of the physical coverage of a snake robot since the coverage can be developed more or less independently of the contact force sensor system. Techniques for measuring the joint constraint forces, of which there are many, remain a topic of future work.

To verify that the external forces on the snake robot indeed can be determined from the joint constraint forces, we investigate the force balance for the links. The force balance of a link which is not in contact with an obstacle is given by (2.25). During contact with an obstacle, the link is additionally influenced by an obstacle constraint force and an obstacle friction force, as described in Section 10.3. The force balance of link $i \in \{1, \ldots, N\}$ can therefore be written as

$$m\ddot{\boldsymbol{a}}_i = \boldsymbol{f}_{R,i} + \boldsymbol{f}_{c,i} + \boldsymbol{f}_{\mu,i} + \boldsymbol{h}_i - \boldsymbol{h}_{i-1}, \tag{11.5}$$

where m and $\ddot{\boldsymbol{a}}_i = [\ddot{x}_i, \ddot{y}_i]^T \in \mathbb{R}^2$ are the mass and translational acceleration, respectively, of link i, $\boldsymbol{f}_{R,i} \in \mathbb{R}^2$ is the ground friction force, $\boldsymbol{f}_{c,i} \in \mathbb{R}^2$ is the constraint force from any external obstacle, $\boldsymbol{f}_{\mu,i} \in \mathbb{R}^2$ is the obstacle friction force, $\boldsymbol{h}_i \in \mathbb{R}^2$ is the joint constraint force from link $i+1$, and $-\boldsymbol{h}_{i-1} \in \mathbb{R}^2$ is the joint constraint force from link $i-1$. With the alternative force sensing

approach, the joint constraint forces $\boldsymbol{h}_i$ and $\boldsymbol{h}_{i-1}$ are measured. The sum of the external forces on link i, which we denote by $\boldsymbol{f}_{\text{ext},i} \in \mathbb{R}^2$, can thereby be calculated as

$$\boldsymbol{f}_{\text{ext},i} = \boldsymbol{f}_{R,i} + \boldsymbol{f}_{c,i} + \boldsymbol{f}_{\mu,i} = m\ddot{\boldsymbol{a}}_i - \boldsymbol{h}_i + \boldsymbol{h}_{i-1}, \tag{11.6}$$

which is given solely from the measured joint constraint forces when the velocity of the link is zero or constant so that $\ddot{\boldsymbol{a}}_i = \boldsymbol{0}$. Moreover, since snake locomotion is usually a smooth gliding form of locomotion with slowly varying link velocities, we conjecture that (11.6), with $\ddot{\boldsymbol{a}}_i$ set to zero, also in general will provide a good approximation of $\boldsymbol{f}_{\text{ext},i}$. Alternatively, the estimate of $\boldsymbol{f}_{\text{ext},i}$ can be improved by also measuring $\ddot{\boldsymbol{a}}_i$, which is easily achieved by installing a small acceleration sensor inside each link.

11.8 Chapter Summary

This chapter is summarized as follows:

- We have presented the design of the snake robot *Kulko*, which was developed for the purpose of experiments related to obstacle-aided locomotion in unstructured environments.
- The robot consists of 10 identical joint modules covered by contact force sensors (to allow the robot to sense its environment) and spherical shells that give the robot a smooth outer surface (to allow slithering motion in unstructured environments).
- The internal structure of Kulko is identical to the internal structure of the snake robot *Wheeko*, which was described in Chapter 3.
- We have presented experimental results that validate the function of the contact force measurement system, and also demonstrated some of the motion capabilities of the robot.
- We have proposed an alternative strategy for contact force sensing which is based on measuring the joint constraint forces at the connection between the links. The advantage of this approach is that the sensor system can be well protected inside the snake robot.

Chapter 12

Hybrid Control of Obstacle-aided Locomotion

In direct accordance with Hypothesis 9.1, we propose in this chapter a control strategy that enables a snake robot to propel its body forward by active use of the interaction with obstacles in its environment. This form of propulsion is called obstacle-aided locomotion and was introduced in Chapter 9. Obstacle-aided locomotion represents an interesting control problem for which previous research is very limited. The literature review presented in Section 1.3 clearly shows that a large majority of control strategies proposed for snake robots so far assume that the environment of the robot is flat. In fact, the works in Bayraktaroglu (2008); Bayraktaroglu and Blazevic (2005); Hirose (1993) present, to our best knowledge, the only known control strategies related to obstacle-aided locomotion. We believe control strategies for snake robots that consider environment interaction are important since the main advantage of these mechanisms are their potential ability to move in unstructured environments.

The difference in complexity between flat surface locomotion, which was considered in Part I of this thesis, and obstacle-aided locomotion is significant. Unlike flat surface locomotion, where we know that periodic body waves will propel a snake robot forward under anisotropic ground friction conditions, there exists no clear intuition as to how we can control a snake robot so that it is propelled forward by obstacle contact forces. One obvious and major challenge is that we do not know in advance how, when, and where the snake robot will make contact with its environment. A second major challenge is to develop a general strategy for adjusting the shape of the robot so that forward propulsion is achieved in any given

contact situation.

Our proposed solution to this problem is simple and, in many ways, obvious. Since we know that the obstacle contact forces are what propel the robot forward, and since the obstacle contact forces act in the normal direction of each link (see Section 10.3), forward propulsion must obviously be achieved due to the normal direction contact forces on the links. From this observation, we state the following hypothesis:

Hypothesis 12.1 *Obstacle-aided snake robot locomotion is achieved by producing body shape changes where the links in contact with obstacles are rotated so that the components of the contact forces in the desired direction of motion are increased.*

In order to investigate this fundamental control principle, we will in this chapter introduce the concepts of *jam detection* and *jam resolution*. A snake robot which moves in an unstructured environment without taking the environment interaction into account, is likely to become *jammed* between the obstacles in its path. We will show that a control strategy based on the control principle in Hypothesis 12.1 is efficient for resolving such jams and maintaining the propulsion of the snake robot. Since a jam of the robot can be regarded as a discrete event, we will employ a hybrid formulation in the control strategy similar to the formulation of the hybrid model in Chapter 10, i.e. we will propose a *hybrid controller*. The performance of the controller will be illustrated with simulation results and with experimental results based on the snake robot Kulko. In Chapter 13, the control principle in Hypothesis 12.1 will be employed to propose a more general control law that combines environment adaptation with path following capabilities in an unstructured environment.

Contributions of this Chapter: The contribution of this chapter is a *hybrid controller* for obstacle-aided locomotion aimed at resolving situations where the snake robot is jammed between obstacles. Included in this contribution is the control principle proposed in Hypothesis 12.1. We strongly believe that this control principle is applicable also in the general case of three-dimensional snake robot locomotion. The concept of detecting and resolving snake robot jams has, to our best knowledge, not been treated in previous literature, but is a genuine challenge during snake robot locomotion in cluttered environments. To our knowledge, this is the first published control strategy for a snake robot involving feedback and explicit use of measured contact forces to achieve propulsion. Note that the work in Hirose (1993)

also considers snake locomotion based on measured contact forces. However, the contact forces in Hirose (1993) are employed to *avoid* obstacles, whereas the contact forces in this chapter are employed to push the snake robot forward. The experimental investigation of the controller by use of the snake robot Kulko is also considered a novel contribution of this chapter. To our best knowledge, this is the first reported experiment where a snake robot is propelled forward based on measurements of the amplitude of contact forces along the body of the robot. The works in Bayraktaroglu (2008); Hirose (1993) also report experiments where a snake robot is propelled by obstacle contact forces. However, the control strategies in these works do not consider the amplitude of the contact forces since discrete contact switches on the robots are used to detect the obstacles.

Organization of this Chapter: A general description of hybrid controllers is presented in Section 12.1. The objective and the basic assumptions underlying the hybrid controller are presented in Sections 12.2 and 12.3, respectively. Section 12.4 presents the hybrid controller, while the closed-loop system (i.e. the hybrid model with the hybrid controller) is summarized in Section 12.5. Sections 12.6 and 12.7 present, respectively, simulation results and experimental results that illustrate the performance of the controller. Finally, the chapter is summarized in Section 12.8.

Publications: The material in this chapter is based on the journal papers Liljebäck *et al.* (2010*h*) and Liljebäck *et al.* (2010*e*), and on the conference paper Liljebäck *et al.* (2009*a*).

12.1 Preliminary Note on Hybrid Controllers

Consider any continuous or hybrid plant (i.e. a model of a dynamical system) with state vector $\boldsymbol{x} \in \mathbb{R}^n$ and control input $\boldsymbol{u} \in \mathbb{R}^m$. If the controller that generates the control input $\boldsymbol{u}$ for the system consists of an algorithm with discrete-valued states, then we denote this a *hybrid controller* (Goebel *et al.*, 2009). A hybrid controller is a hybrid system with state $\boldsymbol{\eta} \in \mathbb{R}^p$ (which can contain e.g. logic states, timers, and counters) that evolves as a function of both the controller state $\boldsymbol{\eta}$ and the plant state $\boldsymbol{x}$. The control input is generally calculated according to a function $\boldsymbol{u} = \boldsymbol{\kappa}(\boldsymbol{x}, \boldsymbol{\eta})$. We can describe a hybrid controller by the hybrid modelling framework introduced in Section 10.1.1. Sometimes hybrid controllers are used to control plants

that are continuous-time systems (see e.g. Goebel *et al.* (2009) for various examples). In this chapter, however, we will propose a hybrid controller for a snake robot described by the hybrid model proposed in Chapter 10, i.e. we will consider a hybrid controller for a hybrid plant.

12.2 Control Objective

A major challenge during obstacle-aided locomotion is to prevent the snake robot from being *jammed* between the obstacles. In a jammed situation, the *propulsive* components (i.e. the force components in the desired direction of motion) of the contact forces from the obstacles are too small to overcome the friction forces on the robot, and hence the forward motion stops. In this jammed situation, the obstacle contact forces will also prevent a number of the snake robot joints from moving to their reference angle. The goal of this chapter is to employ the control principle in Hypothesis 12.1 to develop a strategy for detecting and resolving situations where the motion of the robot is jammed. To this end, we choose the control goal to be locomotion along the global x axis with a positive and nonzero forward velocity. The control problem is thereby to design a feedback control law for the joint torques $\boldsymbol{u} \in \mathbb{R}^{N-1}$ such that the following control objective is reached:

$$\dot{p}_x(t) > 0. \tag{12.1}$$

Note that the controller only targets the propulsion of the snake robot without considering the direction of the motion. Directional control in an obstacle environment is considered in Chapter 13. Another important limitation of the control strategy is the underlying requirement of sufficiently many obstacles in reach of the snake robot. Without obstacles, there are no push-points that the robot can use for propulsion. At the same time, the control strategy will not work if there are too many obstacles so that the path of the robot is blocked. These are not critical issues, however, since the main purpose of the controller is to demonstrate how a snake robot can utilize contact forces from objects in its environment to achieve propulsion.

12.3 Notation and Basic Assumptions

With reference to the obstacle constraint force $\boldsymbol{f}_{c,i}$ defined in Section 10.3, the controller will make use of the following quantities:

Definition 12.1 ***Measured contact force.***
The measured *contact force on link* $i \in \{1, \ldots, N\}$ *is denoted by* $\rho_i \in \mathbb{R}$ *and is defined as the component of the constraint force vector* $\boldsymbol{f}_{c,i}$ *along the* y *axis of the local frame of link* i *(see illustration of the local link frame in Fig. 2.1), i.e. as*

$$\rho_i = [-\sin\theta_i, \cos\theta_i]\, \boldsymbol{f}_{c,i}. \tag{12.2}$$

Moreover, the force measurements of all links are assembled in the vector $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_N]^T \in \mathbb{R}^N$.

Remark 12.1 *Since, as described in Section 10.3, the constraint force vector* $\boldsymbol{f}_{c,i}$ *always acts along the local* y *axis of link* i*, we have that* $|\rho_i| = |\boldsymbol{f}_{c,i}|$.

Definition 12.2 ***Propulsive component.***
The propulsive component *of the contact force on link* $i \in \{1, \ldots, N\}$ *is defined as the component of the constraint force vector* $\boldsymbol{f}_{c,i}$ *along the desired* forward *direction of motion.*

Since the desired forward direction of the controller considered in this chapter is along the global x axis, the propulsive component of a contact force is given as the component of the constraint force vector along the global x axis. As illustrated in Fig. 12.1, this component is denoted by $\rho_{x,i} \in \mathbb{R}$ and is easily calculated as

$$\rho_{x,i} = -\rho_i \sin\theta_i. \tag{12.3}$$

Remark 12.2 *Chapter 13 considers directional control of snake robots in an obstacle environment where the desired forward direction of motion at any time is along the current heading of the robot in order to maintain the forward velocity. The propulsive component of a contact force in Chapter 13 is therefore given as the component along the current heading instead of along the global* x *axis.*

We will base the controller on the following assumptions:

Assumption 12.1 *The initial heading of the snake robot is along the global* x *axis.*

Assumption 12.2 *The control system has access to measurements of the joint angles* $\boldsymbol{\phi}$*, the joint angle velocities* $\dot{\boldsymbol{\phi}}$*, the contact forces* $\boldsymbol{\rho}$*, and at least one of the absolute link angles* θ_i *for some* $i \in \{1, \ldots, N\}$.

Note that the remaining link angles can be calculated from $\boldsymbol{\phi}$ and θ_i.

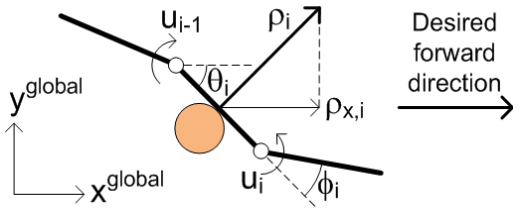


Figure 12.1: The propulsive component $\rho_{x,i}$ of the contact force on link i.

12.4 The Hybrid Controller for Obstacle-aided Locomotion

The control strategy proposed in the following is a *hybrid controller* consisting of a *leader-follower* scheme and a *jam resolution* scheme, and also a supervisory mechanism for switching between these two schemes, denoted the *jam detection* scheme. The *leader-follower* scheme is carried out as long as the robot is able to move without being jammed between the obstacles. If the *jam detection* scheme detects a jam, then the *jam resolution* scheme is carried out in order to effectively 'unlock' the jammed joints.

The *leader-follower* scheme, the *jam detection* scheme, the *jam resolution* scheme, and the joint angle controller are presented in Sections 12.4.1, 12.4.2, 12.4.3, and 12.4.4, respectively, without considering the hybrid nature of the controller. In Section 12.4.5, the complete hybrid controller is summarized and formulated in terms of the framework described in Section 12.1.

12.4.1 The Leader-follower Scheme

In Part I of this thesis, we employed cyclic body wave motion as a means for propelling a snake robot forward. As we are now considering environments with obstacles, we will continue to base the motion of the snake robot on such cyclic body wave motion. The reason for this choice is based on Remark 10.2 in Section 10.3, where we noted that the phenomenon that propels a snake robot forward due to the anisotropic ground friction properties of the links is similar to the phenomenon that propels a snake robot forward due to the interaction of the links with external obstacles. Since cyclic body wave motion produces forward propulsion under anisotropic ground friction conditions, we can thereby expect this to also be the case

in environments with obstacles.

In the leader-follower scheme, we will therefore produce oscillatory body wave motion similar to the lateral undulation gait defined in (4.32). However, we do not specify the body shape motion in terms of (4.32) since the shape of the robot then would be completely predefined, which makes no sense when the environment is not known in advance. Instead, we define the oscillatory body shape motion based on the observation that each part of a biological snake conducting lateral undulation follows the path traced out by the head (see e.g. Gray, 1946), which suggests that we should choose the head joint angle (the angle of the foremost joint), ϕ_{N-1}, as the reference angle for all subsequent joints. This approach is called *leader-following* since all joints follow the motion of the head, i.e. the leader. In contrast to the predetermined lateral undulation motion in (4.32), environment adaptation is, at least to some extent, inherently present in the leader-following motion since all joints follow the *actual* angle of the head joint.

In order to generate a leader-follower based control reference to the joints, the head joint angle ϕ_{N-1} is propagated backwards along the snake body at a constant predefined propagation velocity v_{ref} and used as the reference angle for all subsequent joints according to

$$\phi_{i,\text{ref}}(t) = \phi_{N-1}(t - (N - i - 1)\Delta t), \tag{12.4}$$

where $i \in \{1, \ldots, N-2\}$ and where the time offset Δt between two consecutive joints with intermediate distance $2l$ is found as $\Delta t = 2l/v_{\text{ref}}$.

In order to achieve the sinusoidal motion characteristic of lateral undulation, we alternate between moving the head in the leftward and rightward direction with respect to the global positive x axis. This may be achieved by choosing the reference angle for the head link, $\theta_{N,\text{ref}}$, equal to a suitable positive constant θ_{left} when the head should move leftward and a negative constant θ_{right} when the head should move rightward. The criterion for switching between these two reference directions is defined to be the instant when the amplitude of the head motion perpendicular to the desired forward direction becomes greater than some predefined amplitude. In mathematical terms, we switch the head direction of motion when the distance Δy between the position of the head along the global y axis, y_N, and the y axis coordinate of the CM of the snake robot, p_y, becomes greater than some predefined amplitude, $\Delta y_{\max}$. This criterion is illustrated in Fig. 12.2. The distance Δy is easily calculated as a function of measured state values only. In particular, by inspecting the last row of the matrix

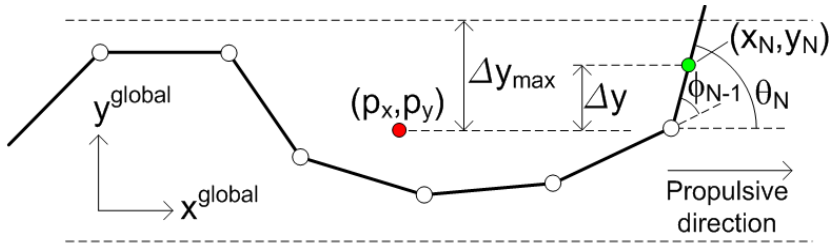


Figure 12.2: The control strategy for the head of the snake robot.

equation in (2.11b), it can easily be verified that Δy is given by

$$\Delta y = y_N - p_y = \sum_{j=1}^{N-1} \frac{l}{N} j \left(\sin\theta_j + \sin\theta_{j+1}\right). \tag{12.5}$$

The reference angle of the head link is in other words set according to the rule

$$\begin{array}{lllll} \text{Leftward motion:} & \theta_{N,\text{ref}} = \theta_{\text{left}} & \text{until} & \Delta y > \Delta y_{\max}, \\ \text{Rightward motion:} & \theta_{N,\text{ref}} = \theta_{\text{right}} & \text{until} & \Delta y < -\Delta y_{\max}, \end{array} \tag{12.6}$$

which means that $\theta_{N,\text{ref}} \in \{\theta_{\text{left}}, \theta_{\text{right}}\}$. Since the head joint (i.e. joint $N-1$) is at the front of the snake robot, the rotation of the head joint mainly affects the angle of the head link (i.e. link N) and not the angle of the subsequent links. From the relation $\phi_{N-1} = \theta_{N-1} - \theta_N$ (see (2.1)), we can therefore track the head link reference angle in (12.6) by controlling the head joint angle ϕ_{N-1} according to the reference

$$\phi_{N-1,\text{ref}} = \theta_{N-1} - \theta_{N,\text{ref}}. \tag{12.7}$$

Note that $\phi_{N-1,\text{ref}}$ will experience a jump each time $\theta_{N,\text{ref}}$ switches. However, the actuator torque applied at joint $N-1$ will still be bounded since the derivative of $\phi_{N-1,\text{ref}}$ with respect to time is not included in the joint controller presented below in Section 12.4.4.

To summarize, the reference angles for all the joints of the snake robot in this leader-follower scheme are

$$\phi_{N-1,\text{ref}}(t) = \theta_{N-1}(t) - \theta_{N,\text{ref}}, \tag{12.8a}$$

$$\phi_{i,\text{ref}}(t) = \phi_{N-1}(t - (N-i-1)\Delta t), \tag{12.8b}$$

where $i \in \{1, \ldots, N-2\}$. The design parameters θ_{left}, θ_{right}, $\Delta y_{\max}$, and v_{ref} were introduced in order to calculate these reference angles.

Remark 12.3 *The implementation of (12.8) requires a buffer which keeps track of the angle history of the head joint,* $\phi_{N-1}(t)$.

12.4.2 The Jam Detection Scheme

We define a single joint of the snake robot to be *jammed* if the deviation between the joint angle and its reference angle exceeds a certain limit, $\Delta\phi_{\max}$. It is reasonable to assume that a jam of a single joint will resolve by itself. However, two jammed joints could be caused by a situation where the obstacle contact forces cause the jammed joints to act 'against' each other. This situation may not always resolve by itself. The entire snake robot is therefore defined to be *jammed* if two or more joints are jammed. If the robot is jammed over a continuous period longer than $t_{\text{jam,max}}$, the leader-follower scheme is stopped in order to carry out the jam resolution scheme. We let the robot execute jam resolution for a predefined amount of time $t_{\text{resolution,max}}$ since it is difficult to come up with a specific criterion for when a jam has been resolved. Subsequently, the leader-follower scheme continues.

In summary, the design parameters $\Delta\phi_{\max}$, $t_{\text{jam,max}}$, and $t_{\text{resolution,max}}$ determine the switching between the *leader-follower* scheme and the *jam resolution* scheme.

12.4.3 The Jam Resolution Scheme

As explained in the introduction, the main purpose of the controller proposed in this chapter is to investigate the control principle proposed in Hypothesis 12.1. To this end, we define the jam resolution scheme in accordance with this control principle, namely, we rotate the links affected by contact forces so that the propulsive component of each contact force increases. In a jammed situation, the propulsive components of the contact forces from the obstacles are too small to overcome the friction forces opposing the motion of the robot. Rotating the contacted links (and thereby the direction of the contact forces) to increase the total propulsive contact force should therefore resolve the jammed situation.

The propulsive component of the contact force on link $i \in \{1, \ldots, N\}$, denoted by $\rho_{x,i}$, was defined in (12.3). The change of the propulsive force due to a change of the link angle is found by differentiating (12.3) with respect to θ_i, which gives

$$\frac{\partial \rho_{x,i}}{\partial \theta_i} = -\rho_i \cos\theta_i. \tag{12.9}$$

We see that changing a link angle near perpendicular to the direction of motion (large θ_i) has a greater effect on the propulsive force than a similar change of a link angle near parallel to the direction of motion (small θ_i). During jam resolution, we therefore prioritize to rotate links with a high propulsive force gradient with respect to the link angle, which suggests that the link angles should be changed according to

$$\Delta\theta_{i,\mathrm{ref}} = k_\theta \frac{\partial \rho_{x,i}}{\partial \theta_i} = -k_\theta \rho_i \cos\theta_i, \tag{12.10}$$

where $k_\theta > 0$ is a controller gain.

Let us now derive how the joint angles ϕ_{i-1} and ϕ_i at each side of link i should be changed to comply with (12.10). We choose that the contact force on link i only should affect the angle of link i, so that $\Delta\theta_{i-1,\mathrm{ref}} = \Delta\theta_{i+1,\mathrm{ref}} = 0$. Since we have from (2.1) that $\phi_i = \theta_i - \theta_{i+1}$, we can immediately write the desired change of the joint angles ϕ_{i-1} and ϕ_i due to the contact force on link i as

$$\Delta\phi_{i-1,\mathrm{ref}} = \Delta\theta_{i-1,\mathrm{ref}} - \Delta\theta_{i,\mathrm{ref}} = k_\theta \rho_i \cos\theta_i, \tag{12.11}$$

$$\Delta\phi_{i,\mathrm{ref}} = \Delta\theta_{i,\mathrm{ref}} - \Delta\theta_{i+1,\mathrm{ref}} = -k_\theta \rho_i \cos\theta_i. \tag{12.12}$$

By combining the desired change of joint angle ϕ_i due to the measured contact forces on the link at each side of the joint, i.e. the contribution from both ρ_i and ρ_{i+1}, we get that the angle of joint $i \in \{1, \ldots, N-1\}$ should be changed during jam resolution as

$$\Delta\phi_{i,\mathrm{ref}} = k_\theta \left(-\rho_i \cos\theta_i + \rho_{i+1} \cos\theta_{i+1}\right). \tag{12.13}$$

We now explain two important controller design choices. First of all, during jam resolution, we leave the head joint angle ϕ_{N-1} unchanged to maintain a smooth head angle. We thereby avoid that any jam resolution motion of the head link propagates backwards to all other links once the leader-follower scheme resumes. This would create undesirable body shapes. Secondly, we use the contact forces that were measured at the instant the jam resolution scheme was initiated as feedback so that the force measurements used in the feedback loop are *constant* during jam resolution. This ensures a steady rotation of the contacted links in accordance with the contact forces that produced the jam. If the force measurements had been updated *during* jam resolution, then jam resolution would very quickly be aborted for most of the contacted links because the link rotation carried out during jam resolution generally causes the links to detach from the obstacles. We denote the measured contact forces on all links at the instant the jam resolution scheme was initiated by $\boldsymbol{\rho}_{\mathrm{jam}} = \left[\rho_{\mathrm{jam},1}, \ldots, \rho_{\mathrm{jam},N}\right]^T \in \mathbb{R}^N$.

From the above discussion, the reference angles for all the joints of the snake robot in the jam resolution scheme are

$$\phi_{N-1,\mathrm{ref}} = \phi_{N-1}, \tag{12.14a}$$

$$\phi_{i,\mathrm{ref}} = \phi_i + k_\theta \left(-\rho_{\mathrm{jam},i} \cos\theta_i + \rho_{\mathrm{jam},i+1} \cos\theta_{i+1}\right), \tag{12.14b}$$

where $i \in \{1, \ldots, N-2\}$ and k_θ is a design parameter.

12.4.4 The Joint Angle Controller

The *leader-follower* scheme and the *jam resolution* scheme provide the joint reference angles $\boldsymbol{\phi}_{\mathrm{ref}} = \left[\phi_{1,\mathrm{ref}}, \ldots, \phi_{N-1,\mathrm{ref}}\right]^T \in \mathbb{R}^{N-1}$. To ensure that the reference angles comply with the maximum allowable deflection of the joints, which we specify as $[-\phi_{\max}, \phi_{\max}]$ for some $\phi_{\max} > 0$, we saturate the reference angle of joint $i \in \{1, \ldots, N-1\}$ according to

$$\widehat{\phi}_{i,\mathrm{ref}} = \max\left(\min\left(\phi_{i,\mathrm{ref}}, \phi_{\max}\right), -\phi_{\max}\right), \tag{12.15}$$

where $\max(\cdot)$ and $\min(\cdot)$ are operators that return the maximum and minimum value of their arguments, respectively. To make the joint angles $\boldsymbol{\phi} = \left[\phi_1, \ldots, \phi_{N-1}\right]^T \in \mathbb{R}^{N-1}$ track the saturated reference angles in (12.15), we set the actuator torque of joint $i \in \{1, \ldots, N-1\}$ according to the PD-controller

$$u_i = k_p \left(\widehat{\phi}_{i,\mathrm{ref}} - \phi_i\right) - k_d \dot{\phi}_i, \tag{12.16}$$

where $k_p > 0$ and $k_d > 0$ are controller gains. A velocity reference is not included in (12.16) since the transitions between the schemes of the control strategy produce steps in the reference angles, which would lead to large and undesirable velocity references.

Remark 12.4 *Compliance is an important issue during control based on force feedback. However, there is no need to explicitly consider compliance for the proposed control strategy since we do not attempt to explicitly control the contact forces on the snake robot. Note that the proportional action of the joint torque controller in (12.16) introduces compliance in the system since the dynamic properties of a proportional controller are similar to those of a mechanical spring.*

12.4.5 The Complete Hybrid Controller

Based on the controller schemes presented in the preceding subsections, we now provide a formal and precise specification of the complete *hybrid*

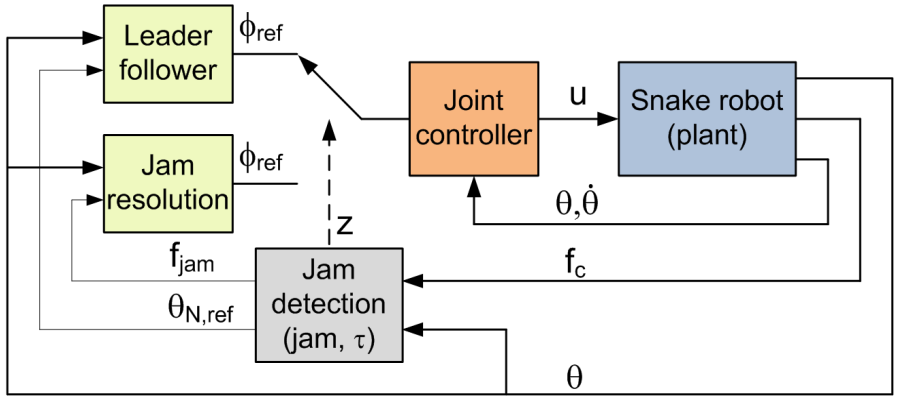


Figure 12.3: The block diagram of the closed-loop system (hybrid plant and hybrid controller).

controller of the snake robot in terms of the modelling framework briefly described in Section 12.1. We begin by defining the state vector of the hybrid controller as

$$\boldsymbol{\eta} = \begin{bmatrix} \theta_{N,\mathrm{ref}} \\ \boldsymbol{\rho}_{\mathrm{jam}} \\ \mathrm{jam} \\ \tau \\ z \end{bmatrix}, \tag{12.17}$$

where $\theta_{N,\mathrm{ref}} \in \{\theta_{\mathrm{left}}, \theta_{\mathrm{right}}\}$ is the current reference angle of the head link, $\boldsymbol{\rho}_{\mathrm{jam}} \in \mathbb{R}^N$ denotes the measured constraint forces on all links at the instant the jam resolution scheme is initiated, $\mathrm{jam} \in \{0, 1\}$ is a boolean variable indicating if the robot is currently jammed ($\mathrm{jam} = 1$ indicates a jam), $\tau \in \mathbb{R}_{\geq 0}$ is a timer variable, and finally $z \in \{0, 1\}$ is a boolean variable that decides if the robot should currently execute the *leader-follower* scheme ($z = 0$) or the *jam resolution* scheme ($z = 1$). The block diagram of the closed-loop system is illustrated in Fig. 12.3, where the state vector $\boldsymbol{\eta}$ is maintained inside the *jam detection* block. The state of the hybrid plant in (10.61) and the state of the hybrid controller is hereafter written as $(\boldsymbol{x}, \boldsymbol{\eta})$.

In the following subsections, we define the *jump set* $\boldsymbol{D}_c$, *jump map* $\boldsymbol{G}_c$, *flow set* $\boldsymbol{C}_c$, and *flow map* $\boldsymbol{F}_c$ of the hybrid controller, where subscript c is used to distinguish these sets from the corresponding sets of the hybrid plant in Section 10.7.

The Jump Set

A *jump* in the state vector $\boldsymbol{\eta}$ of the controller occurs either when the direction of the head link should change, when the jam state changes, or when the jam resolution scheme is initiated or stopped (controlled by switching the value of z).

In direct accordance with Section 12.4.1, the direction of the head link should change when the state of the plant and the controller, $(\boldsymbol{x}, \boldsymbol{\eta})$, belongs to the jump set

$$\begin{aligned} \boldsymbol{D}_{\text{dir}} = \quad & \left\{(\boldsymbol{x}, \boldsymbol{\eta}) \,|z = 0, \theta_{N,\text{ref}} = \theta_{\text{left}}, \Delta y > \Delta y_{\max}\right\} \quad \cup \\ & \left\{(\boldsymbol{x}, \boldsymbol{\eta}) \,|z = 0, \theta_{N,\text{ref}} = \theta_{\text{right}}, \Delta y < -\Delta y_{\max}\right\}. \end{aligned} \tag{12.18}$$

In order to determine the jam state, we define the following index set corresponding to pairs of jammed joints:

$$\boldsymbol{I}_{\text{jam}} = \left\{(i,j) \,|i \neq j, \left|\phi_i - \phi_{i,\text{ref}}\right| > \Delta\phi_{\max}, \left|\phi_j - \phi_{j,\text{ref}}\right| > \Delta\phi_{\max}\right\}. \tag{12.19}$$

We consider pairs of jammed joints since the robot is defined to be jammed when two or more joints are jammed, i.e. when $\boldsymbol{I}_{\text{jam}} \neq \emptyset$. In accordance with Section 12.4.2, the jump set of the jam state is given by

$$\begin{aligned} \boldsymbol{D}_{\text{jam}} = \quad & \left\{(\boldsymbol{x}, \boldsymbol{\eta}) \,|z = 0, \text{jam} = 0, \boldsymbol{I}_{\text{jam}} \neq \emptyset\right\} \quad \cup \\ & \left\{(\boldsymbol{x}, \boldsymbol{\eta}) \,|z = 0, \text{jam} = 1, \boldsymbol{I}_{\text{jam}} = \emptyset\right\}. \end{aligned} \tag{12.20}$$

In accordance with Section 12.4.2, the switching variable z should change when $(\boldsymbol{x}, \boldsymbol{\eta})$ belongs to the jump set

$$\begin{aligned} \boldsymbol{D}_{\text{res}} = \quad & \left\{(\boldsymbol{x}, \boldsymbol{\eta}) \,|z = 0, \text{jam} = 1, \tau > t_{\text{jam,max}}\right\} \quad \cup \\ & \left\{(\boldsymbol{x}, \boldsymbol{\eta}) \,|z = 1, \tau > t_{\text{resolution,max}}\right\}. \end{aligned} \tag{12.21}$$

The complete *jump set* of the hybrid controller may now be compactly expressed as

$$\boldsymbol{D}_c = \boldsymbol{D}_{\text{dir}} \cup \boldsymbol{D}_{\text{jam}} \cup \boldsymbol{D}_{\text{res}}. \tag{12.22}$$

The Jump Map

From the description of the controller schemes in Sections 12.4.1, 12.4.2, and 12.4.3, we can directly state the *jump map* of the hybrid controller as

$$\boldsymbol{\eta}^+ = \boldsymbol{G}_c\left(\boldsymbol{x}, \boldsymbol{\eta}\right) = \begin{bmatrix} \theta^+_{N,\text{ref}} \\ \boldsymbol{\rho}^+_{\text{jam}} \\ \text{jam}^+ \\ \tau^+ \\ z^+ \end{bmatrix}, \tag{12.23}$$

where

$$\theta^{+}_{N,\mathrm{ref}} = \begin{cases} \theta_{\mathrm{left}} & \text{when} \quad (\boldsymbol{x},\boldsymbol{\eta}) \in \boldsymbol{D}_{\mathrm{dir}}, \Delta y < -\Delta y_{\max} \\ \theta_{\mathrm{right}} & \text{when} \quad (\boldsymbol{x},\boldsymbol{\eta}) \in \boldsymbol{D}_{\mathrm{dir}}, \Delta y > \Delta y_{\max} \\ \theta^{-}_{N,\mathrm{ref}} & \text{otherwise} \end{cases} \tag{12.24a}$$

$$\boldsymbol{\rho}^{+}_{\mathrm{jam}} = \boldsymbol{f}_c, \tag{12.24b}$$

$$\mathrm{jam}^{+} = \begin{cases} 1 & \text{when} \quad (\boldsymbol{x},\boldsymbol{\eta}) \in \boldsymbol{D}_{\mathrm{jam}}, \mathrm{jam} = 0 \\ 0 & \text{when} \quad ((\boldsymbol{x},\boldsymbol{\eta}) \in \boldsymbol{D}_{\mathrm{jam}}, \mathrm{jam} = 1) \\ & \qquad \vee (\boldsymbol{x},\boldsymbol{\eta}) \in \boldsymbol{D}_{\mathrm{res}} \\ \mathrm{jam}^{-} & \text{otherwise} \end{cases}, \tag{12.24c}$$

$$\tau^{+} = \begin{cases} 0 & \text{when} \quad (\boldsymbol{x},\boldsymbol{\eta}) \in \boldsymbol{D}_{\mathrm{jam}} \cup \boldsymbol{D}_{\mathrm{res}} \\ \tau^{-} & \text{otherwise} \end{cases}, \tag{12.24d}$$

$$z^{+} = \begin{cases} 1 & \text{when} \quad (\boldsymbol{x},\boldsymbol{\eta}) \in \boldsymbol{D}_{\mathrm{res}}, z = 0 \\ 0 & \text{otherwise} \end{cases}. \tag{12.24e}$$

The Flow Set

We define the *flow set* of the hybrid controller so that the state vector $\boldsymbol{\eta}$ always flows as long as the jump set is empty. The flow set is therefore simply given as

$$\boldsymbol{C}_c = \{(\boldsymbol{x},\boldsymbol{\eta}) \,|\, (\boldsymbol{x},\boldsymbol{\eta}) \notin \boldsymbol{D}_c\}. \tag{12.25}$$

The Flow Map

The only variable in the state vector $\boldsymbol{\eta}$ that should change between jumps is the timer variable, τ. Since the time derivative of τ is 1, the *flow map* is given by

$$\dot{\boldsymbol{\eta}} = \boldsymbol{F}_c(\boldsymbol{x},\boldsymbol{\eta}) = \begin{bmatrix} \dot{\theta}_{N,\mathrm{ref}} \\ \dot{\boldsymbol{\rho}}_{\mathrm{jam}} \\ \dot{\mathrm{jam}} \\ \dot{\tau} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{0}_{N\times 1} \\ 0 \\ 1 \\ 0 \end{bmatrix}. \tag{12.26}$$

Calculation of the Control Input for the Plant

The joint torques $\boldsymbol{u}$ of the snake robot are calculated as

$$\boldsymbol{u} = \begin{cases} \boldsymbol{\kappa}_{\mathrm{nojam}}(\boldsymbol{x},\boldsymbol{\eta}) & \text{when} \quad z = 0 \\ \boldsymbol{\kappa}_{\mathrm{jam}}(\boldsymbol{x},\boldsymbol{\eta}) & \text{when} \quad z = 1 \end{cases}, \tag{12.27}$$

where $\boldsymbol{\kappa}_{\text{nojam}} : \boldsymbol{C}_c \to \mathbb{R}^{N-1}$ is defined by (12.8) and (12.16), and $\boldsymbol{\kappa}_{\text{jam}} : \boldsymbol{C}_c \to \mathbb{R}^{N-1}$ is defined by (12.14) and (12.16).

12.5 Summary of the Closed-loop System

We are now ready to summarize the complete closed-loop system consisting of the hybrid model of the snake robot from Chapter 10 and the hybrid controller presented in Section 12.4.5. The block diagram of the closed-loop system is illustrated in Fig. 12.3.

In accordance with the hybrid modelling framework presented in Section 10.1.1, the closed-loop system is a hybrid system with state $(\boldsymbol{x}, \boldsymbol{\eta})$ and data $(\boldsymbol{C}, \boldsymbol{F}, \boldsymbol{D}, \boldsymbol{G})$, where $\boldsymbol{x}$ is the state of the hybrid plant in (10.61) and $\boldsymbol{\eta}$ is the state of the hybrid controller defined in (12.17). The closed-loop system flows as long as neither $\boldsymbol{x}$ nor $\boldsymbol{\eta}$ should jump. In other words, $(\boldsymbol{x}, \boldsymbol{\eta})$ flows as long as both $\boldsymbol{x} \in \boldsymbol{C}_p$ and $(\boldsymbol{x}, \boldsymbol{\eta}) \in \boldsymbol{C}_c$, and jumps when $\boldsymbol{x} \in \boldsymbol{D}_p$ or $(\boldsymbol{x}, \boldsymbol{\eta}) \in \boldsymbol{D}_c$. The closed-loop system can therefore be written as

$$\begin{aligned} \begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{\eta}} \end{bmatrix} &= \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{\eta}) \quad \text{for all} \quad (\boldsymbol{x}, \boldsymbol{\eta}) \in \boldsymbol{C}, \\ \begin{bmatrix} \boldsymbol{x}^+ \\ \boldsymbol{\eta}^+ \end{bmatrix} &= \boldsymbol{G}(\boldsymbol{x}, \boldsymbol{\eta}) \quad \text{for all} \quad (\boldsymbol{x}, \boldsymbol{\eta}) \in \boldsymbol{D}, \end{aligned} \tag{12.28}$$

where

$$\boldsymbol{C} = \{(\boldsymbol{x}, \boldsymbol{\eta}) \,|\, \boldsymbol{x} \in \boldsymbol{C}_p \wedge (\boldsymbol{x}, \boldsymbol{\eta}) \in \boldsymbol{C}_c\}, \tag{12.29}$$

$$\boldsymbol{D} = \{(\boldsymbol{x}, \boldsymbol{\eta}) \,|\, \boldsymbol{x} \in \boldsymbol{D}_p \vee (\boldsymbol{x}, \boldsymbol{\eta}) \in \boldsymbol{D}_c\}, \tag{12.30}$$

$$\boldsymbol{x}^+ = \begin{cases} \boldsymbol{G}_p(\boldsymbol{x}) & \text{when} \quad \boldsymbol{x} \in \boldsymbol{D}_p \\ \boldsymbol{x}^- & \text{otherwise} \end{cases}, \tag{12.31}$$

$$\boldsymbol{\eta}^+ = \begin{cases} \boldsymbol{G}_c(\boldsymbol{x}, \boldsymbol{\eta}) & \text{when} \quad (\boldsymbol{x}, \boldsymbol{\eta}) \in \boldsymbol{D}_c \\ \boldsymbol{\eta}^- & \text{otherwise} \end{cases}. \tag{12.32}$$

Since, by design, the evolution of $\boldsymbol{\eta}$ always can be uniquely determined, the control input $\boldsymbol{u}$ to the plant is always well-defined. We can therefore conclude that the existence and uniqueness properties stated in Theorem 10.5 also apply to the closed-loop system in (12.28).

12.6 Simulation Study: The Performance of the Hybrid Controller

In this section, we investigate the performance of the hybrid controller described in Section 12.4 by simulating the closed-loop system summarized in (12.28).

12.6.1 Simulation Parameters

The simulator was implemented in *Matlab R2008b* on a laptop running *Windows XP*. The continuous dynamics in (12.28) was calculated with the *ode45* solver in Matlab with a relative and absolute error tolerance of 10^{-3}.

In accordance with the notation from Section 2.2, the parameters characterizing the simulated snake robot were $N = 10$, $l = 0.07$ m, $m = 1$ kg, and $J = 0.0016$ kgm^2. These parameters characterize the snake robot Kulko described in Chapter 11. The ground friction coefficient was set to $\mu = 0.3$ and the obstacle friction coefficient was set to $\mu_o = 0.2$. Two different obstacle environments were considered. In the first environment, the obstacles were chosen to be three rows (parallel to the x axis) of circular objects with equal radius $R_{O_j} = 10$ cm, $j \in \{1, \ldots, k\}$. The centre distance between two obstacles in a row and the distance between two rows were 25 cm. The middle row was displaced with respect to the other two rows by 12.5 cm along the x axis. In the second environment, obstacles of varying radius were placed in a random fashion around the snake robot.

The various parameters of the hybrid controller were $\theta_{\text{left}} = 50°$, $\theta_{\text{right}} = -50°$, $\Delta y_{\max} = 0.14$ m, $v_{\text{ref}} = 0.2$ m/s, $\Delta\phi_{\max} = 20°$, $t_{\text{jam,max}} = 0.5$ s, $t_{\text{resolution,max}} = 0.5$ s, $k_\theta = 0.05$, $\phi_{\max} = 50°$, $k_p = 20$, and $k_d = 5$. The initial link angles and position of the snake robot were $\boldsymbol{\theta} = [7°, -32°, -57°, -46°, -8°, 33°, 53°, 45°, 12°, -23°]^T$ and $\boldsymbol{p} = \mathbf{0}_{2\times 1}$, respectively. The initial shape was more or less randomly chosen in order to give the robot an initial curl around the obstacles without intersecting them.

In order to ensure a unique solution to the LCP in (10.43), we verified at each timestep of the simulation that the matrix $\boldsymbol{A}$ of the LCP in (10.43) was a P-matrix by employing the P-matrix test algorithm presented in Tsatsomeros and Li (2000).

12.6.2 Attempting Lateral Undulation in Open-loop in a Structured Obstacle Environment

We begin by illustrating the need for feedback of obstacle contact forces by controlling the snake robot, not according to the hybrid controller, but according to the open-loop lateral undulation gait defined in (4.32). This simulation is similar to the simulation described in Section 10.8. However, in the present simulation, we do *not* place the obstacles in accordance with the expected evolution of the shape and position of the snake robot as was done in Section 10.8. Instead, we place the obstacles according to the first obstacle environment described in the previous subsection. We chose the gait pattern parameters in (4.32) as $\alpha = 40°$, $\omega = 40°/\text{s}$, and $\delta = 40°$.

The initial ($t = 0$ s) and final ($t = 20$ s) shape and position of the snake robot are shown at the top of Fig. 12.4, where the trace of the head is indicated with a dotted line. We see that the robot was only able to crawl about 0.5 m in 20 s. Since there was no adaptation of the motion to the environment, the obstacles prevented the snake robot from assuming the predetermined body shape. This simulation illustrates that a snake robot is generally unable to locomote in a cluttered environment when the joint motion is preprogrammed.

12.6.3 Hybrid Controller in an Obstacle Environment

The next simulation shows the effectiveness of the hybrid controller proposed in this chapter when the snake robot moves for 30 s in the two obstacle environments described above. The initial ($t = 0$ s) and final ($t = 30$ s) shape and position of the snake robot in these two environments are shown in the middle and at the bottom of Fig. 12.4, respectively, while a plot of the x direction velocity of the snake, $\dot{p}_x$, is shown in Fig. 12.5. Vertical dashed lines in Fig. 12.5 indicate time instants where the jam resolution scheme is initiated. After 30 s, the snake robot has managed to crawl about 2.5 m along the global x axis in both environments. Fig. 12.5 shows that the velocity in both environments varies around 10 cm/s. The jam resolution scheme was initiated six and eight times, respectively, in the first and the second environment, and all the jams were successfully resolved by the proposed algorithm.

In order to give an idea of the forces involved in obstacle-aided locomotion, we provide a plot of the constraint forces on the centre link (link 5), ρ_5, at the top of Fig. 12.6. The actuator torque applied to joint 5, u_5, is plotted at the bottom of this figure. We see that constraint forces above 200

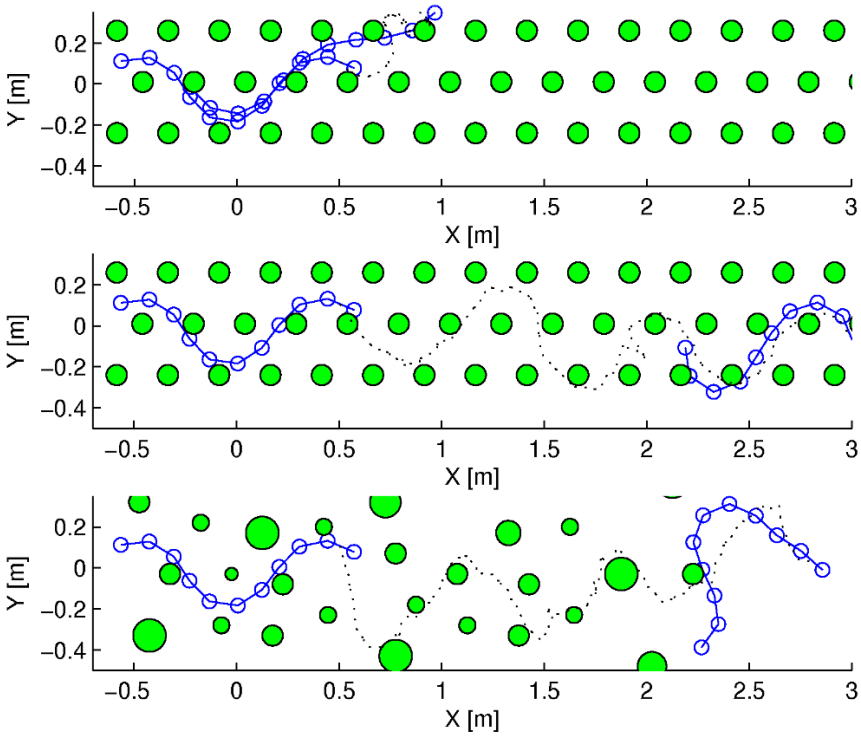


Figure 12.4: The initial and final shape and position, and the trace of the head of the snake robot. Top: Lateral undulation (jam resolution disabled). Middle: Jam resolution in the first obstacle environment. Bottom: Jam resolution in the second obstacle environment.

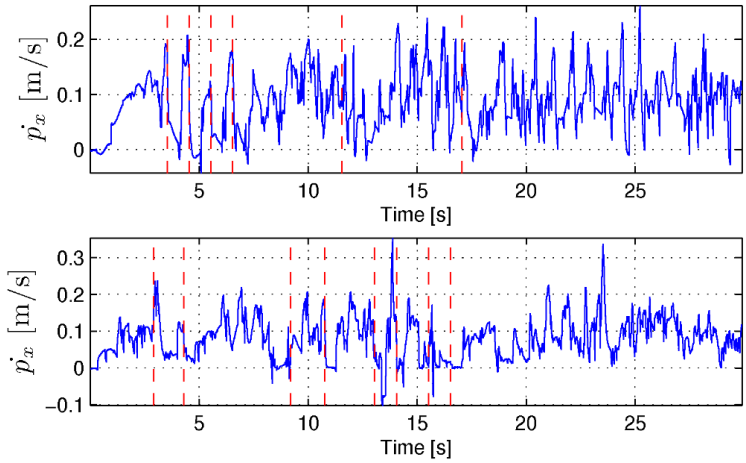


Figure 12.5: The global x direction velocity of the snake robot in the first (top) and second (bottom) obstacle environment. Vertical dashed lines indicate when the jam resolution scheme is initiated.

N occur during the motion, and that the applied joint torque is sometimes as high as 10 Nm.

To clearly illustrate the effect of the jam resolution scheme, a plot of the snake robot before (dashed) and after (solid) a jam resolution is shown in Fig. 12.7. The figure shows the jam occurring in the first obstacle environment at time $t = 17.05$ s and ending at time $t = 17.55$ s, which is caused by contact forces acting on links 4 and 8 (link 1 is the tail). The jam is resolved by rotating link 4 clockwise and link 8 counterclockwise, thereby increasing the propulsive components of the two constraint forces enough to overcome the friction forces from the ground and the obstacles.

In summary, the simulation results illustrate how the proposed jam detection and resolution scheme can help to maintain the propulsion of a snake robot in a cluttered environment. The successful performance of the jam resolution scheme is a support of the control principle proposed in Hypothesis 12.1.

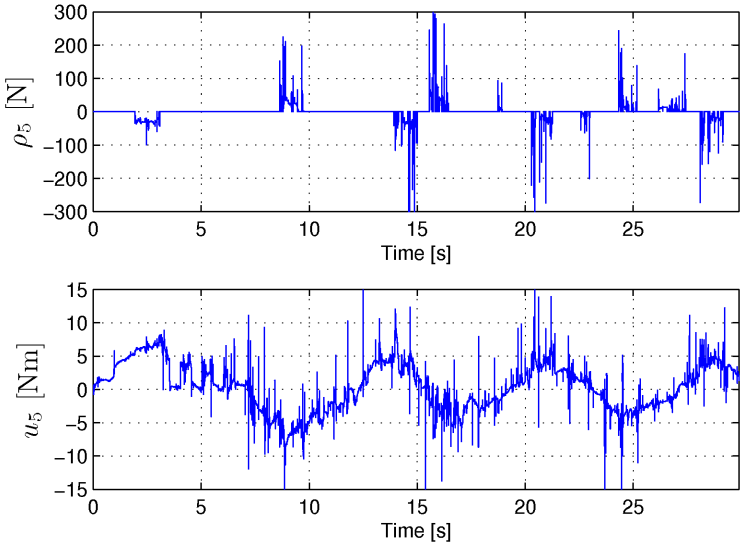


Figure 12.6: The obstacle constraint forces on link 5 (top) and the actuator torques applied to joint 5 (bottom) in the first obstacle environment.

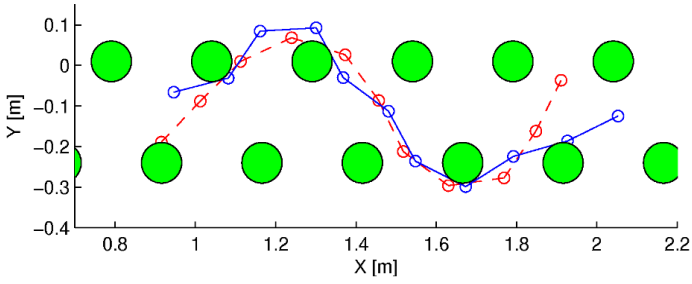


Figure 12.7: The shape of the snake robot before (dashed) and after (solid) an instance of the jam resolution scheme.

12.7 Experimental Study: The Performance of the Hybrid Controller

This section presents experimental results that illustrate the performance of the hybrid controller described in Section 12.4. The experiments show that the snake robot Kulko, which was described in Chapter 11, is propelled through several different obstacle courses when the robot is controlled according to the hybrid controller.

12.7.1 Experimental Setup

The experimental setup, which is described in more detail in Chapter 11, consisted of the snake robot Kulko and the camera-based position measurement system. The hybrid controller described in Section 12.4 was implemented on an external computer with the parameters $N = 10$, $l = 0.07$ m, $\theta_{\text{left}} = 50°$, $\theta_{\text{right}} = -50°$, $\Delta y_{\max} = 14$ cm, $v_{\text{ref}} = 5$ cm/s, $\Delta\phi_{\max} = 20°$, $t_{\text{jam,max}} = 1$ s, $t_{\text{resolution,max}} = 1$ s, and $k_\theta = 0.05$. The joint torque controller given by (12.16) was not implemented since accurate torque control is not supported by the servo motors installed in the snake robot. The joint angles were instead controlled according to a proportional controller implemented in the microcontroller of each joint module.

Three different obstacle environments were considered. The first obstacle environment contained five obstacles with x coordinates (-123.9, -89.6, -48.4, -8.2, -0.6) cm, y coordinates (20.2, -15.7, 13.2, -23.5, 24.8) cm, and diameters (30, 20, 30, 30, 20) cm, respectively. The second obstacle environment contained four obstacles with x coordinates (-90.9, -35.5, 5.1, 31.7) cm, y coordinates (-20.3, 4.2, -28.9, 15.9) cm, and diameters (30, 30, 30, 30) cm, respectively. The third and final obstacle environment contained four obstacles with x coordinates (-93.1, -79.4, -17.4, 14.6) cm, y coordinates (-61.7, -6.3, -18.9, 24.3) cm, and diameters (30, 30, 30, 30) cm, respectively.

Three trials were carried out in each obstacle environment. The initial link angles in the first, second, and third environment were approximately $\boldsymbol{\theta} = [49°, 43°, 6°, 14°, -19°, -20°, -3°, 16°, 11°, 1°]^T$, $\boldsymbol{\theta} = [58°, 47°, 25°, -14°, -35°, -27°, -12°, 3°, 28°, 25°]^T$, and $\boldsymbol{\theta} = [-4°, -4°, -2°, 15°, 30°, 65°, 40°, 29°, -3°, -22°]^T$, respectively. The initial position of the head link was $(x_N = 0, y_N = 0)$ and the initial reference angle for the head link was $\theta_{N,\text{ref}} = \theta_{\text{right}}$ in all three environments.

12.7.2 Experimental Results

The experimental results from the three obstacle environments are shown in Figures 12.8 and 12.9, Figures 12.10 and 12.11, and Figures 12.12 and 12.13, respectively. Three trials were carried out in each obstacle environment. The measured position of the head link along the forward direction (the global x axis) during the three trials in each environment are shown in Figures 12.8(a), 12.10(a), and 12.12(a), respectively, and the control scheme executed during each trial (i.e. *leader-following* or *jam resolution*) is shown in Figures 12.8(b), 12.10(b), and 12.12(b). Furthermore, the sideways position (along the global y axis) and the orientation of the head link are shown in Figures 12.8(c)-(d), 12.10(c)-(d), and 12.12(c)-(d). In order to give an idea of the forces needed to propel the robot forward, the measured contact forces on joint modules 3 - 8 (module 1 is the tail) during the first trial in each environment are shown in Figures 12.8(e)-(f), 12.10(e)-(f), and 12.12(e)-(f), respectively. The motion of the snake robot during the first trial in each environment is visualized in Figures 12.9, 12.11, and 12.13, respectively.

As seen by the plots of the head position along the forward direction in Figures 12.8(a), 12.10(a), and 12.12(a), the overall forward propulsion of the snake robot was maintained throughout the trials in all three environments. This was also the main goal of the experiments. In other words, using the same controller with the same set of controller parameters, the snake robot was able to move through three different obstacle environments. The plots of the sideways position and orientation of the head link suggest that the reference angles from the leader-follower scheme were rather different in the individual trials in each environment. However, there is a fairly good repeatability in the forward direction plots from the trials in each environment, which is indicative of the robustness and environment adaptability properties of the proposed controller.

The forward direction plots show that the forward speed of the robot was relatively slow in all trials. This limited speed was due to the limited torque of the joints of the snake robot compared to the rather large ground and obstacle friction forces opposing the motion. In particular, the snake robot is rather heavy (about 10 kg) compared to its maximum actuator torque (about 4 Nm). To cope with the limited strength of the physical snake robot, the propagation velocity of the head joint angle in the leader-follower scheme was set to a rather small value during the experiments, namely $v_{\text{ref}} = 5$ cm/s. Since v_{ref} determines the propagation velocity of the body waves produced during the locomotion, a small value of v_{ref} will naturally

lead to a small forward speed of the robot. In other words, we claim that the limited speed during the experiments was caused by limitations of the physical snake robot, and is not a general property of the proposed control strategy. Had the experiments been carried out using a snake robot with a larger actuator strength to weight ratio, then the controller parameters could have been adjusted to increase the forward speed significantly.

There is a clear tendency in the forward direction plots from all three environments that the forward velocity of the robot starts to decrease after about 50 s. The reason for this decrease in velocity is seen from the visualizations in Figures 12.9, 12.11, and 12.13, which show that the robot used about 50 s to pass through each obstacle course. Since the flat surface outside the obstacle courses contained no push-points that the robot could use for propulsion, the forward velocity decreased as the robot moved out of each obstacle course. The proposed control strategy is, in other words, dependent on external objects in order to propel a snake robot with isotropic ground friction properties forward.

As seen in Figures 12.8(b), 12.10(b), and 12.12(b), the snake robot was jammed and executed jam resolution several times during each trial. In order to visualize the behaviour of the snake robot during jam resolution, Fig. 12.14 shows the snake robot in the first obstacle environment at two time instants when it was jammed ($t = 6$ s and $t = 21$ s) and after jam resolution had been carried out ($t = 7$ s and $t = 22$ s). It is clearly seen from the figure that the jam resolution scheme increased the angles of the jammed joints with respect to the forward direction, thereby increasing the propulsive components of the subsequent obstacle contact forces at these locations. This behaviour is also the intended purpose of the jam resolution scheme, as described in Section 12.4.3.

The high number of jams that occurred during the trials in the three obstacle environments is an interesting observation since it suggests that the *jam* state of the snake robot should be treated as a *continuous* rather than a discrete state. Furthermore, the high number of jams suggests that leader-following should not be conducted in open-loop, but rather combined with continuous use of the measured contact forces. In particular, the jam resolution scheme was active a large number of times during the experiments because the leader-follower scheme did not consider the environment interaction, which caused the robot to become jammed over and over. Based on this observation, we will in Chapter 13 propose a continuous control law for obstacle-aided locomotion where jam resolution is a continuous action that is performed in parallel with the cyclic wave motion

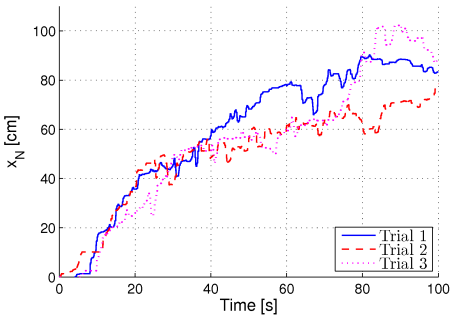
of the snake robot.

In summary, the experimental results indicate that the proposed control strategy is robust with respect to maintaining the overall forward propulsion of a snake robot in various obstacle environments. The ability of the robot to resolve jams clearly suggests that rotating links in contact with obstacles to increase the propulsive force on the robot is a control principle that should be pursued in further work on obstacle-aided locomotion. We therefore claim that the experimental results support Hypothesis 12.1.

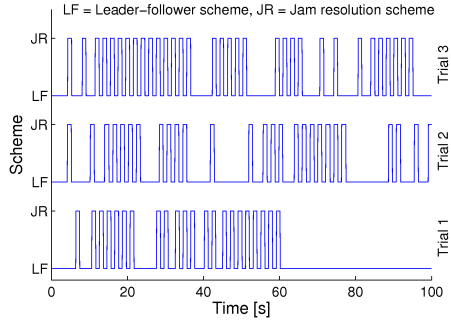
12.8 Chapter Summary

This chapter is summarized as follows:

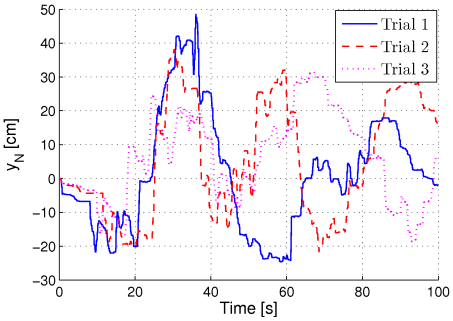
- We have proposed a fundamental control principle (see Hypothesis 12.1), where we claim that obstacle-aided snake robot locomotion is achieved by producing body shape changes where the links in contact with obstacles are rotated to increase the propulsive forces on the robot.
- We have investigated this control principle by using it as a basis for a proposed control strategy for obstacle-aided locomotion.
- The proposed control strategy is a *hybrid controller* aimed at resolving situations where the snake robot is jammed between obstacles.
- In particular, body waves are produced in open-loop in a *leader-follower* scheme as long as the robot is able to move without being jammed between the obstacles. If a *jam* is detected, then a *jam resolution* scheme is carried out in order to effectively 'unlock' the jammed joints.
- We have presented simulation results and experimental results based on the snake robot Kulko, where the hybrid controller was shown to maintain the propulsion of the snake robot in different obstacle environments.
- The high number of jams that occurred during the experiments with the snake robot suggests that the *jam* state of the robot should be treated as a *continuous* rather than a discrete state, and that leader-following should not be conducted in open-loop, but rather combined with continuous use of the measured contact forces.



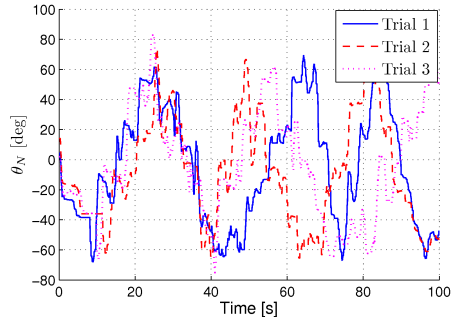
(a) The head coordinate, x_N.



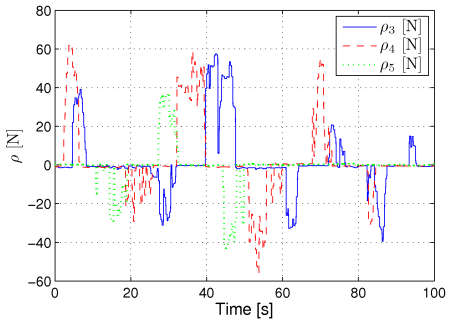
(b) The executed control scheme.



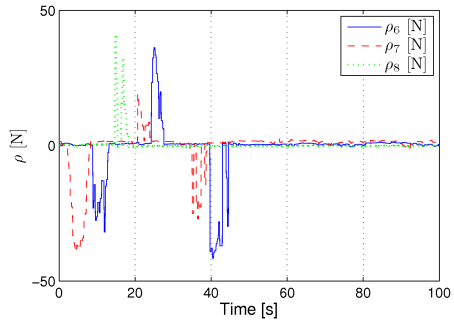
(c) The head coordinate, y_N.



(d) The head link angle, θ_N.



(e) Contact forces, ρ_3-ρ_5 (Trial 1).



(f) Contact forces, ρ_6-ρ_8 (Trial 1).

Figure 12.8: Experimental results of obstacle-aided locomotion in the first obstacle environment.

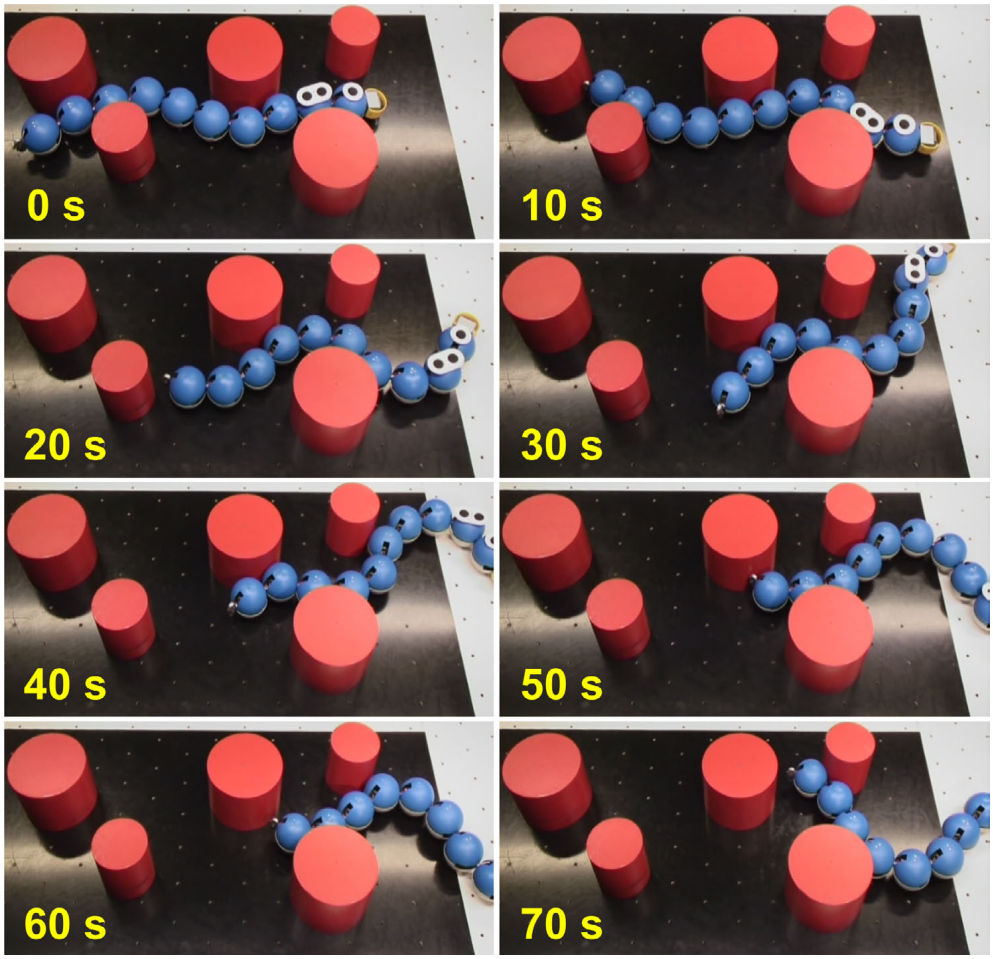
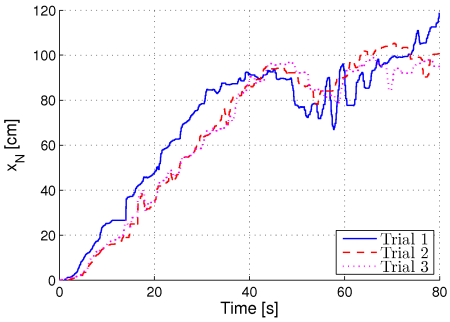
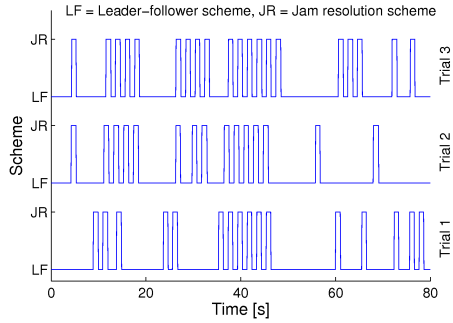


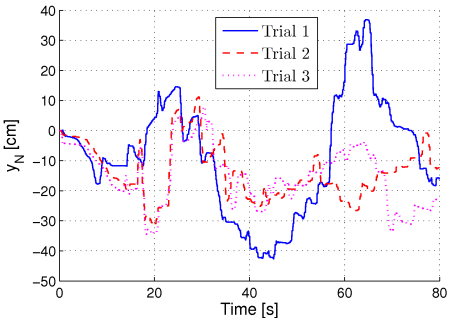
Figure 12.9: The motion of the snake robot in the first obstacle environment (Trial 1).



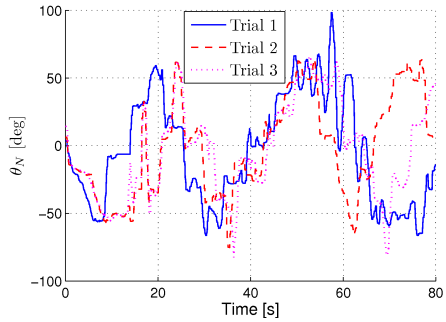
(a) The head coordinate, x_N.



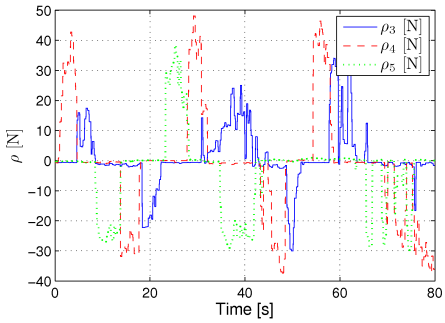
(b) The executed control scheme.



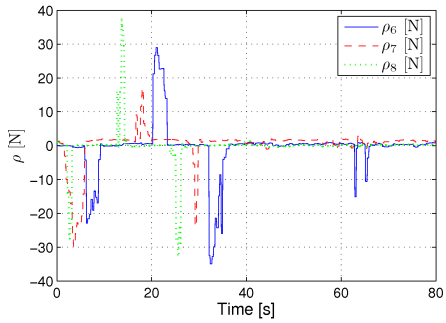
(c) The head coordinate, y_N.



(d) The head link angle, θ_N.



(e) Contact forces, ρ_3-ρ_5 (Trial 1).



(f) Contact forces, ρ_6-ρ_8 (Trial 1).

Figure 12.10: Experimental results of obstacle-aided locomotion in the second obstacle environment.

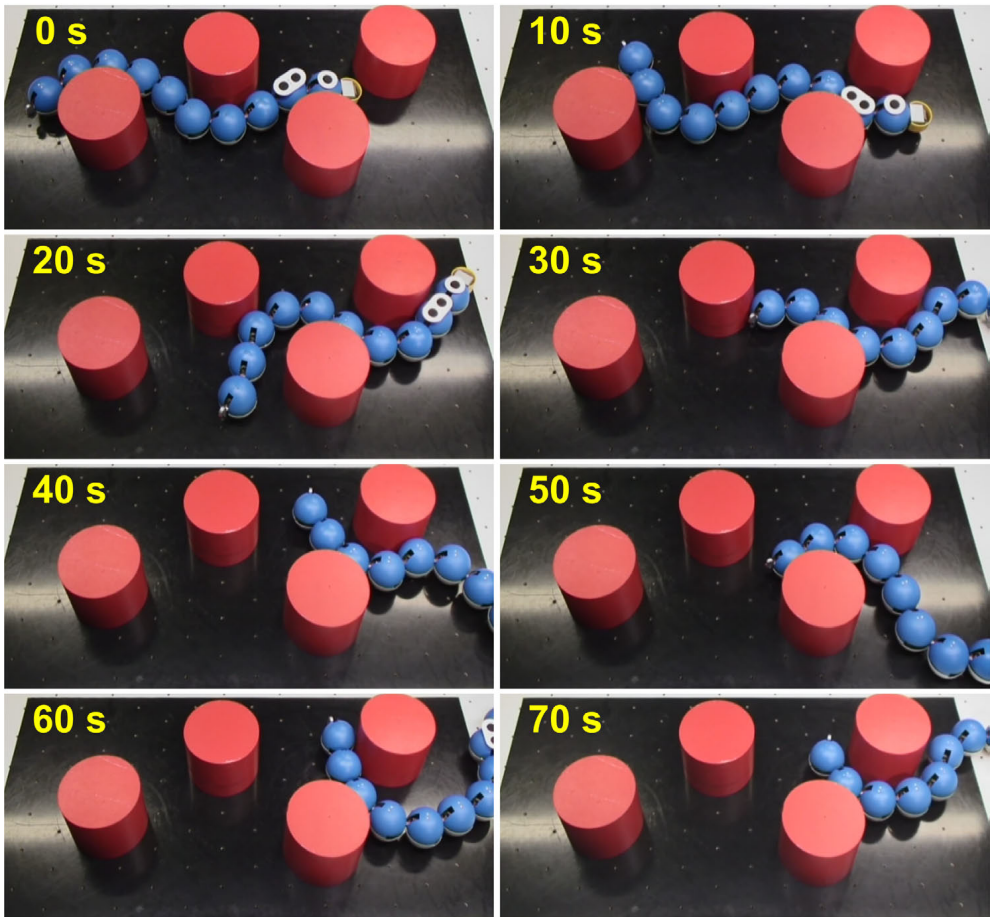
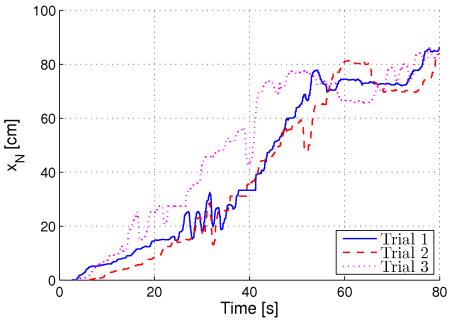
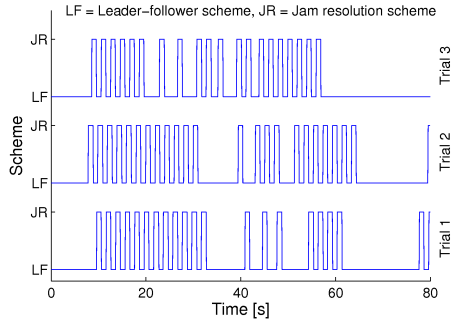


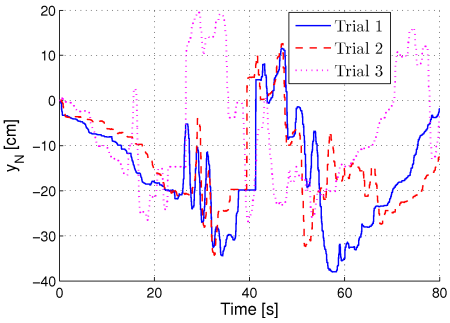
Figure 12.11: The motion of the snake robot in the second obstacle environment (Trial 1).



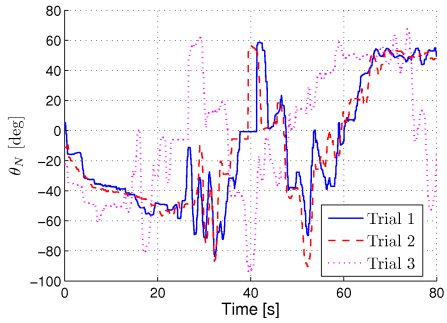
(a) The head coordinate, x_N.



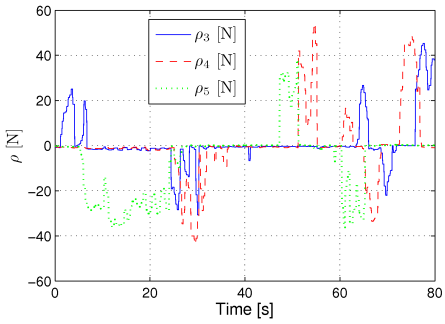
(b) The executed control scheme.



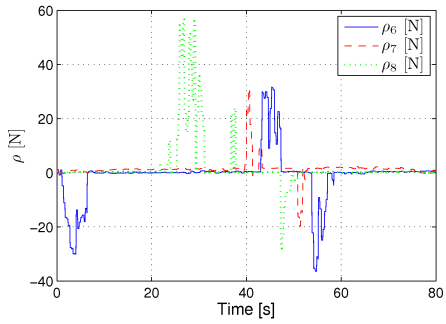
(c) The head coordinate, y_N.



(d) The head link angle, θ_N.



(e) Contact forces, ρ_3-ρ_5 (Trial 1).



(f) Contact forces, ρ_6-ρ_8 (Trial 1).

Figure 12.12: Experimental results of obstacle-aided locomotion in the third obstacle environment.

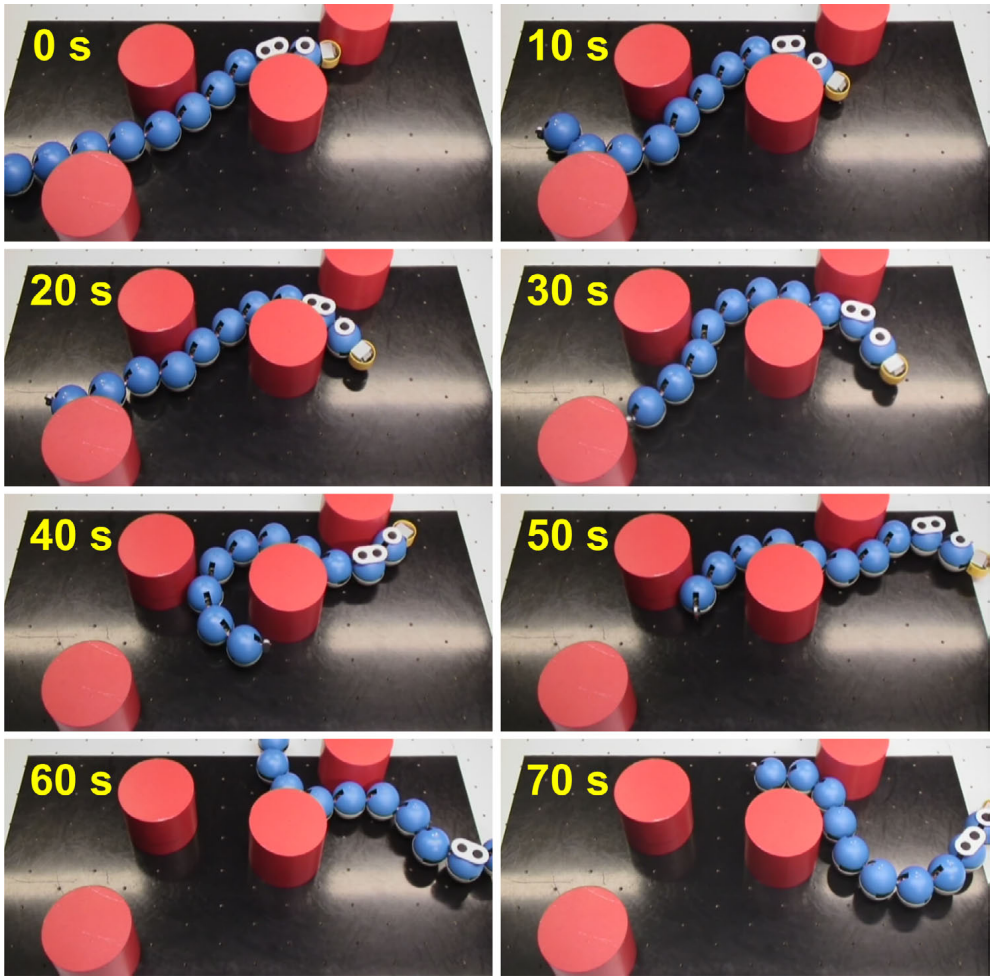


Figure 12.13: The motion of the snake robot in the third obstacle environment (Trial 1).

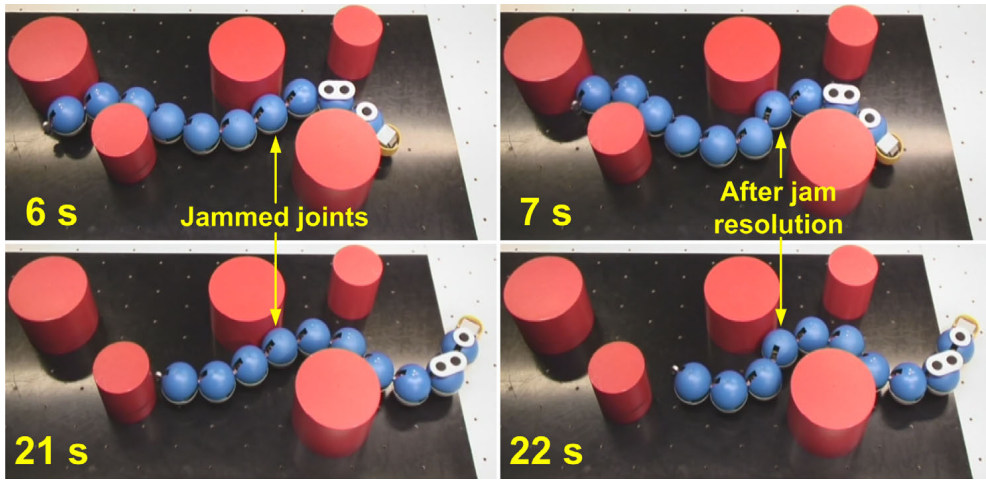


Figure 12.14: The shape of the snake robot before and after jam resolution in the first obstacle environment.

Chapter 13

Path Following Control of Snake Robots in Unstructured Environments

With respect to control design, this final chapter represents the culmination of this thesis, and can be regarded as a fusion of the control efforts reported in the previous chapters. In particular, forward propulsion of snake robots on planar surfaces based on oscillatory body shape changes was studied in Chapter 4. Subsequently, this oscillatory gait pattern was extended in Chapters 5 and 8 with directional control capabilities that enabled the snake robot to track straight paths and paths defined by waypoints. Furthermore, the oscillatory gait pattern was extended in Chapter 12 with environment adaptation capabilities that enabled the snake robot to maintain forward propulsion in unstructured environments with obstacles. Directional control was, however, not considered in Chapter 12.

In this chapter, we will employ the knowledge gained from these control design efforts to propose a general *framework* for motion control of snake robots. The framework allows the motion of the snake robot to be specified in terms of three separate components, namely a *body wave* component, an *environment adaptation* component, and a *heading control* component. The framework is subsequently used to propose a continuous control strategy for straight line path following control of snake robots in unstructured environments. We will also combine the proposed path following controller with the waypoint guidance strategy from Chapter 8 in order to enable a snake robot to move between waypoints in an obstacle environment. The performance of the path following controller and the waypoint guidance

strategy is illustrated with simulation results where a snake robot is successfully steered between waypoints in an environment with obstacles. We will also present results from an experimental validation of the path following controller by use of the snake robot Kulko.

Contributions of this Chapter: The first contribution of this chapter is the general framework for motion control of snake robots, where the motion is specified in terms of a *body wave* component, an *environment adaptation* component, and a *heading control* component. The second contribution is the control law (based on the controller framework) for straight line path following control of snake robots in environments with obstacles, and the fitting of this control law within the waypoint guidance strategy from Chapter 8. A significant contribution of the path following controller is the idea of a continuous jam resolution action that is performed in parallel with the cyclic wave motion of the robot to continuously adapt the body shape to the environment and prevent the motion from being jammed. This continuous jam resolution action is contained in the *environment adaptation* component of the controller, and is based on the jam resolution scheme from Chapter 12. However, whereas a complex hybrid formulation is employed in Chapter 12, the jam resolution action will in this chapter be specified in terms of simple continuous equations. We consider the step from a *hybrid* to a *continuous* formulation to be important since it makes a formal analysis of the controller more feasible. As a third contribution of this chapter, we present experimental results where the snake robot Kulko is successfully propelled through three different obstacle environments with the proposed controller.

Organization of this Chapter: Section 13.1 presents the general framework for motion control of snake robots. A straight line path following controller based on this framework is proposed in Section 13.2, and subsequently fitted within a waypoint guidance strategy in Section 13.3. Simulation results and experimental results that illustrate the successful performance of the path following controller are presented in Sections 13.4 and 13.5, respectively. Finally, the chapter is summarized in Section 13.6.

Publications: The material in this chapter is based on the journal paper Liljebäck *et al.* (2010*k*) and the conference paper Liljebäck *et al.* (2011*c*).

13.1 A Controller Framework for Snake Robot Locomotion

In this section, we propose a general framework for motion control of snake robots. To motivate this framework, we begin by stating a set of claims that we justify with reference to research results reported in the previous chapters of this thesis:

Claim 13.1 ***The controller should produce body wave motion:*** *During snake robot locomotion on flat surfaces, we showed in Chapter 4 that cyclic body wave motion produces forward propulsion under anisotropic ground friction conditions. The argument for also considering cyclic body wave motion in environments with obstacles was given in the beginning of Section 12.4.1. In particular, it was noted in Remark 10.2 in Section 10.3 that the phenomenon that propels a snake robot forward due to the anisotropic ground friction properties of the links is similar to the phenomenon that propels a snake robot forward due to the interaction of the links with external obstacles. Since cyclic body wave motion produces forward propulsion under anisotropic ground friction conditions, we can thereby expect this to also be the case in environments with obstacles. Note that the claim is also supported by the motion of biological snakes in nature.*

Claim 13.2 ***The controller should continuously perform environment adaptation:*** *The experimental investigation of the hybrid controller presented in Chapter 12, showed that the motion of the physical snake robot was jammed quite frequently. These experimental results indicate that conducting cyclic body wave motion in open-loop* will *eventually jam the motion of the robot, which strongly suggests that the cyclic body wave motion should* not *be conducted in open-loop, but rather adjusted continuously according to the interaction of the robot with its environment. We therefore claim that environment adaptation should be conducted continuously in parallel with the cyclic body wave motion of the snake robot.*

Claim 13.3 ***The controller should steer the heading:*** *This requirement is obvious in order to be able to steer the snake robot to a desired location. We demonstrated path following capabilities of a snake robot in Chapters 5 and 8.*

Based on the above claims, we propose the following general controller framework for snake robots:

Hypothesis 13.1 ***The controller framework.***
Efficient and intelligent snake robot locomotion in unknown and unstructured environments can be achieved by specifying the reference angles $\boldsymbol{\phi}_{\text{ref}} = \left[\phi_{1,\text{ref}}, \ldots, \phi_{N-1,\text{ref}}\right]^T \in \mathbb{R}^{N-1}$ *of the robot as the sum of three individual motion components, namely as*

$$\boldsymbol{\phi}_{\text{ref}} = \boldsymbol{\phi}_{wave} + \boldsymbol{\phi}_{adapt} + \boldsymbol{\phi}_{heading}, \tag{13.1}$$

where $\boldsymbol{\phi}_{wave}$ *is a* body wave *component that induces propulsive forces on the robot from the environment,* $\boldsymbol{\phi}_{adapt}$ *is an* environment adaptation *component that adjusts the body shape to the environment, and* $\boldsymbol{\phi}_{heading}$ *is a* heading control *component that steers the robot according to a specified reference direction.*

Remark 13.1 *The lateral undulation gait considered in Part I of this thesis, which is also considered in the majority of the literature on snake robot locomotion, fits nicely within the framework proposed in (13.1). As defined in (4.32), this gait pattern is achieved by controlling joint* i *of the snake robot according to*

$$\phi_{i,\text{ref}} = \underbrace{\alpha \sin\left(\omega t + (i-1)\,\delta\right)}_{\boldsymbol{\phi}_{wave}} + \underbrace{\phi_o}_{\boldsymbol{\phi}_{heading}}, \tag{13.2}$$

where the sinus term constitutes the body wave component, $\boldsymbol{\phi}_{wave}$*, and* ϕ_o*, which is an angular offset used to control the direction of the motion, constitutes the heading component,* $\boldsymbol{\phi}_{heading}$*. The gait pattern does not involve adaptation of the body shape to the environment, which means that* $\boldsymbol{\phi}_{adapt} = \mathbf{0}$*.*

Remark 13.2 *The environment adaptation component* $\boldsymbol{\phi}_{adapt}$ *requires that the snake robot can* sense *its environment in some way. We demonstrated this capability in Chapter 11 by describing the development of a snake robot with contact force sensors installed along its body. Snake robots with discrete contact switches are employed in Bayraktaroglu (2008); Hirose (1993). The environment can also be indirectly sensed through the joint angle measurements, as considered in Andruska and Peterson (2008); Date and Takita (2007).*

13.2 Straight Line Path Following Control in Unstructured Environments

In this section, we employ the controller framework presented in Section 13.1 to propose a straight line path following controller for snake robots in environments with obstacles. The *body wave* component of the control law is based on a *predecessor-follower* scheme, where each joint follows the angle of the preceding joint ahead of itself. This approach is an improvement over the leader-follower scheme considered in Chapter 12 since leader-following relies on the assumption that the robot moves forward with the same speed as the head angle propagates backward. The *environment adaptation* component is based on the jam resolution principle from Chapter 12. However, whereas a complex hybrid formulation is employed in Chapter 12, the jam resolution motion of the present controller is specified in terms of simple continuous equations. The *heading control* component is similar to a guidance law of the straight line path following controllers proposed in Chapters 5 and 8.

13.2.1 Control Objective

We choose the control objectives to be identical to the objectives of the path following controller proposed in Chapter 5. The objectives are restated here for completeness.

In order to track the desired straight path, we define the global coordinate system so that the global x axis is aligned with the desired path. The position of the snake robot along the global y axis, p_y, is then the shortest distance from the robot to the desired path (i.e. the cross-track error) and the heading $\overline{\theta}$ of the robot, which was defined in (2.2), is the angle that the robot forms with the desired path. The control objective is thereby to regulate p_y and $\overline{\theta}$ so that they oscillate about zero. We will not attempt to regulate p_y and $\overline{\theta}$ to zero since the heading and position of the robot are expected to display oscillating behaviour during locomotion along the desired path.

Since snake robot locomotion is a slow form of robotic mobility, which is generally employed for traversability purposes, we consider it less important to accurately control the forward velocity of the robot. During path following with a snake robot, it therefore makes sense to focus all the control efforts on converging to the path and subsequently progressing along the path at some nonzero forward velocity $\overline{v}_t(t) > 0$, where $\overline{v}_t(t)$ is the forward velocity of the robot defined in (2.5).

From the above discussion, the control problem is to design a feedback control law such that for all $t > t_c \geq 0$, there exists a $\tau \in [t, t+T]$ satisfying

$$p_y(\tau) = 0, \tag{13.3}$$

$$\overline{\theta}(\tau) = 0, \tag{13.4}$$

$$\overline{v}_t(t) > 0, \tag{13.5}$$

where t_c is some (unknown) finite time duration corresponding to the time it takes the snake robot to converge to the desired straight path, and $T > 0$ is some constant that characterize the time period of the cyclic gait pattern of the snake robot. In other words, we require that p_y and $\overline{\theta}$ are zero at least once *within* each cycle of the locomotion since this means that p_y and $\overline{\theta}$ oscillate about zero. Note that we require $\overline{v}_t(t) > 0$ for all $t > t_c$.

The idea behind the controller proposed in the following is to use the body wave component $\boldsymbol{\phi}_{\text{wave}}$ and the adaptation component $\boldsymbol{\phi}_{\text{adapt}}$ to achieve control objective (13.5), and simultaneously use the heading component $\boldsymbol{\phi}_{\text{heading}}$ to achieve control objectives (13.3) and (13.4).

13.2.2 Notation and Basic Assumptions

So far in this thesis, we have described the link angles $\boldsymbol{\theta}$ according to Definition 2.1, i.e. with respect to the global x axis. During path following in an obstacle environment, however, we will employ a measure of the link angles with respect to the current heading $\overline{\theta}$ defined in (2.2). As illustrated in Fig. 13.1, we define this measure as follows:

Definition 13.1 ***The heading-adjusted link angle.***
The heading-adjusted *angle of link* $i \in \{1, \ldots, N\}$ *is denoted by* $\widetilde{\theta}_i$ *and is given as the angle of link* i *with respect to the current heading* $\overline{\theta}$*, i.e. as*

$$\widetilde{\theta}_i = \theta_i - \overline{\theta}. \tag{13.6}$$

Control objective (13.5) simply states that the robot should maintain a positive and nonzero forward velocity. This means that the desired forward direction of motion at any time is along the current heading of the robot. In accordance with Definition 12.2 in Section 12.3, the *propulsive component* of the contact force on link $i \in \{1, \ldots, N\}$ is therefore given as the component of the constraint force vector $\boldsymbol{f}_{c,i}$ along the current heading $\overline{\theta}$. This component, which we denote by $\rho_{\text{prop},i} \in \mathbb{R}$, is illustrated in Fig. 13.1 and is easily calculated as

$$\rho_{\text{prop},i} = -\rho_i \sin \widetilde{\theta}_i, \tag{13.7}$$

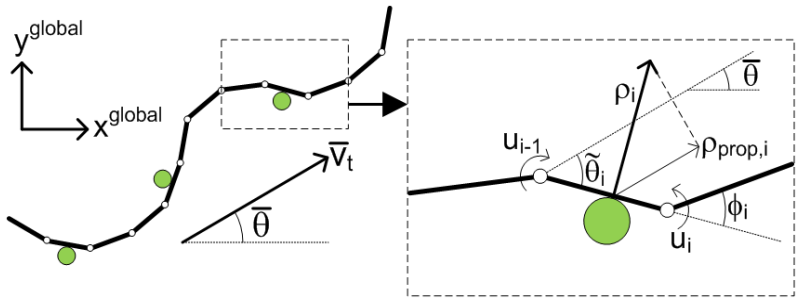


Figure 13.1: Illustration of the heading-adjusted angle of link i, denoted by $\widetilde{\theta}_i$, and the propulsive component of the contact force on link i, denoted by $\rho_{\text{prop},i}$.

where ρ_i was defined in Section 12.3 as the measured contact force on link $i \in \{1, \ldots, N\}$.

Remark 13.3 *In Chapter 12, the propulsive component of a contact force was denoted by $\rho_{x,i}$ and given as the force component along the global x axis. As long as the robot is headed along the global x axis, we have that $\rho_{x,i} = \rho_{\text{prop},i}$. In contrast to $\rho_{x,i}$, however, $\rho_{\text{prop},i}$ represents a suitable measure of the propulsive force on link i regardless of the heading of the robot.*

We will base the path following controller on the following assumption:

Assumption 13.1 *The control system has access to measurements of the cross-track error p_y, the joint angles $\boldsymbol{\phi}$, the joint angle velocities $\dot{\boldsymbol{\phi}}$, the contact forces $\boldsymbol{\rho}$, and at least one of the absolute link angles θ_i for some $i \in \{1, \ldots, N\}$.*

Note that the remaining link angles, and thereby also the heading $\overline{\theta}$, can be calculated from $\boldsymbol{\phi}$ and θ_i.

13.2.3 The Body Wave Component

We begin by presenting the body wave component $\boldsymbol{\phi}_{\text{wave}}$ of the joint reference angles. The hybrid controller proposed in Chapter 12 produces body waves through a leader-follower approach, where the angle of the foremost (head) joint is propagated backwards along the snake body at a constant

velocity and used as the reference angle for all subsequent joints. The drawback of this approach is that it relies on the assumption that the snake robot moves forward with the same speed as the head angle propagates backward. If this is not the case, then the head joint angle will generally not be a suitable reference angle for the remaining joints.

In the following, we therefore employ a *predecessor-follower* scheme, where each joint follows the angle of the preceding joint ahead of itself with a specified time delay Δt. The angle of joint i is always a suitable reference angle for joint $i-1$ since the current shape of the snake robot always represents a feasible reference trajectory. The resulting reference angle of joint $i \in \{1, \ldots, N-2\}$ in this predecessor-follower scheme can be written

$$\phi_{\text{wave},i}(t) = \phi_{i+1}(t - \Delta t). \tag{13.8}$$

In order to produce body wave motion, we introduce a sinusoidal reference angle for the heading-adjusted angle of the head link, $\widetilde{\theta}_N$, given by

$$\widetilde{\theta}_{N,\text{ref}}(t) = \alpha \sin(\omega t), \tag{13.9}$$

where α and ω are the amplitude and angular frequency, respectively, of the sinusoidal motion. Since the head joint (i.e. joint $N-1$) is at the front of the snake robot, the rotation of the head joint mainly affects the angle of the head link (i.e. link N) and not the angle of the subsequent links. Since it follows from (2.1) that $\phi_{N-1} = \widetilde{\theta}_{N-1} - \widetilde{\theta}_N$, we can track the head link reference angle in (13.9) by controlling the head joint according to the reference

$$\phi_{\text{wave},N-1}(t) = \widetilde{\theta}_{N-1} - \alpha \sin(\omega t). \tag{13.10}$$

From (13.8) and (13.10), we can now write the complete body wave component $\boldsymbol{\phi}_{\text{wave}}$ in matrix form as

$$\boldsymbol{\phi}_{\text{wave}} = \boldsymbol{S}_{\text{head}} \left(\widetilde{\theta}_{N-1} - \alpha \sin(\omega t) \right) + \boldsymbol{S}_{\text{joints}} \boldsymbol{\phi}(t - \Delta t), \tag{13.11}$$

where $\boldsymbol{\phi}(t - \Delta t)$ are the measured joint angles at time $t - \Delta t$, and where $\boldsymbol{S}_{\text{head}}$ and $\boldsymbol{S}_{\text{joints}}$ are, respectively, a selection vector and a selection matrix

defined as

$$\boldsymbol{S}_{\text{head}} = [0, \ldots, 0, 1]^T \in \mathbb{R}^{N-1}, \tag{13.12}$$

$$\boldsymbol{S}_{\text{joints}} = \begin{bmatrix} 0 & 1 & 0 & & \\ & 0 & 1 & 0 & \\ & & \ddots & \ddots & \\ & & & 0 & 1 \\ & & & & 0 \end{bmatrix} \in \mathbb{R}^{(N-1)\times(N-1)}. \tag{13.13}$$

Remark 13.4 *The implementation of (13.11) requires a buffer which keeps track of the angle history of each joint except for the first (tail) joint.*

13.2.4 The Environment Adaptation Component

The environment adaptation component $\boldsymbol{\phi}_{\text{adapt}}$ is based on the control principle proposed in Hypothesis 12.1, that was implemented in the jam resolution scheme of the hybrid controller from Chapter 12. However, whereas a complex hybrid formulation is employed in Chapter 12, the jam resolution action of the present control strategy will be specified in terms of simple continuous equations. We consider the step from a *hybrid* to a *continuous* formulation to be a significant contribution since a continuous formulation makes a formal analysis of the controller more feasible. As explained below in Remark 13.5, a formal analysis of the present control strategy remains a topic of future work.

In accordance with the derivation of the jam resolution scheme in Section 12.4.3, the idea behind the adaptation strategy is to rotate the links affected by contact forces so that the propulsive component of each contact force increases. Since the propulsive components of the contact forces are what propel the snake robot forward, we conjecture that rotating the contacted links to increase the total propulsive force will adapt the body shape to the environment in a way that maintains or increases the propulsion of the robot. Note that the adaptation strategy only aims at satisfying control objective (13.5), i.e. propelling the snake robot forward in the direction of its current heading.

The change of the propulsive force on link $i \in \{1, \ldots, N\}$ due to a change of the link angle is found by differentiating (13.7) with respect to $\widetilde{\theta}_i$, which gives

$$\frac{\partial \rho_{\text{prop},i}}{\partial \widetilde{\theta}_i} = -\rho_i \cos \widetilde{\theta}_i. \tag{13.14}$$

During adaptation, we choose to rotate links with a high propulsive force gradient with respect to the link angle, which suggests that link i is rotated according to

$$\Delta\widetilde{\theta}_{i,\mathrm{ref}} = k_\rho \frac{\partial \rho_{\mathrm{prop},i}}{\partial \widetilde{\theta}_i} = -k_\rho \rho_i \cos\widetilde{\theta}_i, \tag{13.15}$$

where $k_\rho > 0$ is a controller gain.

Let us now derive how the joint angles ϕ_{i-1} and ϕ_i at each side of link i should be changed to comply with (13.15). We choose that the contact force on link i only should affect the angle of link i, so that $\Delta\widetilde{\theta}_{i-1,\mathrm{ref}} = \Delta\widetilde{\theta}_{i+1,\mathrm{ref}} = 0$. Since we have from (2.1) that $\phi_i = \widetilde{\theta}_i - \widetilde{\theta}_{i+1}$, we can immediately write the desired change of the joint angles ϕ_{i-1} and ϕ_i due to the contact force on link i as

$$\Delta\phi_{i-1,\mathrm{ref}} = \Delta\widetilde{\theta}_{i-1,\mathrm{ref}} - \Delta\widetilde{\theta}_{i,\mathrm{ref}} = k_\rho \rho_i \cos\widetilde{\theta}_i, \tag{13.16}$$

$$\Delta\phi_{i,\mathrm{ref}} = \Delta\widetilde{\theta}_{i,\mathrm{ref}} - \Delta\widetilde{\theta}_{i+1,\mathrm{ref}} = -k_\rho \rho_i \cos\widetilde{\theta}_i. \tag{13.17}$$

By combining the desired change of joint angle ϕ_i due to the measured contact forces on the link at each side of the joint, i.e. the contribution from both ρ_i and ρ_{i+1}, we get that the angle of joint $i \in \{1, \ldots, N-1\}$ in the environment adaptation component $\boldsymbol{\phi}_{\mathrm{adapt}}$ is given by

$$\phi_{\mathrm{adapt},i} = -k_\rho \left(\rho_i \cos\widetilde{\theta}_i - \rho_{i+1} \cos\widetilde{\theta}_{i+1} \right). \tag{13.18}$$

The complete environment adaptation component $\boldsymbol{\phi}_{\mathrm{adapt}}$ can thereby be written in matrix form as

$$\boldsymbol{\phi}_{\mathrm{adapt}} = -k_\rho \boldsymbol{D}\, \mathrm{diag}(\boldsymbol{\rho}) \cos\widetilde{\boldsymbol{\theta}}, \tag{13.19}$$

where $\mathrm{diag}\,(\cdot)$ produces a diagonal matrix with the elements of its argument along its diagonal, and where

$$\boldsymbol{\rho} = [\rho_1, \ldots, \rho_N]^T \in \mathbb{R}^N, \tag{13.20}$$

$$\cos\widetilde{\boldsymbol{\theta}} = \left[\cos\widetilde{\theta}_1, \ldots, \cos\widetilde{\theta}_N\right]^T \in \mathbb{R}^N, \tag{13.21}$$

$$\boldsymbol{D} = \begin{bmatrix} 1 & -1 & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \end{bmatrix} \in \mathbb{R}^{(N-1)\times N}. \tag{13.22}$$

13.2.5 The Heading Control Component

The heading control component $\boldsymbol{\phi}_{\text{heading}}$ of the joint reference angles is similar to the guidance law of the straight line path following controllers proposed in Chapters 5 and 8. In particular, we steer the snake robot towards the desired straight path by employing the Line-of-Sight (LOS) guidance law

$$\overline{\theta}_{\text{ref}} = -\arctan\left(\frac{p_y}{\Delta}\right), \tag{13.23}$$

where p_y is the cross-track error and $\Delta > 0$ is a design parameter referred to as the *look-ahead distance* that influences the rate of convergence to the desired path. As illustrated in Fig. 13.2, the LOS angle $\overline{\theta}_{\text{ref}}$ corresponds to the orientation of the snake robot when it is headed towards the point located a distance Δ ahead of itself along the desired path. To steer the heading $\overline{\theta}$ according to the LOS angle in (13.23), we employ the same approach that was used for controlling the direction of the snake robot with the lateral undulation gait in (4.32), i.e. we offset the reference angle of the head joint according to

$$\phi_{\text{heading},N-1} = k_\theta \left(\overline{\theta} - \overline{\theta}_{\text{ref}}\right), \tag{13.24}$$

where $k_\theta > 0$ is a controller gain. Using (13.12), the heading component can be written in matrix form as

$$\boldsymbol{\phi}_{\text{heading}} = \boldsymbol{S}_{\text{head}} k_\theta \left(\overline{\theta} - \overline{\theta}_{\text{ref}}\right). \tag{13.25}$$

13.2.6 The Joint Angle Controller

In order to make the joint angles $\boldsymbol{\phi}$ track the reference angles given by $\boldsymbol{\phi}_{\text{ref}}$, we set the joint actuator torques $\boldsymbol{u}$ according to the PD-controller

$$\boldsymbol{u} = k_p \left(\boldsymbol{\phi}_{\text{ref}} - \boldsymbol{\phi}\right) + k_d \left(\dot{\boldsymbol{\phi}}_{\text{ref}} - \dot{\boldsymbol{\phi}}\right), \tag{13.26}$$

where $k_p > 0$ and $k_d > 0$ are controller gains.

13.2.7 Summary of the Path Following Controller

The complete straight line path following controller is now summarized. In accordance with the general controller framework defined in (13.1), we conjecture that control objectives (13.3), (13.4), and (13.5) are achieved by employing the PD-controller in (13.26) to control the joint angles of the snake robot according to

$$\boldsymbol{\phi}_{\text{ref}} = \boldsymbol{\phi}_{\text{wave}} + \boldsymbol{\phi}_{\text{adapt}} + \boldsymbol{\phi}_{\text{heading}}, \tag{13.27}$$

where

$$\boldsymbol{\phi}_{\text{wave}} = \boldsymbol{S}_{\text{head}} \left(\widetilde{\theta}_{N-1} - \alpha \sin(\omega t) \right) + \boldsymbol{S}_{\text{joints}} \boldsymbol{\phi}(t - \Delta t), \quad (13.28)$$

$$\boldsymbol{\phi}_{\text{adapt}} = -k_\rho \boldsymbol{D} \operatorname{diag}(\boldsymbol{\rho}) \cos \widetilde{\boldsymbol{\theta}}, \quad (13.29)$$

$$\boldsymbol{\phi}_{\text{heading}} = \boldsymbol{S}_{\text{head}} k_\theta \left(\overline{\theta} - \overline{\theta}_{\text{ref}} \right). \quad (13.30)$$

Remark 13.5 *Due to the complexity of the hybrid model of the snake robot in (10.61), we are currently unable to provide a formal proof of the achievement of objectives (13.3), (13.4), and (13.5) with the proposed controller. It is probably not possible to develop such a proof solely based on the model and control strategy of the robot combined with knowledge of the obstacle locations since it is difficult, if not impossible, to analytically predict the interaction between the robot and the obstacles in advance. However, it may be possible to develop logical arguments regarding the achievement of the control objectives by making assumptions regarding the obstacle interactions. One approach could be to parametrize the contact situation in some way and show that, in certain contact situations, the control action defined by (13.26) and (13.27) will move the snake robot a step closer to the desired path. Such logical arguments remain a topic of future work.*

13.3 Waypoint Guidance Control in Unstructured Environments

The straight line path following controller defined by (13.26) and (13.27) can easily be combined with the waypoint guidance strategy proposed in Section 8.4. We thereby arrive at a complete control strategy for steering a snake robot between waypoints in an unstructured environment, i.e. we obtain a control strategy for obstacle-aided locomotion along arbitrary paths given by waypoints interconnected by straight lines. In the following, we describe the waypoint guidance strategy from Section 8.4 within the context of this chapter.

The reason for specifying the path of the robot in terms of waypoints is that future applications of snake robots will generally involve bringing sensors and/or tools to a single or several specified target location(s). In these situations, the exact path taken by the robot as it moves towards the target(s) is generally of less interest as long as the robot reaches the target(s) within a reasonable amount of time. Specifying the motion of a snake robot in terms of waypoints supports this target-oriented control approach.

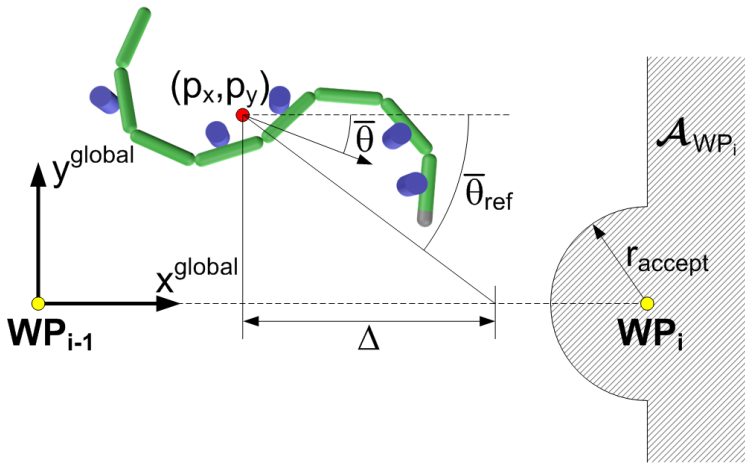


Figure 13.2: Straight line path following control of the snake robot combined with waypoint guidance in an obstacle environment.

There are k waypoints and the ith waypoint is denoted by WP_i, where $i \in \{1, \ldots, k\}$. As illustrated in Fig. 13.2, we interconnect the waypoints by straight lines and control the snake robot according to (13.27) in order to steer the robot towards the straight line leading to the next waypoint. The next waypoint is activated as soon as the position of the robot enters inside an *acceptance region* consisting of an *acceptance circle* (with radius r_{accept}) centred in the current waypoint and also the right half plane of a coordinate system with origo in the current waypoint and x axis pointing away from the previous waypoint (see illustration in Fig. 13.2). The acceptance region of WP_i is denoted by $\mathcal{A}_{WP_i}$. The intended purpose of employing the straight line path following controller is to steer the robot into the acceptance circle of the current waypoint. However, in situations where the obstacle environment prevents the robot from entering inside the acceptance circle, the robot will still proceed towards the next waypoint as soon as the position enters inside the right half plane contained in the acceptance region. With only acceptance circles enclosing each waypoint, there would be the risk that the robot misses a waypoint, e.g. due to the placement of the obstacles, which would make the robot proceed indefinitely along the path away from the waypoint that was missed.

The above definitions were formalized in Section 8.4. We can now state the waypoint guidance strategy as follows:

Algorithm 13.1 ***The waypoint guidance strategy.***

1. Define the initial position of the snake robot as WP_0.
2. Repeat for all $i \in \{0, \cdots, k-1\}$:
   (a) Move the origin of the global frame to WP_i and orient the global x axis towards WP_{i+1}.
   (b) Conduct path following according to (13.27) until $(p_x, p_y) \in \mathcal{A}_{WP_{i+1}}$.

13.4 Simulation Study: The Performance of the Path Following Controller

This section presents simulation results that illustrate how the path following controller defined by (13.26) and (13.27) performs in combination with the waypoint guidance strategy in Algorithm 13.1.

13.4.1 Simulation Parameters

The hybrid model of the snake robot (10.61) and the guidance strategy in Algorithm 13.1 were implemented in *Matlab R2008b* on a laptop running *Windows XP*. The continuous dynamics of the hybrid model were calculated with the *ode45* solver in Matlab with a relative and absolute error tolerance of 10^{-3}.

The parameters characterizing the simulated snake robot were $N = 10$, $l = 0.07$ m, $m = 1$ kg, and $J = 0.0016$ kgm^2. Circular obstacles measuring 10 cm in diameter were placed in a random fashion in the environment of the snake robot. The ground and obstacle friction coefficients were $\mu = 0.3$ and $\mu_o = 0.25$, respectively. The initial link angles and position of the snake robot were $\boldsymbol{\theta} = [-30°, -10°, 30°, 60°, 40°, 0°, -40°, -60°, -30°, 0°]^T$ and $\boldsymbol{p} = \mathbf{0}_{2\times 1}$, respectively.

We defined $k = 5$ waypoints with global frame coordinates $(2.5, 0)$, $(2.5, 1)$, $(0, 1)$, $(1, 2)$, and $(3, 2)$, respectively. The radius of the acceptance circle enclosing each waypoint was $r_{\text{accept}} = 0.5$ m. The remaining controller parameters were $\Delta = 0.7$ m, $k_\theta = 1.3$, $k_\rho = 0.02$, $\Delta t = 0.7$ s, $\alpha = 60°$, $\omega = 40°/\text{s}$, $k_p = 20$, and $k_d = 5$. In order to prevent the measured contact forces in $\boldsymbol{\phi}_{\text{adapt}}$ from producing steps in the joint reference angles $\boldsymbol{\phi}_{\text{ref}}$ in (13.27), the reference angles were filtered using a 2nd order low-pass filtering reference model (see Appendix C.1). The parameters of the reference model

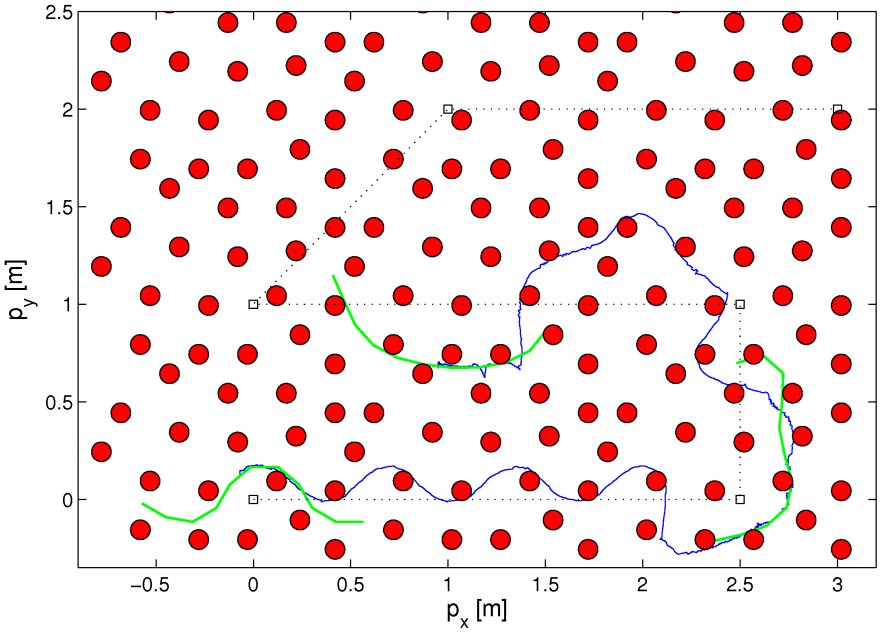
were $\omega = 3\pi/2$ and $\zeta = 1$. This filter also provided the derivative of $\boldsymbol{\phi}_{\text{ref}}$ with respect to time, which is needed by the PD-controller in (13.26).

13.4.2 Simulation Results

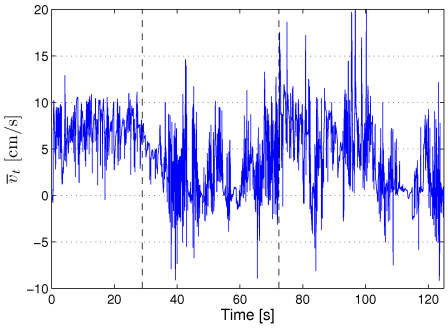
To illustrate the importance of environment adaptation, the path following controller of the waypoint guidance strategy was first simulated *without* adaptation ($\boldsymbol{\phi}_{\text{adapt}} = \mathbf{0}$). The path of the centre link of the snake robot (link 5) is shown in blue in Fig. 13.3(a), where black squares indicate the waypoints, the dotted black lines indicate the straight paths between the waypoints, and where the shape and position of the robot are shown in green at $t = 0$ s, $t = 65$ s, and $t = 125$ s, respectively. Furthermore, Fig. 13.3(b)-(c) show the forward velocity, $\overline{v}_t$, and the obstacle constraint force on link 5, ρ_5, respectively. The vertical dashed lines in the plots indicate time instants where the guidance strategy switched to the next waypoint. We see from Fig. 13.3(a) that the robot managed to reach the acceptance region of the two first waypoints. However, the motion was jammed about halfway to the third waypoint, as can be seen from Fig. 13.3(b), which shows that the forward velocity varied around zero after about 110 s. Note that there is a slight overlap between the path of link 5 and some of the obstacles. This issue was commented in Remark 10.4 of Section 10.3, and is a consequence of modelling obstacle contact solely by a unilateral force on the contacted link, which means that there is nothing preventing the foremost link (the head) of the snake robot from penetrating an obstacle head-on along its tangential direction.

The same plots for the case where environment adaptation was present (i.e. where $\boldsymbol{\phi}_{\text{adapt}}$ was set according to (13.29)) are shown in Fig. 13.4 and Fig. 13.5. In addition to the forward velocity and the obstacle constraint forces, Fig. 13.5 also shows the cross-track error p_y and the heading angle $\overline{\theta}$. With environment adaptation, the propulsion of the robot was maintained through all the waypoints. As seen from Fig. 13.5(c), the forward velocity varied between 5 - 10 cm/s, which suggests that control objective (13.5) was achieved. Figures 13.5(a)-(b) show that the cross-track error and the heading angle had an oscillatory behaviour around zero after each waypoint switch, which suggests that control objectives (13.3) and (13.4) were also achieved.

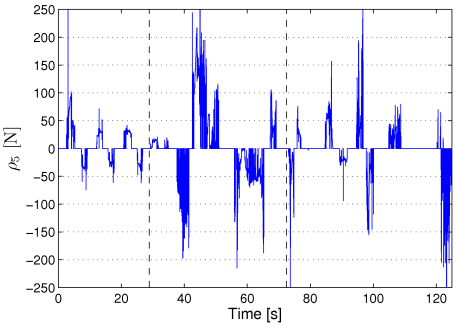
It is interesting to note that, in addition to improving the propulsion of the robot, the environment adaptation strategy reduces the constraint forces on the robot significantly, which is seen by comparing Fig. 13.5(d) with Fig. 13.3(c). This is also expected since the environment contact forces



(a) The path of the centre link (link 5) of the snake robot.



(b) Forward velocity, $\overline{v}_t$.



(c) Contact force on link 5, ρ_5.

Figure 13.3: Simulation of the waypoint guidance strategy *without* environment adaptation.

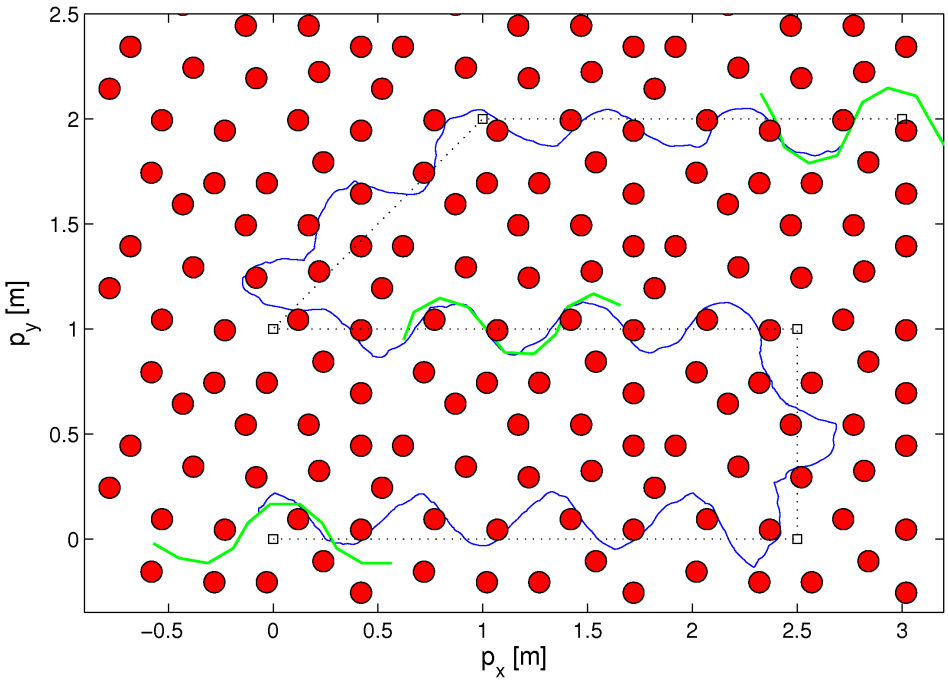


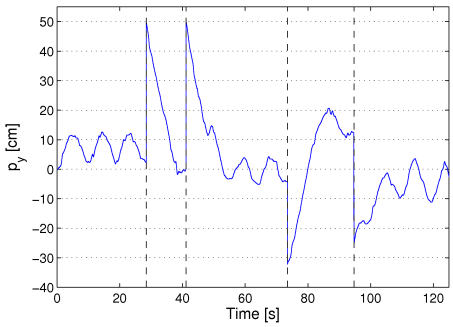
Figure 13.4: The path of the centre link (link 5) of the snake robot during waypoint guidance *with* environment adaptation.

opposing the motion will naturally be larger when the motion is performed without considering the environment.

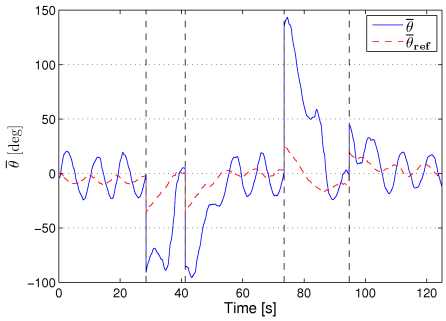
In summary, the path following controller defined by (13.26) and (13.27) combined with the waypoint guidance strategy in Algorithm 13.1 maintained the propulsion and steered the snake robot to the acceptance region of each waypoint in the obstacle environment.

13.5 Experimental Study: The Performance of the Environment Adaptation Strategy

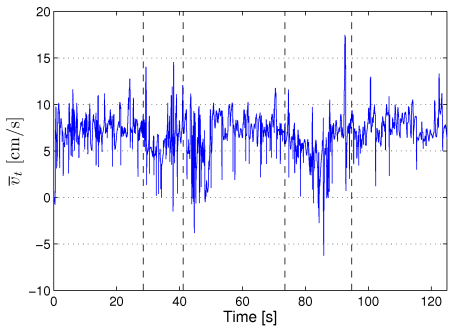
In this section, we present experimental results in order to demonstrate that a snake robot is propelled forward in an obstacle environment when the joints are controlled according to (13.27). In particular, the results demonstrate the propulsion produced by the body wave component in (13.28)



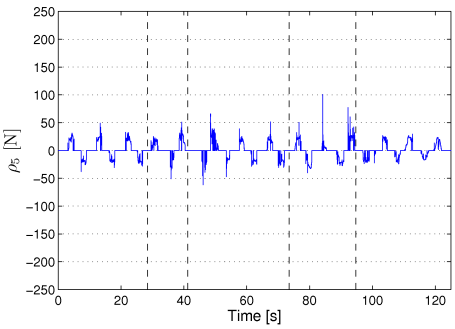
(a) Cross-track error, p_y.



(b) Heading angle, $\overline{\theta}$.



(c) Forward velocity, $\overline{v}_t$.



(d) Contact force on link 5, ρ_5.

Figure 13.5: Simulation of the waypoint guidance strategy *with* environment adaptation.

and the environment adaptation component in (13.29). Due to the limited range of the obstacle course, we did not consider heading control during the experiments, which means that the heading control component in (13.30) was set to zero.

13.5.1 Experimental Setup

We employed the same experimental setup that was used to investigate the jam resolution controller in Section 12.7. In particular, the experimental setup consisted of the snake robot Kulko and the camera-based position measurement system presented in Chapter 11. The joint reference angles defined by (13.27) were calculated on an external computer with the parameters $N = 10$, $l = 0.07$ m, $k_\rho = 0.01$, $\Delta t = 0.9$ s, $\alpha = 60°$, and $\omega = 30°/\text{s}$, and with $\boldsymbol{\phi}_{\text{heading}} = 0$. The joint torque controller given by (13.26) was not implemented since accurate torque control is not supported by the servo motors installed in the snake robot. The joint angles were instead controlled according to a proportional controller implemented in the microcontroller of each joint module.

We considered three obstacle environments with similar obstacle configuration as in the experiments described in Section 12.7. The first obstacle environment contained five obstacles with x coordinates (-123.9, -89.6, -48.4, -8.2, -0.6) cm, y coordinates (20.2, -15.7, 13.2, -23.5, 24.8) cm, and diameters (30, 20, 30, 30, 20) cm, respectively. The second obstacle environment contained four obstacles with x coordinates (-90.9, -35.5, 5.1, 31.7) cm, y coordinates (-20.3, 4.2, -28.9, 15.9) cm, and diameters (30, 30, 30, 30) cm, respectively. The third and final obstacle environment contained five obstacles with x coordinates (-93.1, -79.4, -45.4, -17.4, 14.6) cm, y coordinates (-61.7, -6.3, 29.4, -18.9, 24.3) cm, and diameters (30, 30, 20, 30, 30) cm, respectively.

The initial link angles in the first, second, and third environment were $\boldsymbol{\theta} = [49°, 43°, 6°, 14°, -19°, -20°, -3°, 16°, 11°, 1°]^T$, $\boldsymbol{\theta} = [58°, 47°, 25°, -14°, -35°, -27°, -12°, 3°, 28°, 25°]^T$, and $\boldsymbol{\theta} = [-4°, -4°, -2°, 15°, 30°, 65°, 40°, 29°, -3°, -22°]^T$, respectively, and the initial position of the head link was $(x_N = 0, y_N = 0)$.

13.5.2 Experimental Results

The experimental results from the three obstacle environments are shown in Figures 13.6 and 13.7, Figures 13.8 and 13.9, and Figures 13.10 and 13.11, respectively. As seen by the plots of the head position along the

forward direction in Figures 13.6(a), 13.8(a), and 13.10(a), the overall forward propulsion of the robot was maintained throughout all three trials. This was also the main goal of the experiments. In other words, using the same controller with the same set of controller parameters, the snake robot was able to move through three different obstacle environments, which we consider to be evidence that the proposed control strategy provides a snake robot with environment adaptation skills.

As explained in conjunction with the experimental results of Section 12.7, the forward speed of the robot was relatively slow in all trials mainly due to the limited torque of the joints of the snake robot compared to the rather large ground and obstacle friction forces opposing the motion. In particular, the snake robot is rather heavy (about 10 kg) compared to its maximum actuator torque (about 4 Nm). We therefore claim that the limited speed during the experiments was caused by limitations of the physical snake robot, and is not a general property of the proposed control strategy. Had the experiments been carried out using a snake robot with a larger actuator strength to weight ratio, then the controller parameters could have been adjusted to increase the forward speed significantly.

To give an idea of the forces needed to propel the robot forward in the three environments, the measured contact forces on joint modules 4, 5, and 6 (module 1 is the tail) are shown in Figures 13.6(d), 13.8(d), and 13.10(d), respectively. We see that contact forces in the range 30 - 60 N occurred during the motion.

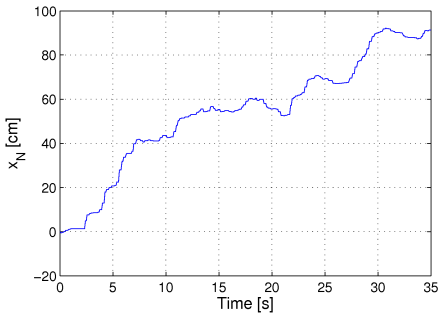
The motion of two joints of the snake robot during the three trials is shown in Figures 13.6(e)-(f), 13.8(e)-(f), and 13.10(e)-(f), respectively. These figures show the measured (solid line) and the reference angles (dashed line) of the two joints interconnecting modules 4, 5, and 6, namely joints 4 and 5. In accordance with the environment adaptation strategy in (13.29), the measured contact forces on modules 4, 5, and 6 affect the reference angles of joints 4 and 5. To illustrate this effect, we have plotted the environment adaptation components of the reference angles, i.e. $\phi_{\text{adapt},4}$ and $\phi_{\text{adapt},5}$, with a dotted line together with the reference angles. It was clearly observed during the experiments that the adaptation component of the joint reference angles serves as a 'curvature generator' at contacted points along the body of the snake robot. In other words, whenever the snake robot makes contact with an obstacle, the environment adaptation component in (13.29) produces more body curvature at this location. This curvature is subsequently propagated backwards by the body wave component in (13.28), which generates a push against the contacted obstacle.

It is interesting to compare the experimental results of this section with the experimental results from the jam resolution controller in Section 12.7. In particular, the forward velocity of the snake robot in the experimental results from this section is higher than in the experimental results from Section 12.7, which can be verified by comparing the plots of the forward position of the robot during the experiments. This difference in forward velocity is not surprising since the jam resolution controller from Chapter 12 spends much time resolving jams that occur during the motion. With the control strategy proposed in this chapter, jams are resolved continuously since there is a continuous adaptation of the body shape to the environment, which is clearly a more efficient strategy than employing an explicit jam resolution scheme. Moreover, the control strategy proposed in this chapter is a lot easier to implement than the jam resolution controller from Chapter 12.

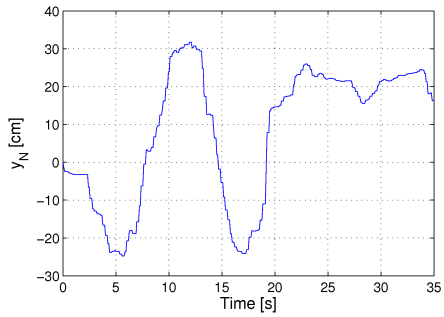
13.6 Chapter Summary

This chapter is summarized as follows:

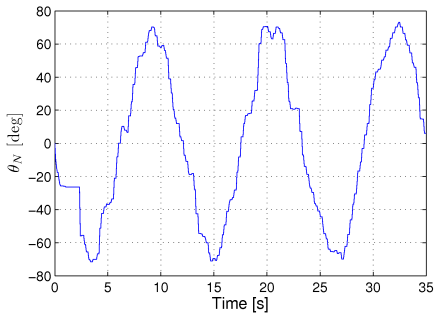
- We have proposed a general framework in (13.1) for motion control of snake robots, where the motion is specified in terms of a *body wave* component, an *environment adaptation* component, and a *heading control* component.
- We have employed the controller framework to propose a continuous control strategy, defined by (13.26) and (13.27), for straight line path following control of snake robots in unstructured environments.
- We have fitted the path following controller within the waypoint guidance strategy from Chapter 8 (see Algorithm 13.1).
- We have presented simulation results where the path following controller, in combination with the waypoint guidance strategy, was seen to successfully steer the snake robot between waypoints in an obstacle environment.
- We have presented experimental results where the snake robot Kulko was successfully propelled through three different obstacle environments when the joints were controlled according to (13.27).



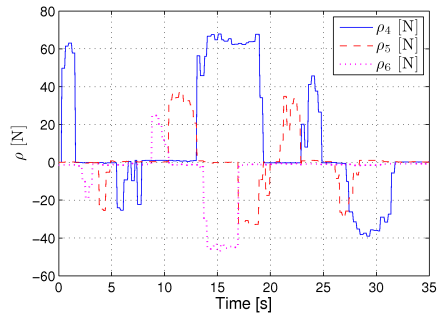
(a) The x coordinate of the head.



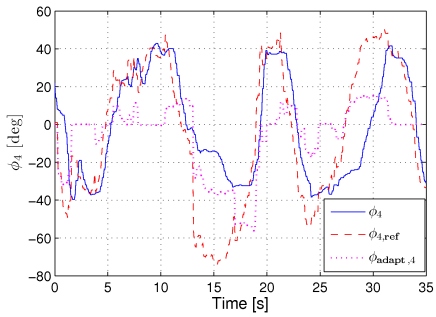
(b) The y coordinate of the head.



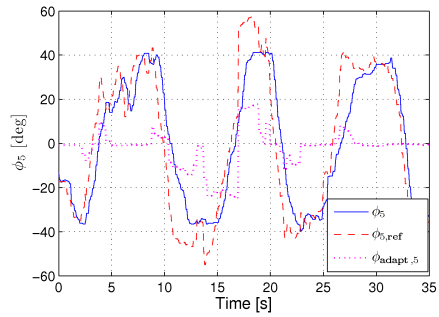
(c) The angle of the head link.



(d) Contact forces.



(e) The angle of joint 4.



(f) The angle of joint 5.

Figure 13.6: Experimental results of obstacle-aided locomotion in the first obstacle environment.

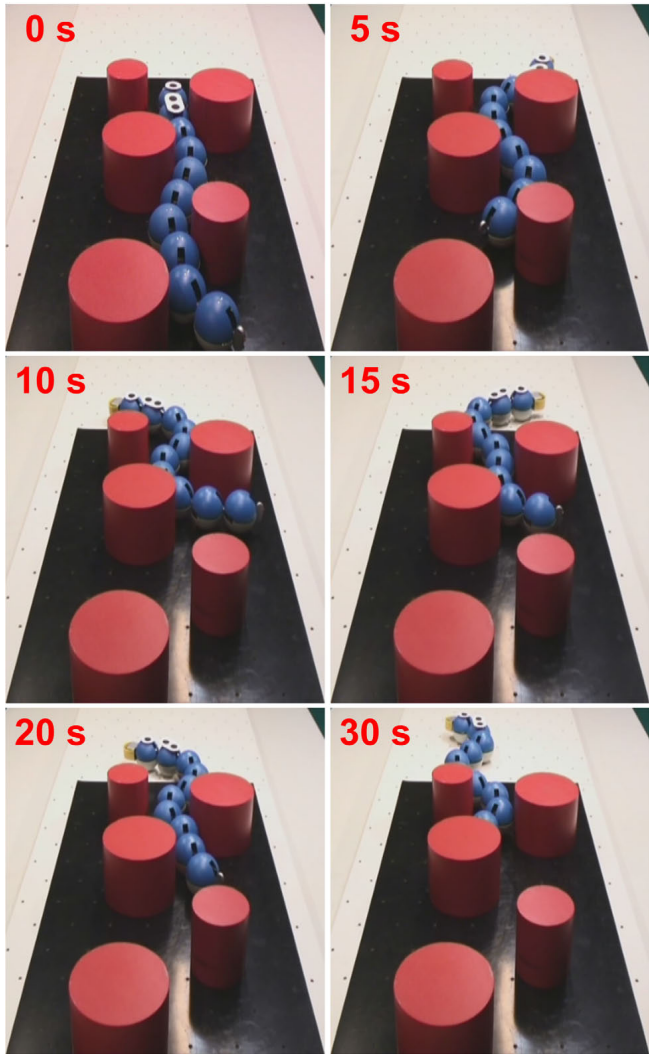
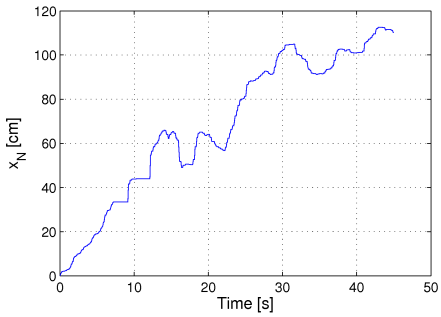
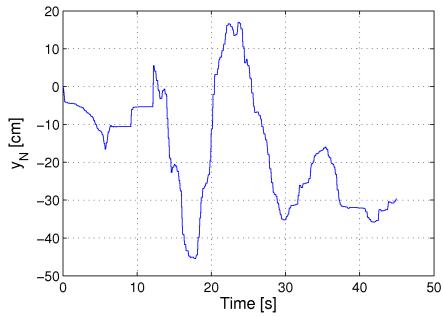


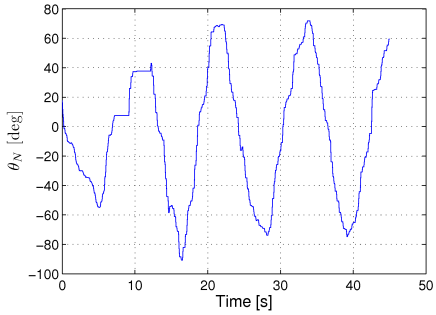
Figure 13.7: The motion of the snake robot in the first obstacle environment.



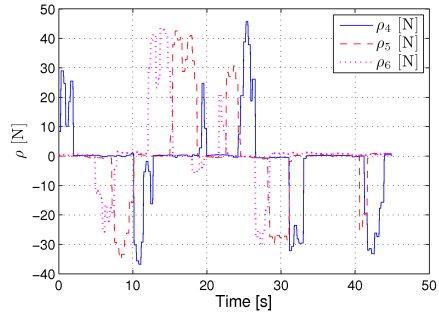
(a) The x coordinate of the head.



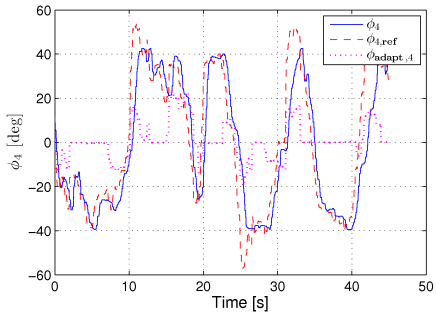
(b) The y coordinate of the head.



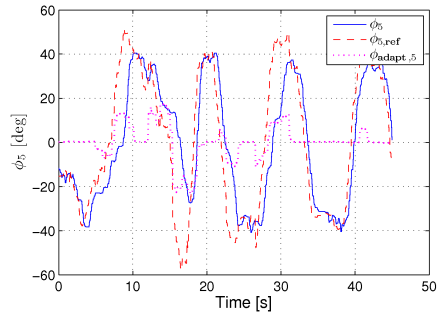
(c) The angle of the head link.



(d) Contact forces.



(e) The angle of joint 4.



(f) The angle of joint 5.

Figure 13.8: Experimental results of obstacle-aided locomotion in the second obstacle environment.

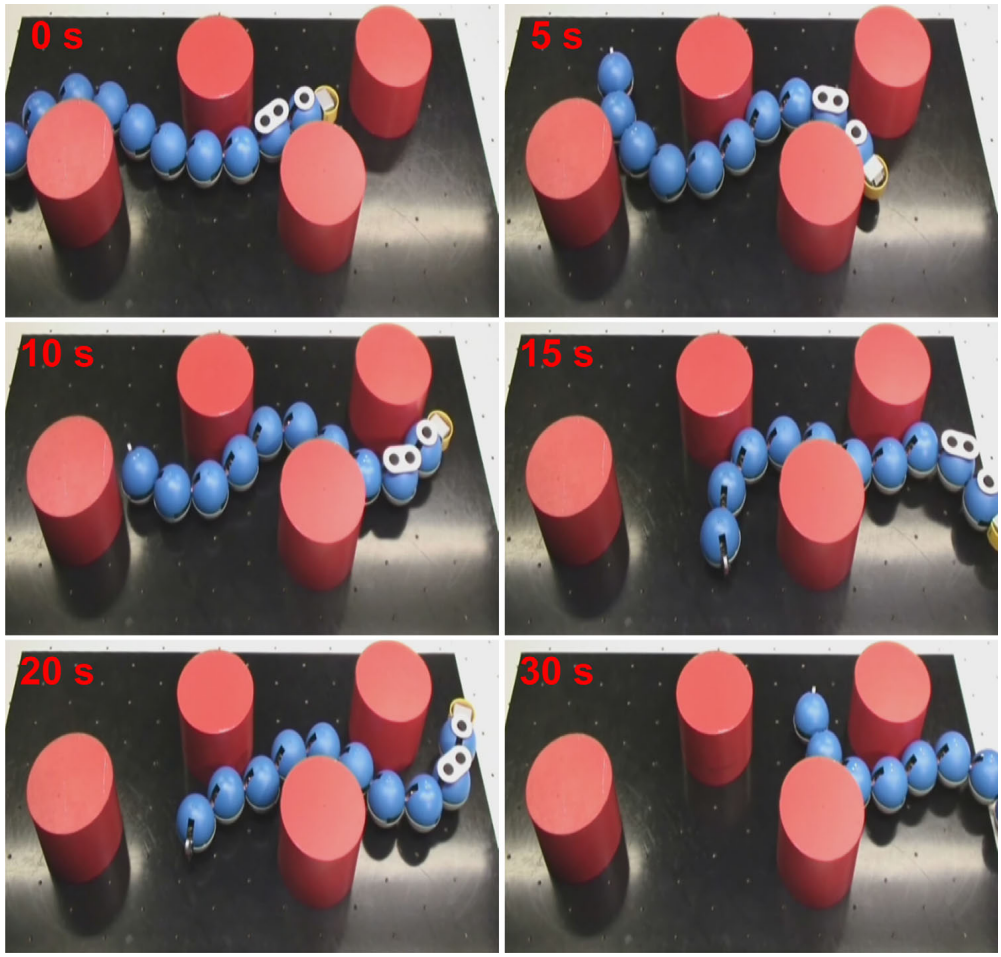
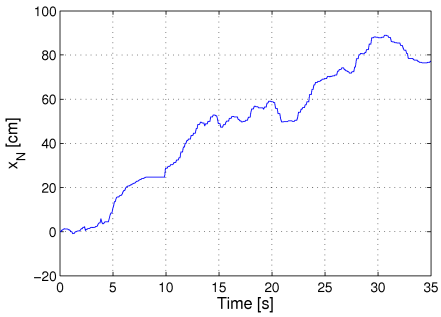
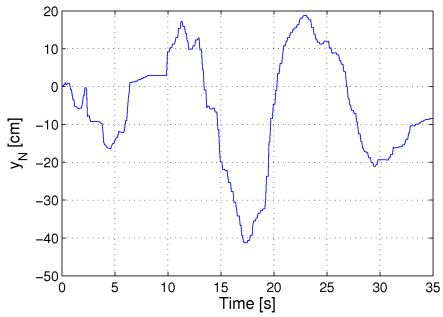


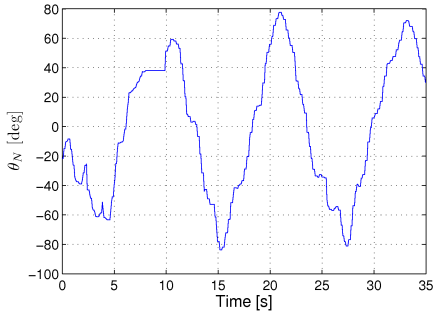
Figure 13.9: The motion of the snake robot in the second obstacle environment.



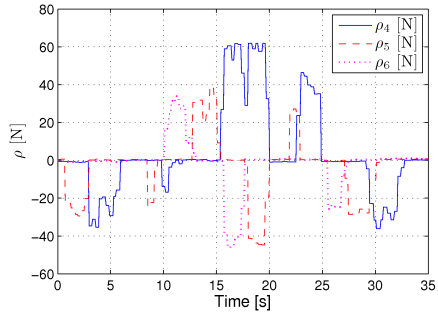
(a) The x coordinate of the head.



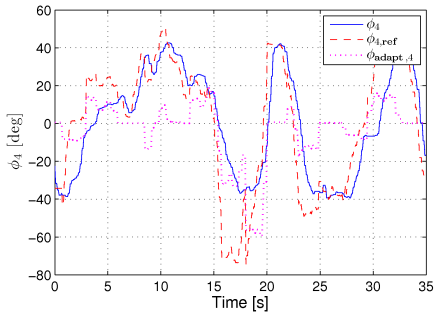
(b) The y coordinate of the head.



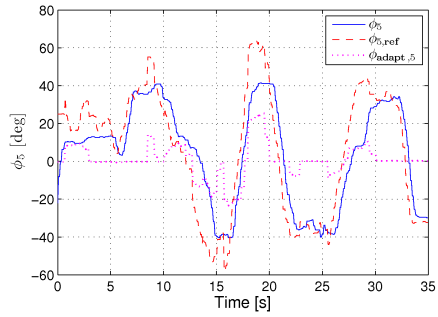
(c) The angle of the head link.



(d) Contact forces.



(e) The angle of joint 4.



(f) The angle of joint 5.

Figure 13.10: Experimental results of obstacle-aided locomotion in the third obstacle environment.

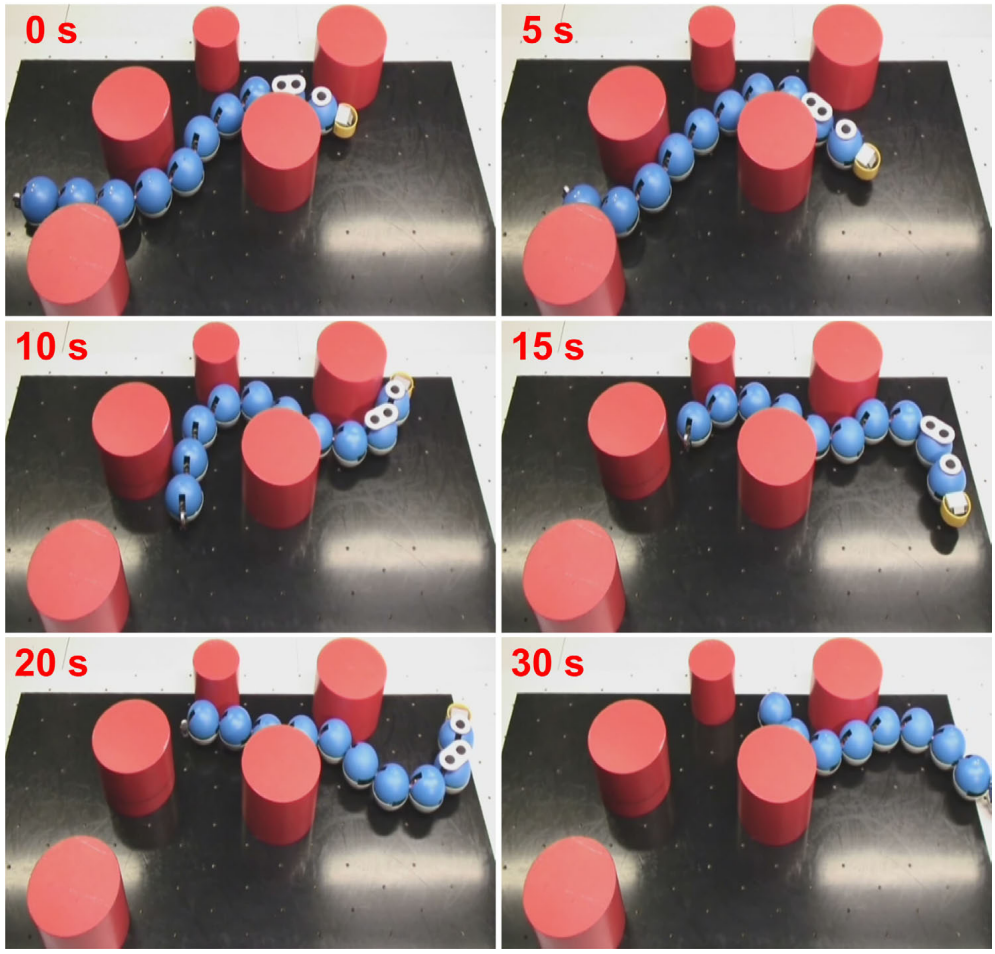
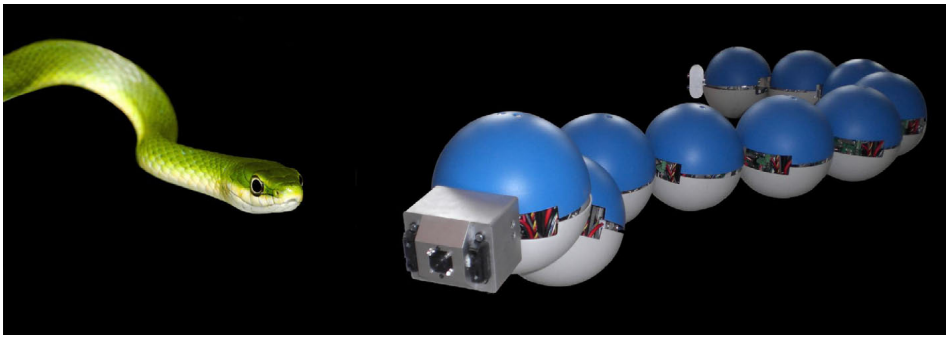


Figure 13.11: The motion of the snake robot in the third obstacle environment.

Chapter 14

Conclusions and Future Challenges



14.1 Conclusions of Part I - Snake Robot Locomotion on Planar Surfaces

The scope and objective of Part I

The objective of Part I of this thesis has been to increase our understanding of snake robot locomotion across horizontal and flat surfaces through analytical investigations of the equations of motion. We considered purely planar motion based on the hypothesis that the essential control principles of snake robot locomotion are contained in a planar perspective. In contrast to a significant part of previous research, which considers snake robots with nonholonomic constraints that prevent sideways motion of the links, this thesis has considered snake robots where the links are allowed to slip

sideways. We consider results based on such conditions to be more relevant to operations in unknown and cluttered environments, which represent the long-term goal of our research.

In the following, we summarize the contributions and conclusions from the individual chapters of Part I in relation to the scope and objective presented above.

Conclusions of Chapter 2

We presented a mathematical model of a planar snake robot with *anisotropic* ground friction properties moving across a flat and horizontal surface. A change of coordinates enabled us to partition the model into an *actuated* and an *unactuated* part, and a subsequent input transformation enabled us to linearize the actuated part of the snake robot dynamics. The partially feedback linearized model is more suitable for control design and analysis than the more complex non-linearized model of the snake robot.

Conclusions of Chapter 3

We presented the design of the snake robot *Wheeko*, which was developed for motion control experiments on flat surfaces. The robot consists of 10 identical modules with a novel joint design, and with anisotropic ground friction properties by use of passive wheels.

Conclusions of Chapter 4

We employed nonlinear system analysis tools for investigating fundamental properties of snake robot dynamics. In particular, we presented a stabilizability analysis that proves that any asymptotically stabilizing control law for a planar snake robot to an equilibrium point must be *time-varying*, i.e. not of pure-state feedback type. Furthermore, we presented a controllability analysis of planar snake robots influenced by viscous ground friction forces that shows that a snake robot is *not* controllable when the ground friction is *isotropic*, but that a snake robot becomes *strongly accessible* when the ground friction is *anisotropic*. The analysis also shows that the snake robot does *not* satisfy sufficient conditions for *small-time local controllability*. To our best knowledge, no formal controllability analysis has previously been reported for the position and link angles of a wheel-less snake robot influenced by ground friction. The results from the controllability analysis were not sufficient to conclude that a snake robot with anisotropic ground friction is controllable. However, the analysis proves that propulsion of

a snake robot under viscous friction conditions requires the friction to be *anisotropic*, and also that the joint angles of a snake robot should be *out of phase* during snake locomotion. These claims have been assumed in the snake robot literature, but have never before been formally proven.

This chapter also presented a simple relationship between link velocities *normal* to the direction of motion and *propulsive forces* in the direction of motion that explains how snake robots influenced by anisotropic ground friction are able to locomote forward on a planar surface. In our opinion, previously published research on snake robots has not presented an explicit mathematical description that *easily* explains how a snake robot achieves forward propulsion.

Finally, this chapter presented mathematical arguments that support the empirically derived description of *lateral undulation*, which is a gait pattern previously proposed in the snake robot literature based on observations of biological snakes. Moreover, we explained how offsetting the joint angles during lateral undulation enables directional control of the motion, and pointed out the observation that the body shape changes during lateral undulation mainly consist of relative displacements of the links *normal* to the forward direction of motion, while the relative displacements of the links along the forward direction are approximately constant.

Conclusions of Chapter 5

We proposed a control law that enables snake robots to track a planar straight path, and we analysed the stability of this path following controller by use of a *Poincaré map*. In particular, we showed that all state variables of the snake robot, except for the position along the path, trace out an *exponentially stable* periodic orbit during path following with the proposed controller. To our best knowledge, a *Poincaré map* has never before been used to study the stability properties of snake robot locomotion.

Conclusions of Chapter 6

We presented a simplified model of planar snake robot locomotion, which is intended for control design and stability analysis purposes. Moreover, we provided support of the claim that the simplified model captures the essential part of the dynamics of planar snake robot locomotion. In particular, we showed that the stabilizability and controllability properties of the simplified model are similar to the corresponding properties of the more complex model of the snake robot.

Conclusions of Chapter 7

We employed *averaging theory* to study the average effect of the joint motion that propels the robot during lateral undulation. In particular, we developed an *averaged* model of the velocity dynamics of a snake robot during lateral undulation, and we showed that the average velocity during lateral undulation converges exponentially fast to a steady state velocity. An analytical expression for calculating the steady state velocity as a function of the gait pattern parameters was presented. To our best knowledge, this is the first formal proof that a snake robot with anisotropic ground friction properties achieves forward propulsion when it moves by lateral undulation.

This chapter also derived fundamental relationships between the gait pattern parameters of lateral undulation and the resulting forward velocity of a planar snake robot. In particular, the derived properties state that the average forward velocity of a snake robot 1) is proportional to the squared amplitude of the sinusoidal motion of each joint, 2) is proportional to the angular frequency of the sinusoidal motion of each joint, 3) is proportional to a particular function of the constant phase shift between the joints, and 4) is maximized by the phase shift between the joints that also maximizes the particular phase shift function. To our best knowledge, these fundamental properties of snake locomotion have never before been derived analytically. The chapter presented experimental results based on the snake robot Wheeko that supported the derived properties.

Conclusions of Chapter 8

We proposed a straight line path following controller, and employed cascaded systems theory to prove that the controller $\mathcal{K}$-exponentially stabilizes a snake robot to any desired straight path. The proof relies on the assumption that the forward velocity of the robot is contained in some nonzero and positive interval. Specifying the bounds of this interval as a function of the gait pattern parameters remains a topic of future work. To our best knowledge, this is the first time the stability properties of a path following controller for a snake robot without nonholonomic constraints are formally proved. The performance of the path following controller was investigated through experiments with the snake robot Wheeko, where the proposed controller was shown to successfully steer the snake robot towards and along the desired straight path.

This chapter also described how the straight line path following con-

troller can be extended to path following of general curved paths by employing an approach previously proposed in the marine control literature for path following control of marine vessels.

Finally, this chapter proposed a waypoint guidance strategy for steering a snake robot along a path defined by waypoints interconnected by straight lines. It was proven that the waypoint guidance strategy is guaranteed to steer the position of the snake robot into the acceptance region of each waypoint. To our best knowledge, waypoint guidance has not previously been considered for motion control of snake robots.

14.2 Conclusions of Part II - Snake Robot Locomotion in Unstructured Environments

The scope and objective of Part II

The objective of Part II of this thesis has been to develop new control strategies for snake robot locomotion in environments containing external objects (or obstacles), which is in line with practical applications of these mechanisms. The underlying hypothesis has been that intelligent and efficient snake robot locomotion in unknown and unstructured environments requires that the snake robot can *sense* its environment and *adapt* its body shape and movements accordingly. As a result, we have focused on control strategies for *obstacle-aided locomotion*, where the snake robot is propelled by active use of the interaction with obstacles in its environment. Previous research on obstacle-aided locomotion is very limited. However, we believe control strategies for snake robots that consider environment interaction are important since the main advantage of these mechanisms are their potential ability to move in unstructured environments.

In the following, we summarize the contributions and conclusions from the individual chapters of Part II in relation to the scope and objective presented above.

Conclusions of Chapter 10

We proposed a *hybrid model* of a planar snake robot interacting with obstacles in its environment. Obstacle interaction was modelled by introducing a *unilateral velocity constraint* on each contacted link of the snake robot, which is a novel approach. In particular, the conventional approach for modelling the obstacle interaction would be to assume that the obstacle constraint force points in the normal direction of the *obstacle*. With the

approach taken in this thesis, the shape of the obstacles does not have to be considered explicitly as we instead calculate constraint forces with respect to the normal direction of the *contacted links*, which simplifies the equations of motion.

The equations governing the obstacle contact forces were formulated as a *linear complementarity problem* (LCP). This formulation enabled us to determine the existence and uniqueness properties of solutions to the hybrid system equations by employing existing general results concerning existence and uniqueness of solutions to LCPs.

Conclusions of Chapter 11

We presented the design of the snake robot *Kulko*, which was developed for the purpose of experiments related to obstacle-aided locomotion in unstructured environments. Previous snake robot design efforts have given very limited attention to the exterior gliding surface of such robots, and to methods for enabling snake robots to sense their environment. The design of Kulko is therefore novel since it combines a smooth outer surface (independently of how the joints are flexed) with contact force sensing capabilities. Experimental results were presented that validated the function of the contact force measurement system. To our best knowledge, this is the first reported snake robot that can measure the magnitude of external forces applied along its body. We also proposed an alternative approach for sensing environment contact forces, which has the advantage that force measurements are only required at the locations of the joints, and that the sensor system can be well protected inside the snake robot.

Conclusions of Chapter 12

We proposed a general control principle for snake robots which suggests that obstacle-aided locomotion is achieved by producing body shape changes where the links in contact with obstacles are rotated to increase the propulsive forces on the robot. The control principle was used to propose a *hybrid controller* for obstacle-aided locomotion aimed at resolving situations where the snake robot is jammed between obstacles. The concept of detecting and resolving snake robot jams has, to our best knowledge, not been treated in previous literature, but is a genuine challenge during snake robot locomotion in cluttered environments. Moreover, this is the first published control strategy for a snake robot involving feedback and explicit use of measured contact forces to achieve propulsion. We presented experimental results

based on the snake robot Kulko, where the hybrid controller was shown to maintain the propulsion of the snake robot in different obstacle environments. Also to our best knowledge, this is the first reported experiment where a snake robot is propelled forward based on measurements of the contact force amplitudes along the robot body.

Conclusions of Chapter 13

A general framework for motion control of snake robots was proposed, where the motion is specified in terms of a *body wave* component, an *environment adaptation* component, and a *heading control* component. Furthermore, we employed the controller framework to propose a continuous control strategy for straight line path following control of snake robots in unstructured environments. A significant contribution of the controller was the idea of a continuous jam resolution action that is performed in parallel with the cyclic wave motion of the robot to continuously adapt the body shape to the environment and prevent the motion from being jammed. A formal analysis of the performance of the path following controller remains a topic of future work. Experimental results were presented where the proposed controller successfully propelled the snake robot Kulko through three different obstacle environments.

14.3 Important Challenges of Future Research Efforts

Based on our research on snake robot locomotion so far, we end this thesis with an elaboration on what we consider to be the most significant research challenges that must be addressed before we will ever see useful snake robots outside the laboratory. We have presented in this thesis experimental results that demonstrate *planar* snake robot locomotion in unstructured environments based on environment sensing and body shape adaptation. However, to our best knowledge, the fact still remains that *non-planar* locomotion in unstructured environments based on environment sensing and body shape adaptation has *not* yet been demonstrated. Our primary claim is therefore that future applications of snake robots require significantly more research on adaptive behaviour during motion in unknown and cluttered environments.

14.3.1 Control Design Challenges

Analysable Mathematical Models

Future control design efforts for adaptive motion of snake robots should go beyond pure heuristics and instead base the controllers on analysable mathematical models and well established control design techniques. Model-based control design for snake robots is, however, a major challenge. As shown in this thesis, a mathematical model of the dynamics of a snake robot on a flat surface is very complex due to the many degrees of freedom of the robot. When contact forces from an unstructured environment are included, the model becomes even more complex because the discrete nature of the contact forces turns the model of the robot into a hybrid system. However, model-based control design can be achieved by pursuing simplified mathematical descriptions of the interaction between a snake robot and its environment that can be analysed from a control perspective. In particular, a simple relationship between body shape changes of a snake robot during environment contact and the resulting translational and rotational motion of the robot could enable an analytical derivation of the joint torques that will produce the desired motion. The controller development in this thesis based on the simplified model of snake robot locomotion is an example of how a simplified modelling approach can be employed to derive model-based control strategies for snake robots.

Feedback Control Laws based on Environment Sensing

Environment sensing is a requirement for efficient snake robot locomotion in unknown and unstructured environments. The challenge of utilizing this sensor information intelligently to maintain the propulsion of the robot is closely related to the challenge of developing analysable models of the robot. With a suitable description of how the environment interaction affects the motion, it is possible to analytically derive the control action that, in a given environment, will propel the robot in a desired direction. Control design for snake robots is also challenging because these mechanisms are generally underactuated, i.e. they have more degrees of freedom than actuators.

The approaches for adaptive motion control of snake robots considered in this thesis are *local* approaches since the body shape is adjusted locally at each contact point to increase the propulsive force without considering how the adjustment affects the remaining contact forces on the robot. More efficient control strategies are, however, likely to be obtained by considering *global* approaches to environment adaption, i.e. approaches where the

body shape adjustments are made by considering the overall interaction of the robot with its environment. In particular, situations where body shape adjustments act 'against' each other are likely to occur with a local approach, but are likely to be prevented by a global approach. Such global approaches represent a significant control design challenge.

Motion Planning Strategies

Enabling a snake robot to plan a route through an unstructured environment has, to our best knowledge, not been treated in previous literature. A snake robot can be equipped with vision capabilities by making use of sensor techniques already considered for more conventional mobile robots. However, due to their unique form of locomotion, motion planning for snake robots should be attacked with a somewhat different approach than motion planning for more conventional mobile robots. In particular, while *obstacle avoidance* is an important topic for wheeled, tracked and legged robots, the goal of snake robot locomotion is rather *obstacle utilization* since objects in the environment of a snake robot represent push points that the robot can use for propulsion.

14.3.2 Hardware Design Challenges

Environment Sensing

Measuring external contact forces on the snake robot is a natural approach for sensing the environment. The force sensing system of a snake robot is, however, particularly challenging since the robot is articulated, which introduces the challenge of preventing the joint motion from interfering with the measurements of the external forces. Measuring forces directly along the body of a snake robot is, in other words, a significant design challenge. The design of the snake robot *Kulko*, which was presented in this thesis, represents a proposed solution to this challenge. However, the robot cannot measure external forces very well at the connection point between two modules.

An alternative approach, which was proposed in this thesis, is to estimate the external forces acting on the robot solely through force measurements at each articulation point along the robot. The instrumentation system of this solution is significantly simpler than the instrumentation required to measure external forces on the robot directly. Further investigations of this approach represent an interesting and important topic of future work.

We consider research on environment sensing for snake robots to be highly relevant also to many other application areas within robotics, which suggests that researchers working with snake robots should identify and pursue synergies with other robotic research areas where environment adaptation is important.

Ground Friction Force Limitation

If the propulsion of the snake robot is based on forward gliding motion similar to the motion of biological snakes, then a sufficiently smooth exterior surface is very important since any irregularities along the body may potentially induce large obstructive friction forces on the robot. Obtaining a smooth surface combined with contact force sensing at articulated parts of the robot represents a significant design challenge. The snake robot Kulko presented in this thesis has a smooth surface due to spherical shells that cover each joint module, and can measure external forces due to force sensors mounted underneath the shells.

The friction forces opposing the motion of a snake robot can also be limited by introducing active propulsion along the body. This approach is employed by the snake robot with active tracks presented in Granosik *et al.* (2006) and by the skin drive mechanism described in McKenna *et al.* (2008). The drawback of active propulsion along the body of a snake robot is that the mechanical complexity of the robot is significantly increased. In our opinion, the ideal solution is a snake robot with a passive and smooth tactile skin that can glide forward like a biological snake. Mechanism simplicity is important to the future use of snake robots since this increases the reliability and reduces the development cost of the robots.

Robust and Strong Actuation Mechanisms

In order to move in unstructured environments, the snake robot must generally be able to lift parts of its body. This means that there is some lower limit to the ratio between the strength of the actuators and the weight of the robot. Developing joint mechanisms for snake robots where this ratio is maximized is an important design challenge that must be addressed. Furthermore, locomotion in unstructured environments generally requires that the actuators can work against environment contact forces over time without overheating. The modified servo motor for snake robots described in Wright *et al.* (2007) targets this design challenge. A compliant joint mechanism is advantageous during locomotion in unstructured environments,

which is the motivation for the use of pneumatic actuators in the snake robot with active tracks presented in Granosik *et al.* (2006). However, compliance can also be enforced by the controller of the robot if the contact forces along the body are measured.

Dustproofing and Waterproofing

In order to make use of snake robots outside the generally clean lab environments, the robots must be able to operate despite of mud and dirt in their environment. Water resistance is also generally a great advantage. Both dustproofing and waterproofing a snake robot is challenging, in particular when we also require force sensing capabilities and a smooth exterior surface.

Bibliography

Andruska, A. M. and K. S. Peterson (2008). Control of a snake-like robot in an elastically deformable channel. *IEEE/ASME Trans. Mechatronics* **13**(2), 219–227.

Bauchot, R. (1994). *Snakes: A Natural History*. Sterling Publishing Company.

Bayraktaroglu, Z. Y. (2008). Snake-like locomotion: Experimentations with a biologically inspired wheel-less snake robot. *Mechanism and Machine Theory* **44**(3), 591–602.

Bayraktaroglu, Z.Y. and P. Blazevic (2005). Understanding snakelike locomotion through a novel push-point approach. *J. Dyn. Syst. - Trans. ASME* **127**(1), 146–152.

Bernstein, D. S. (2009). *Matrix Mathematics: Theory, Facts, and Formulas (Second Edition)*. 2 ed.. Princeton University Press.

Bianchini, R. M. and G. Stefani (1990). Graded approximations and controllability along a trajectory. *SIAM J. Control and Optimization* **28**(4), 903–924.

Bloch, A. M., J. Baillieul, P. Crouch and J. Marsden (2003). *Nonholonomic Mechanics and Control*. Springer-Verlag.

Børhaug, E. (2008). Nonlinear Control and Synchronization of Mechanical Systems. PhD thesis. Norwegian University of Science and Technology.

Boyer, F., M. Porez and W. Khalil (2006). Macro-continuous computed torque algorithm for a three-dimensional eel-like robot. *IEEE Trans. Robot.* **22**(4), 763–775.

Brockett, R. (1983). Asymptotic stability and feedback stabilization. *Differential Geometric Control Theory* pp. 181–191.

Brogliato, B. (1999). *Nonsmooth Mechanics*. 2 ed.. Springer. London.

Brunete, A., E. Gambao, J. E. Torres and M. Hernando (2006). A 2 DoF servomotor-based module for pipe inspection modular micro-robots. In: *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. pp. 1329–1334.

Burdick, J. W., J. Radford and G.S. Chirikjian (1995). A sidewinding locomotion gait for hyper-redundant robots. *Advanced Robotics* **9**(3), 195–216.

Chen, Li, Yuechao Wang, Bin Li, Shugen Ma and Dengping Duan (2007). *Study on Locomotion of a Crawling Robot for Adaptation to the Environment*. Chap. 18, pp. 301–316. I-Tech Education and Publishing.

Chen, Li, Yuechao Wang, Shugen Ma and Bin Li; (2004). Studies on lateral rolling locomotion of a snake robot. In: *Proc. IEEE Int. Conf. Robotics and Automation*. pp. 5070–5074.

Chen, T. L. T., S. Liu and J. Yen (2008). A bio-mimetic snake-like robot: Sensor based gait control. In: *Advanced robotics and Its Social Impacts, 2008. ARSO 2008. IEEE Workshop on*. pp. 1–6.

Chernousko, F. (2003). Snake-like locomotions of multilink mechanisms. *J. Vib. Cont.* **9**(1-2), 235–256.

Chernousko, F. (2005). Modelling of snake-like locomotion. *Appl. Math. Comput.* **164**(2), 415–434.

Chirikjian, Gregory S. (1992). Theory and Applications of Hyper-Redundant Robotic Manipulators. PhD thesis. California Institute of Technology. Pasadena, California.

Chirikjian, G.S. and J.W. Burdick (1995). The kinematics of hyper-redundant robot locomotion. *IEEE Trans. Robot. Autom.* **11**(6), 781–793.

Coron, J.-M. and L. Rosier (1994). A relation between continuous time-varying and discontinuous feedback stabilization. *J. of Mathematical Systems, Estimation, and Control* **4**(1), 67–84.

Cottle, R. W., J. S. Pang and R. E. Stone (1992). *The linear complementarity problem*. Academic Press.

Crespi, A. and A. J. Ijspeert (2008). Online optimization of swimming and crawling in an amphibious snake robot. *IEEE Trans. Robotics* **24**(1), 75–87.

Date, H. and Y. Takita (2005). Control of 3d snake-like locomotive mechanism based on continuum modeling. In: *Proc. ASME 2005 International Design Engineering Technical Conferences*. no. DETC2005-85130.

Date, H., M. Sampei and S. Nakaura (2001*a*). Control of a snake robot in consideration of constraint force. In: *Proc. IEEE Int. Conf. on Control Applications*. pp. 966–971.

Date, H., Y. Hoshi and M. Sampei (2000). Locomotion control of a snake-like robot based on dynamic manipulability. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*.

Date, H., Y. Hoshi, M. Sampei and N. Shigeki (2001*b*). Locomotion control of a snake robot with constraint force attenuation. In: *Proc. American Control Conference*. pp. 113–118.

Date, Hisashi and Yoshihiro Takita (2007). Adaptive locomotion of a snake like robot based on curvature derivatives. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. San Diego, CA, USA. pp. 3554–3559.

Do, K.D. and J. Pan (2003). Global tracking control of underactuated ships with off-diagonal terms. In: *IEEE Conf. Decision and Control*. Vol. 2. pp. 1250–1255.

Dowling, K. (1999). Limbless locomotion: learning to crawl. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Vol. 4.

Dowling, Kevin J. (1997). Limbless Locomotion. Learning to Crawl with a Snake Robot. PhD thesis. Carnegie Mellon University.

Egeland, Olav and Jan Tommy Gravdahl (2002). *Modeling and Simulation for Automatic Control*. Marine Cybernetics. Trondheim, Norway.

Endo, G., K. Togawa and S. Hirose (1999). Study on self-contained and terrain adaptive active cord mechanism. In: *Proceedings IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. Vol. 3. pp. 1399–1405.

EUROP/CARE (2009). Robotic visions to 2020 and beyond - the strategic research agenda for robotics in Europe. Technical report.

Fjerdingen, S. A., J. R. Mathiassen, H. Schumann-Olsen and E. Kyrkjebø (2008). Adaptive snake robot locomotion: A benchmarking facility for experiments. In: *European Robotics Symposium 2008*. Vol. 44. pp. 13 – 22.

Fossen, Thor I. (2002). *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*. Marine Cybernetics. Trondheim, Norway.

Fredriksen, E. and K. Y. Pettersen (2006). Global κ-exponential way-point maneuvering of ships: Theory and experiments. *Automatica* **42**, 677 – 687.

Gao, J., X. Gao, W. Zhu, J. Zhu and B. Wei (2008). Design and research of a new structure rescue snake robot with all body drive system. In: *IEEE Int. Conf. Mechatronics and Automation*. pp. 119–124.

Goebel, R., R. Sanfelice and A. Teel (2009). Hybrid dynamical systems. *IEEE Contr. Syst. Magazine* **29**(2), 28–93.

Goldstein, H., C. Poole and J. Safko (2002). *Classical Mechanics - Third Edition*. Addision Wesley.

Gonzalez-Gomez, J., Houxiang Zhang and Eduardo Boemo (2007). *Locomotion Principles of 1D Topology Pitch and Pitch-Yaw-Connecting Modular Robots*. Chap. 24, pp. 403–428. Advanced Robotics Systems International and I-Tech Education and Publishing.

Gonzalez-Gomez, J., J. Gonzalez-Quijano, H. Zhang and M. Abderrahim (2010). Toward the sense of touch in snake modular robots for search and rescue operations. In: *Proc. ICRA 2010 Workshop "Modular Robots: State of the Art"*. pp. 63–68.

Grabec, Igor (2002). Control of a creeping snake-like robot. In: *Proc. 7th Int. Workshop on Advanced Motion Control*. pp. 526–513.

Granosik, Grzegorz, Johann Borenstein and Malik G. Hansen (2006). *Industrial Robotics: Programming, Simulation and Applications*. Chap. 33, pp. 633–662. Pro Literatur Verlag, Germany / ARS, Austria.

Gray, J. (1946). The mechanism of locomotion in snakes. *J. Exp. Biol.* **23**(2), 101–120.

Greenfield, A., A. A. Rizzi and H. Choset (2005). Dynamic ambiguities in frictional rigid-body systems with application to climbing via bracing. In: *Proc. IEEE Int. Conf. Robotics and Automation*. pp. 1947–1952.

Gu, Y.-L. and Y. Xu (1993). A normal form augmentation approach to adaptive control of space robot systems. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Vol. 2. pp. 731–737.

Hara, M., S. Satomura, H. Fukushima, T. Kamegawa, H. Igarashi and F. Matsuno (2007). Control of a snake-like robot using the screw drive mechanism. In: *IEEE Int. Conf. Robotics and Automation*. pp. 3883–3888.

Harville, D. A. (2000). *Matrix Algebra From a Statistician's Perspective*. Springer.

Hatton, R. L. and H. Choset (2010). Sidewinding on slopes. In: *IEEE Int. Conf. Robotics and Automation*. pp. 691–696.

Hatton, R.L. and H. Choset (2009*a*). Generating gaits for snake robots by annealed chain fitting and keyframe wave extraction. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. pp. 840–845.

Hatton, R.ăL. and H. Choset (2009*b*). Approximating displacement with the body velocity integral. In: *Proc. Robotics: Science and Systems*.

Hicks, Gregory and Kazufumi Ito (2005). A method for determination of optimal gaits with application to a snake-like serial-link structure. *IEEE Trans. Autom. Contr.* **50**(9), 1291–1306.

Hicks, Gregory P. (2003). Modeling and Control of a Snake-like Serial-link Structure. PhD thesis. North Carolina State University.

Hirose, S. (1993). *Biologically Inspired Robots: Snake-Like Locomotors and Manipulators*. Oxford University Press. Oxford.

Hu, D.ăL., J. Nirody, T. Scott and M.ăJ. Shelley (2009). The mechanics of slithering locomotion. In: *Proc. National Academy of Sciences, USA*. Vol. 106. p. 10081 Ű 10085.

Ishikawa, M. (2009). Iterative feedback control of snake-like robot based on principal fiber bundle modeling. *Int. J. Advanced Mechatronic Systems* **1**(3), 175–182.

Ishikaway, M., K. Owaki, M. Shinagawa and T. Sugie (2010). Control of snake-like robot based on nonlinear controllability analysis. In: *IEEE Int. Conf. Control Applications*. pp. 1134–1139.

Kamegawa, T., T. Harada and A. Gofuku (2009). Realization of cylinder climbing locomotion with helical form by a snake robot with passive wheels. In: *Proc. IEEE Int. Conf. Robotics and Automation*. pp. 3067–3072.

Kamegawa, T., T. Yarnasaki, H. Igarashi and F. Matsuno (2004). Development of the snake-like rescue robot 'Kohga'. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Vol. 5. pp. 5081–5086.

Kane, T.R. and D.A. Lecison (2000). Locomotion of snakes: A mechanical 'explanation'. *Int. J. Solids Struct.* **37**(41), 5829–5837.

Kanso, E., J.E. Marsden, C. W. Rowley and J. B. Melli-Huber (2005). Locomotion of articulated bodies in a perfect fluid. *J. Nonlinear Sci.* **15**, 255Ű289.

Kelly, S.D. and Richard M. Murray (1995). Geometric phases and robotic locomotion. *J. Robotic Systems* **12**(6), 417–431.

Khalil, Hassan K. (2002). *Nonlinear Systems*. 3rd ed.. Prentice Hall.

Kimura, H. and S. Hirose (2002). Development of genbu : Active wheel passive joint articulated mobile robot. In: *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. Vol. 1. pp. 823–828.

Krishnaprasad, P.S. and D.P. Tsakiris (1994). G-snakes: Nonholonomic kinematic chains on lie groups. In: *Proc. 33rd IEEE Conf. Decision and Control*. Vol. 3. Lake Buena Vista, FL USA. pp. 2955–2960.

Kulali, GM, M. Gevher, AM Erkmen and I. Erkmen (2002). Intelligent gait synthesizer for serpentine robots. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Vol. 2.

Kuwada, A., S. Wakimoto, K. Suzumori and Y. Adomi (2008). Automatic pipe negotiation control for snake-like robot. In: *Proc. IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*. pp. 558–563.

Lefeber, E. (2000). Tracking Control of Nonlinear Mechanical Systems. PhD thesis. University of Twente. Dept. of Applied Mathematics. The Netherlands.

Li, J. and J. Shan (2008). Passivity control of underactuated snake-like robots. In: *Proc. 7th World Congress on Intelligent Control and Automation*. pp. 485–490.

Liljebäck, P. and K. Y. Pettersen (2011). Waypoint guidance control of snake robots. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Shanghai, China. Accepted.

Liljebäck, P., I. U. Haugstuen and K. Y. Pettersen (2010*a*). Experimental investigation of a path following controller for planar snake robots. In: *Proc. IEEE Int. Conf. Control, Automation, Robotics, and Vision (ICARCV)*. Singapore. pp. 2325–2332.

Liljebäck, P., I. U. Haugstuen and K. Y. Pettersen (2010*b*). Path following control of planar snake robots using a cascaded approach. In: *Proc. IEEE Conf. Decision and Control*. Atlanta, GA, USA. pp. 1969–1976.

Liljebäck, P., I. U. Haugstuen and K. Y. Pettersen (2011*a*). Path following control of planar snake robots using a cascaded approach. *IEEE Trans. Control Systems Technology*. To appear.

Liljebäck, P., K. Y. Pettersen and Ø. Stavdahl (2009*a*). Modelling and control of obstacle-aided snake robot locomotion based on jam resolution. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Kobe, Japan. pp. 3807–3814.

Liljebäck, P., K. Y. Pettersen and Ø. Stavdahl (2010*c*). A snake robot with a contact force measurement system for obstacle-aided locomotion. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Anchorage, AK, USA. pp. 683–690.

Liljebäck, P., K. Y. Pettersen, Ø. Stavdahl and J. T. Gravdahl (2009*b*). Controllability analysis of planar snake robots influenced by viscous ground friction. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. St. Louis, MO, USA. pp. 3615–3622.

Liljebäck, P., K. Y. Pettersen, Ø. Stavdahl and J. T. Gravdahl (2009*c*). Stability analysis of snake robot locomotion based on Poincaré maps. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. St. Louis, MO, USA. pp. 3623–3630.

Liljebäck, P., K. Y. Pettersen, Ø. Stavdahl and J. T. Gravdahl (2010*d*). Experimental investigation of fundamental properties of snake robot locomotion. In: *Proc. IEEE Int. Conf. Control, Automation, Robotics, and*

Vision (ICARCV). Singapore. pp. 187–194. Finalist for the Best Paper Award.

Liljebäck, P., K. Y. Pettersen, Ø. Stavdahl and J. T. Gravdahl (2010*e*). Experimental investigation of obstacle-aided locomotion with a snake robot. *IEEE Trans. Robotics*. Conditionally accepted.

Liljebäck, P., K. Y. Pettersen, Ø. Stavdahl and J. T. Gravdahl (2010*f*). Fundamental properties of snake robot locomotion. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. Taipei, Taiwan. pp. 2876–2883.

Liljebäck, P., K. Y. Pettersen, Ø. Stavdahl and J. T. Gravdahl (2010*g*). A hybrid model of obstacle-aided snake robot locomotion. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Anchorage, AK, USA. pp. 675–682.

Liljebäck, P., K. Y. Pettersen, Ø. Stavdahl and J. T. Gravdahl (2010*h*). Hybrid modelling and control of obstacle-aided snake robot locomotion. *IEEE Trans. Robotics* **26**(5), 781–799.

Liljebäck, P., K. Y. Pettersen, Ø. Stavdahl and J. T. Gravdahl (2010*i*). A review on modelling, implementation, and control of snake robots. *Robotics and Autonomous Systems*. Submitted.

Liljebäck, P., K. Y. Pettersen, Ø. Stavdahl and J. T. Gravdahl (2010*j*). A simplified model of planar snake robot locomotion. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. Taipei, Taiwan. pp. 2868–2875.

Liljebäck, P., K. Y. Pettersen, Ø. Stavdahl and J. T. Gravdahl (2010*k*). Snake robot locomotion in environments with obstacles. *IEEE Trans. Mechatronics*. Submitted.

Liljebäck, P., K. Y. Pettersen, Ø. Stavdahl and J. T. Gravdahl (2010*l*). Stability analysis of snake robot locomotion based on averaging theory. In: *Proc. IEEE Conf. Decision and Control*. Atlanta, GA, USA. pp. 1977–1984.

Liljebäck, P., K. Y. Pettersen, Ø. Stavdahl and J. T. Gravdahl (2011*b*). Controllability and stability analysis of planar snake robot locomotion. *IEEE Trans. Automatic Control*. To appear.

Liljebäck, P., K. Y. Pettersen, Ø. Stavdahl and J. T. Gravdahl (2011*c*). Path following control of snake robots in unstructured environments. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Shanghai, China. Accepted.

Liljebäck, P., K. Y. Pettersen, Ø. Stavdahl and J. T. Gravdahl (2011*d*). A simplified model and fundamental properties of snake robot locomotion. *Robotica*. To be submitted.

Liljebäck, P., Ø. Stavdahl, K. Y. Pettersen and J. T. Gravdahl (2011*e*). Two new design concepts for snake robot locomotion in unstructured environments. *Paladyn. Journal of Behavioral Robotics*. To appear.

Liljebäck, P., S. Fjerdingen, K. Y. Pettersen and Ø. Stavdahl (2009*d*). A snake robot joint mechanism with a contact force measurement system. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Kobe, Japan. pp. 3815–3820.

Liljebäck, Pål, Øyvind Stavdahl and Anders Beitnes (2006). SnakeFighter - development of a water hydraulic fire fighting snake robot. In: *Proc. IEEE Int. Conf. Control, Automation, Robotics, and Vision (ICARCV)*. Singapore.

Linnemann, R., K. L. Paap, B. Klaassen and J. Vollmer (1999). Motion control of a snakelike robot. In: *Proc. Third European Workshop on Advanced Mobile Robots*. pp. 1–8.

Lipkin, Kevin, Isaac Brown, Howie Choset, Justine Rembisz, Philip Gianfortoni and Allison Naaktgeboren (2007). Differentiable and piecewise differentiable gaits for snake robots. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. San Diego, CA, USA. pp. 1864–1869.

Lochmatter, T., P. Roduit, C. Cianci, N. Correll, J. Jacot and A. Martinoli (2008). Swistrack - a flexible open source tracking software for multi-agent systems. In: *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. pp. 4004–4010.

Lötstedt, P. (1981). Coulomb friction in two-dimensional rigid body systems. *Zeitschrift für Angewandte Mathematik und Mechanik* **61**, 605–615.

Lötstedt, P. (1982). Mechanical systems of rigid bodies subject to unilateral constraints. *SIAM J. Appl. Math.* **42**(2), 281–296.

Ma, Shugen (1999). Analysis of snake movement forms for realization of snake-like robots. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Vol. 4. Detroit, MI USA. pp. 3007–3013.

Ma, Shugen (2001). Analysis of creeping locomotion of a snake-like robot. *Adv. Robotics* **15**(2), 205–224.

Ma, Shugen and Naoki Tadokoro (2006). Analysis of creeping locomotion of a snake-like robot on a slope. *Autonomous Robots* **20**, 15–23.

Ma, Shugen, Hiroaki Araya and Li Li (2001). Development of a creeping snake-robot. In: *Proc. IEEE Int. Symp. Computational Intelligence in Robotics and Automation*. pp. 77–82.

Ma, Shugen, Y. Ohmameuda and K. Inoue (2004). Dynamic analysis of 3-dimensional snake robots. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. pp. 767–772.

Ma, Shugen, Yoshihiro Ohmameuda, Kousuke Inoue and Bin Li (2003). Control of a 3-dimensional snake-like robot. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Vol. 2. Taipei, Taiwan. pp. 2067 – 2072.

Masayuki, A., T. Takayama and S. Hirose (2004). Development of "Souryu-III": connected crawler vehicle for inspection inside narrow and winding spaces. In: *Proc. IEEE Int. Conf. Intelligent Robots and Systems*. Vol. 1. pp. 52–57.

Mason, M. T. and Y. Wang (1988). On the inconsistency of rigid-body frictional planar mechanics. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Vol. 1. pp. 524–528.

Matsuno, F. and H. Sato (2005). Trajectory tracking control of snake robots based on dynamic model. In: *Proc. IEEE Int. Conf. on Robotics and Automation*. pp. 3029–3034.

Matsuno, F. and K. Mogi (2000). Redundancy controllable system and control of snake robots based on kinematic model. In: *Proc. IEEE Int. Conf. Decision and Control*. Vol. 5. pp. 4791–4796 vol.5.

Matsuno, F. and K. Suenaga (2003). Control of redundant 3D snake robot based on kinematic model. In: *Proc. IEEE Int. Conf. Robotics and Automation*. pp. 2061–2066.

Mattison, Chris (2002). *The Encyclopaedia of Snakes*. Cassell Paperbacks. London.

McIsaac, K.A. and J.P. Ostrowski (2003*a*). Motion planning for anguilliform locomotion. *IEEE Trans. Rob. Aut.* **19**(4), 637–625.

McIsaac, Kenneth A. and James P. Ostrowski (2003*b*). A framework for steering dynamic robotic locomotion systems. *Int. J. Robot. Res.* **22**(2), 83–97.

McKenna, James C., David J. Anhalt, Frederick M. Bronson, H. Ben Brown, Michael Schwerin, Elie Shammas and Howie Choset (2008). Toroidal skin drive for snake robot locomotion. In: *Proc. IEEE Int. Conf. Robotics and Automation*. pp. 1150–1155.

Mehta, V., S. Brennan and F. Gandhi (2008). Experimentally verified optimal serpentine gait and hyperredundancy of a rigid-link snake robot. *IEEE Trans. on Robotics* **24**(2), 348–360.

Melli, J. B., C. W. Rowley and D. S. Rufat (2006). Motion planning for an articulated body in a perfect planar fluid. *SIAM J. Applied Dynamical Systems* **5**(4), 650Ű669.

Miller, Gavin (2002). *Neurotechnology for Biomimetic Robots*. Chap. Snake Robots for Search and Rescue, pp. 271–284. MIT Press Cambridge, MA, USA. Cambridge/London.

Moon, B.R. and C. Gans (1998). Kinematics, muscular activity and propulsion in gopher snakes. *Journal of Experimental Biology* **201**, 2669–2684.

Morgansen, K.A., B.I. Triplett and D.J. Klein (2007). Geometric methods for modeling and control of free-swimming fin-actuated underwater vehicles. *IEEE Trans. Robotics* **23**(6), 1184–1199.

Morgansen, K.A., P.A. Vela and J.W. Burdick (2002). Trajectory stabilization for a planar carangiform robot fish. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Vol. 1. pp. 756–762.

Morgansen, K.A., V. Duidam, R.J. Mason, J.W. Burdick and R.M. Murray (2001). Nonlinear control methods for planar carangiform robot fish locomotion. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Vol. 1. pp. 427–434 vol.1.

Mori, M. and S. Hirose (2002). Three-dimensional serpentine motion and lateral rolling by active cord mechanism ACM-R3. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. pp. 829–834.

Murugendran, B., A. A. Transeth and S. A. Fjerdingen (2009). Modeling and path-following for a snake robot with active wheels. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. pp. 3643 – 3650.

Nijmeijer, Henk and A. van der Schaft (1990). *Nonlinear Dynamical Control Systems*. Springer-Verlag. New York.

Nilsson, Martin (1997). Ripple and roll: Slip-free snake robot locomotion. In: *Proc. Mechatronical Computer Systems for Perception and Action*. Piza, Italy.

Nilsson, Martin (1998). Snake robot - free climbing. *IEEE Contr. Syst. Mag.* **18**(1), 21–26.

Nilsson, Martin (2004). Serpentine locomotion on surfaces with uniform friction. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. pp. 1751–1755.

Ohashi, T. Yamada, H. and S. Hirose (2010). Loop forming snake-like robot acm-r7 and its serpenoid oval control. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. pp. 413–418.

Ohno, H. and S. Hirose (2001). Design of slim slime robot and its gait of locomotion. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. Vol. 2. pp. 707–715.

Ostrowski, J. and J. Burdick (1998). The geometric mechanics of undulatory robotic locomotion. *Int. J. Robot. Res.* **17**(7), 683–701.

Ostrowski, James Patrick (1996). The Mechanics and Control of Undulatory Robotic Locomotion. PhD thesis. California Institute of Technology.

Paap, K.L., F. Kirchner and B. Klaassen (1999). Motion control scheme for a snake-like robot. In: *Proc. IEEE Int. Symp. Computational Intelligence in Robotics and Automation*. pp. 59–63.

Panteley, E. and A. Loria (1998). On global uniform asymptotic stability of nonlinear time-varying systems in cascade. *Systems & Control Letters* **33**(2), 131 – 138.

Panteley, E., E. Lefeber, A. Loria and H. Nijmeijer (1998). Exponential tracking control of a mobile car using a cascaded approach. In: *Proc. IFAC Workshop on Motion Control*. pp. 221 – 226.

Parker, T. S. and L. O. Chua (1989). *Practical Numerical Algorithms for Chaotic Systems*. Springer-Verlag.

Pavlov, A., E. Børhaug, E. Panteley and K. Y. Pettersen (2007). Straight line path following for formations of underactuated surface vessels. In: *Proc. IFAC NOLCOS*.

Pettersen, K. Y. and E. Lefeber (2001). Way-point tracking control of ships. In: *Proc. IEEE Conf. Decision and Control*. pp. 940–945.

Pettersen, K. Y. and O. Egeland (1996). Exponential stabilization of an underactuated surface vessel. In: *Proc. 35th IEEE conf. Decision and Control*. Vol. 1. pp. 967–972.

Poi, G., C. Scarabeo and B. Allotta (1998). Traveling wave locomotion hyper-redundant mobile robot. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Vol. 1. pp. 418–423.

Prautsch, P. and T. Mita (1999). Control and analysis of the gait of snake robots. In: *Proc. IEEE Int. Conf. Control Applications*. Kohala Coast, HI USA. pp. 502–507.

Prautsch, P., T. Mita and T. Iwasaki (2000). Analysis and control of a gait of snake robot. *Trans. IEE J. Ind. Appl. Soc.* **120-D**, 372–381. Mer utfyllende versjon av "Control and Analysis of the Gait of Snake Robots" (Prautsch1999).

Reyhanoglu, M., A. van der Schaft, N.H. McClamroch and I. Kolmanovsky (1999). Dynamics and control of a class of underactuated mechanical systems. *IEEE Transactions on Automatic Control* **44**(9), 1663–1671.

Rincon, D.M. and J. Sotelo (2003). Ver-Vite: Dynamic and experimental analysis for inchwormlike biomimetic robots. *IEEE Robot. Autom. Mag.* **10**(4), 53–57.

Saito, M., M. Fukaya and T. Iwasaki (2002). Serpentine locomotion with robotic snakes. *IEEE Contr. Syst. Mag.* **22**(1), 64–81.

Sanders, J. A., F. Verhulst and J. Murdock (2007). *Averaging Methods in Nonlinear Dynamical Systems*. Vol. 59 of *Applied Mathematical Sciences*. 2nd ed.. Springer.

Sato, T., W. Watanabe and A. Ishiguro (2010). An adaptive decentralized control of a serpentine robot based on the discrepancy between body, brain and environment. In: *Proc. IEEE Int. Conf. Robotics and Automation*. pp. 709 –714.

Schaft, A. J. van der and J. M. Schumacher (2000). *An Introduction to Hybrid Dynamical System*. Springer.

Sfakiotakis, M. and D. Tsakiris (2007). Biomimetic centering for undulatory robots. *The Int. Journal of Robotics Research* **26**, 1267–1282.

Shan, Y. and Y. Koren (1993). Design and motion planning of a mechanical snake. *IEEE Trans. Syst. Man Cyb.* **23**(4), 1091–1100.

Shan, Yansong and Yoram Koren (1995). Obstacle accommodation motion planning. *IEEE Trans. Robotic. Autom.* **11**(1), 36–49.

Shapiro, A., A. Greenfield and H. Choset (2007). Frictional compliance model development and experiments for snake robot climbing. In: *Proc. IEEE Int. Conf. Robotics and Automation*. pp. 574–579.

Song, P., P. Kraus, V. Kumar and P. Dupont (2000). Analysis of rigid body dynamic models for simulation of systems with frictional contacts. *Journal of Applied Mechanics* **68**, 118–128.

Sørdalen, O. J. and O. Egeland (1995). Exponential stabilization of nonholonomic chained systems. *IEEE Trans. Automatic Control* **40**(1), 35 – 49.

Spong, M. W. (1994). Partial feedback linearization of underactuated mechanical systems. In: *Proc. IEEE/RSJ/GI Int. Conf. Intelligent Robots and Systems*. Vol. 1. pp. 314–321.

Sussmann, H. J. (1987). A general theorem on local controllability. *SIAM Journal on Control and Optimization* **25**(1), 158–194.

Taal, S. R., H. Yamada and S. Hirose (2009). 3 axial force sensor for a semi-autonomous snake robot. In: *Proc. IEEE Int. Conf. Robotics and Automation*. pp. 4057–4062.

Tanaka, M. and F. Matsuno (2008*a*). Control of 3-dimensional snake robots by using redundancy. In: *IEEE Int. Conf. Robotics and Automation*. pp. 1156–1161.

Tanaka, M. and F. Matsuno (2009). A study on sinus-lifting motion of a snake robot with switching constraints. In: *Proc. IEEE Int. Conf. Robotics and Automation*. pp. 2270–2275.

Tanaka, Motoyasu and Fumitoshi Matsuno (2008*b*). Modeling and control of a snake robot with switching constraints. In: *SICE Annual Conference*.

Tanev, I., T. Ray and A. Buller (2005). Automated evolutionary design, robustness, and adaptation of sidewinding locomotion of a simulated snake-like robot. *IEEE Trans. on Robotics* **21**(4), 632–645.

Togawa, K., M. Mori and S. Hirose (2000). Study on three-dimensional active cord mechanism: Development of ACM-R2. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. Vol. 3. pp. 2242–2247.

Transeth, A. A., R. I. Leine, C. Glocker, K. Y. Pettersen and P. Liljebäck (2008*a*). Snake robot obstacle aided locomotion: Modeling, simulations, and experiments. *IEEE Trans. Robot.* **24**(1), 88–104.

Transeth, Aksel A., Nathan van de Wouw, Alexey Pavlov, Joao P. Hespanha and Kristin Y. Pettersen (2007). Tracking control for snake robot joints. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. San Diego, CA, USA. pp. 3539–3546.

Transeth, Aksel A., Remco I. Leine, Christoph Glocker and Kristin Y. Pettersen (2008*b*). 3D snake robot motion: Non-smooth modeling, simulations, and experiments. *IEEE Trans. on Robotics* **24**(2), 361–376.

Trinkle, J., J. S. Pang, S. Sudarsky and G. Lo (1997). On dynamic multi-rigid-body contact problems with coulomb friction. *Zeitschrift für Angewandte Mathematik und Mechanik* **77**, 267–280.

Tsatsomeros, M. J. and L. Li (2000). A recursive test for P-matrices. *Bit Numerical Mathematics* **40**, 410–414.

Ute, J. and K. Ono (2002). Fast and efficient locomotion of a snake robot based on self-excitation principle. In: *Proc. 7th Int. Workshop on Advanced Motion Control*. pp. 532–539.

Vela, P. A., K. A. Morgansen and J. W. Burdick (2002). Underwater locomotion from oscillatory shape deformations. In: *Proc. IEEE Conf. Decision and Control*. Vol. 2. pp. 2074–2080 vol.2.

Wang, Z., S. Ma, B. Li and Y. Wang (2010). Stability and adaptability of passive creeping of a snake-like robot. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. pp. 395–400.

Watanabe, K., M. Iwase, S. Hatakeyama and T. Maruyama (2008). Control strategy for a snake-like robot based on constraint force and verification by experiment. In: *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. pp. 1618–1623.

Westervelt, E. R., J. W. Grizzle, C. Chevallereau, J. H. Choi and B. Morris (2007). *Feedback Control of Dynamic Bipedal Robot Locomotion*. CRC Press.

Wiriyacharoensunthorn, P. and S. Laowattana (2002). Analysis and design of a multi-link mobile robot (serpentine). In: *Proc. IEEE Int. Conf. Robotics, Intelligent Systems and Signal Processing*. Vol. 2. pp. 694–699.

Worst, R. and R. Linnemann (1996). Construction and operation of a snake-like robot. In: *Proc. IEEE Int. Joint Symp. Intelligence and Systems*. Rockville, MD USA. pp. 164–169.

Wright, Cornell, Aaron Johnson, Aaron Peck, Zachary McCord, Allison Naaktgeboren, Philip Gianfortoni, Manuel Gonzalez-Rivero, Ross Hatton and Howie Choset (2007). Design of a modular snake robot. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. pp. 2609–2614.

Yamada, H. and S. Hirose (2006*a*). Study on the 3d shape of active cord mechanism. In: *Proc. IEEE Int. Conf. Robotics and Automation*. pp. 2890–2895.

Yamada, H. and S. Hirose (2008). Study of active cord mechanism - generalized basic equations of the locomotive dynamics of acm and analysis of sinus lifting. *Journal of the Robotics Society of Japan* **26**(7), 801–811. In Japanese.

Yamada, H. and S. Hirose (2009). Study of a 2-dof joint for the small active cord mechanism. In: *Proc. IEEE Int. Conf. Robotics and Automation*. pp. 3827–3832.

Yamada, H. and S. Hirose (2010). Steering of pedal wave of a snake-like robot by superposition of curvatures. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. pp. 419–424.

Yamada, Hiroya and Shigeo Hirose (2006*b*). Development of practical 3-dimensional active cord mechanism ACM-R4. *Journal of Robotics and Mechatronics* **18**(3), 1–7.

Yamada, Hiroya, S. Chigisaki, M. Mori, K.Takita, K. Ogami and Shigeo Hirose (2005). Development of amphibious snake-like robot ACM-R5. In: *Proc. 36th Int. Symp. Robotics*.

Yamakita, Masaki, Minoru Hashimoto and Takeshi Yamada (2003). Control of locomotion and head configuration of 3d snake robot (sma). In: *Proc. IEEE Int. Conf. Robotics and Automation*. Vol. 2. Taipei, Taiwan. pp. 2055 – 2060.

Ye, C., S. Ma, B. Li and Y. Wang (2004*a*). Locomotion control of a novel snake-like robot. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. Vol. 1.

Ye, C., S. Ma, B. Li and Y. Wang (2004*b*). Turning and side motion of snake-like robot. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Vol. 5. pp. 5075–5080.

Ye, C., S. Ma, B. Li, H. Liu and H. Wang (2007). Development of a 3d snake-like robot: Perambulator-II. In: *Int. Conf. Mechatronics and Automation*. pp. 117–122.

Yim, M. (1994). New locomotion gaits. In: *Proc. IEEE Int. Conf. on Robotics and Automation*. Vol. 3. pp. 2508–2514.

Yim, M., D.G. Duff and K.D. Roufas (2002). Walk on the wild side. *IEEE Robotics & Automation Magazine* **9**(4), 49–53.

Yu, S., S. Ma, B. Li and Y. Wang (2008). Analysis of helical gait of a snake-like robot. In: *IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics*. pp. 1183–1188.

Yu, S., S. Ma, B. Li and Y. Wang (2009). An amphibious snake-like robot: Design and motion experiments on ground and in water. In: *Int. Conf. Information and Automation*. pp. 500–505.

Zarrouk, D., I. Sharf and M. Shoham (2010). Analysis of earthworm-like robotic locomotion on compliant surfaces. In: *Proc. IEEE Int. Conf. Robotics and Automation*. pp. 1574–1579.

Zuo, Z., Z. Wang, B. Li and S. Ma (2008). Serpentine locomotion of a snake-like robot in water environment. In: *IEEE Int. Conf. Robotics and Biomimetics*. pp. 25–30.

Appendix A

Proof of Lemma 8.2

The proof of Lemma 8.2 has previously been presented in Pavlov *et al.* (2007) and is included here for completeness. The Lemma is proved by showing that a quadratic Lyapunov function candidate of the system (8.40) is negative definite, thereby implying that (8.40) is UGAS.

The system (8.40) can be written as

$$\begin{bmatrix} \dot{\overline{p}}_y \\ \dot{\overline{v}}_n \end{bmatrix} = \begin{bmatrix} -\frac{v_t}{\sqrt{\overline{p}_y^2+\Delta^2}} & \frac{\Delta}{\sqrt{\overline{p}_y^2+\Delta^2}} \\ \frac{X\Delta v_t}{\left(\sqrt{\overline{p}_y^2+\Delta^2}\right)^3} & \left(Y - \frac{X\Delta^2}{\left(\sqrt{\overline{p}_y^2+\Delta^2}\right)^3}\right) \end{bmatrix} \begin{bmatrix} \overline{p}_y \\ \overline{v}_n \end{bmatrix}. \tag{A.1}$$

Consider the quadratic Lyapunov function candidate $V = 1/2\overline{p}_y^2 + \kappa/2\overline{v}_n^2$ with $\kappa > 0$. The derivative of V along the solutions of (A.1) is given by

$$\begin{aligned} \dot{V} = \overline{p}_y\dot{\overline{p}}_y + \kappa\overline{v}_n\dot{\overline{v}}_n = -\frac{v_t\overline{p}_y^2}{\sqrt{\overline{p}_y^2+\Delta^2}} + \frac{\Delta\overline{p}_y\overline{v}_n}{\sqrt{\overline{p}_y^2+\Delta^2}} \\ +\kappa\frac{X\Delta v_t\overline{p}_y\overline{v}_n}{\left(\sqrt{\overline{p}_y^2+\Delta^2}\right)^3} + \kappa\left(Y - \frac{X\Delta^2}{\left(\sqrt{\overline{p}_y^2+\Delta^2}\right)^3}\right)\overline{v}_n^2. \end{aligned} \tag{A.2}$$

Since $v_t \in [V_{\min}, V_{\max}]$ by Assumption 8.1, and since $X \leq |X|$, we can estimate $\dot{V}$ as

$$\begin{aligned} \dot{V} \leq -\frac{V_{\min}\overline{p}_y^2}{\sqrt{\overline{p}_y^2+\Delta^2}} + \frac{\Delta\overline{p}_y\overline{v}_n}{\sqrt{\overline{p}_y^2+\Delta^2}} \\ +\kappa\frac{|X|\Delta V_{\max}\overline{p}_y\overline{v}_n}{\left(\sqrt{\overline{p}_y^2+\Delta^2}\right)^3} + \kappa\left(Y + \frac{|X|\Delta^2}{\left(\sqrt{\overline{p}_y^2+\Delta^2}\right)^3}\right)\overline{v}_n^2. \end{aligned} \tag{A.3}$$

By introducing the variable $z = \left|\overline{p}_y\right| / \sqrt{\overline{p}_y^2 + \Delta^2}$, this estimate can be writ-

ten as

$$\begin{aligned}\dot{V} &\leq -V_{\min} z^2 \sqrt{\overline{p}_y^2 + \Delta^2} \\ &+ \left(\Delta + \kappa \tfrac{|X| \Delta V_{\max}}{\overline{p}_y^2 + \Delta^2}\right) z \left|\overline{v}_n\right| + \kappa \left(Y + \tfrac{|X| \Delta^2}{\left(\sqrt{\overline{p}_y^2 + \Delta^2}\right)^3}\right) \overline{v}_n^2.\end{aligned} \tag{A.4}$$

Finally, using the inequalities $-\sqrt{\overline{p}_y^2 + \Delta^2} \leq -\Delta$ and $1/\left(\overline{p}_y^2 + \Delta^2\right) \leq 1/\Delta^2$, we obtain

$$\dot{V} \leq -V_{\min} \Delta z^2 + \left(\Delta + \kappa \frac{|X| \, V_{\max}}{\Delta}\right) z \left|\overline{v}_n\right| + \kappa \left(Y + \frac{|X|}{\Delta}\right) \overline{v}_n^2. \tag{A.5}$$

We now choose $\kappa = \Delta^2 \left(2\beta - 1\right) / \left(|X| \, V_{\max}\right)$, where

$$\beta = \frac{V_{\min} \left(-\Delta Y - |X|\right)}{V_{\max} \, |X|}. \tag{A.6}$$

It is straightforward to show that condition (8.28) of Theorem 8.2 is equivalent to $\beta > 1$. The chosen value of κ is therefore strictly positive. Substituting this κ into (A.5) gives

$$\begin{aligned}\dot{V} &\leq -V_{\min} \Delta z^2 + 2\beta \Delta z \left|\overline{v}_n\right| - \tfrac{\Delta(2\beta-1)\beta}{V_{\min}} \overline{v}_n^2 \\ &= -\Delta \left(\sqrt{V_{\min}} z - \tfrac{\beta |\overline{v}_n|}{\sqrt{V_{\min}}}\right)^2 - \tfrac{\Delta(\beta-1)\beta}{V_{\min}} \overline{v}_n^2.\end{aligned} \tag{A.7}$$

Finally, substituting the expression for z into this estimate gives

$$\dot{V} \leq -\Delta \left(\frac{\sqrt{V_{\min}} \left|\overline{p}_y\right|}{\sqrt{\overline{p}_y^2 + \Delta^2}} - \frac{\beta \left|\overline{v}_n\right|}{\sqrt{V_{\min}}}\right)^2 - \frac{\Delta \left(\beta - 1\right) \beta}{V_{\min}} \overline{v}_n^2. \tag{A.8}$$

Since condition (8.28) guarantees that $\beta > 1$, we can conclude that $\dot{V} < 0$, which implies that the origin of the system (8.40) is UGAS (see Khalil, 2002). This completes the proof of Lemma 8.2.

Appendix B

Proof of Lemma 8.3

The lemma is proved by showing that the system (8.40) is ULES (uniformly locally exponentially stable), which, together with the UGAS property established by Lemma 8.2, implies that (8.40) is globally $\mathcal{K}$-exponentially stable according to Corollary 8.1.

The linearization of the system (8.40) about the origin is easily calculated as

$$\begin{bmatrix} \dot{\overline{p}}_y \\ \dot{\overline{v}}_n \end{bmatrix} = \begin{bmatrix} -\frac{v_t}{\Delta} & 1 \\ \frac{Xv_t}{\Delta^2} & Y - \frac{X}{\Delta} \end{bmatrix} \begin{bmatrix} \overline{p}_y \\ \overline{v}_n \end{bmatrix}. \tag{B.1}$$

Denoting the system matrix of (B.1) by $\boldsymbol{W}$, we can calculate the eigenvalues of $\boldsymbol{W}$ from its characteristic equation

$$\lambda^2 - \operatorname{tr}(\boldsymbol{W})\,\lambda + \det(\boldsymbol{W}) = 0, \tag{B.2}$$

where $\operatorname{tr}(\boldsymbol{W})$ and $\det(\boldsymbol{W})$ are the trace and the determinant of $\boldsymbol{W}$, respectively. $\boldsymbol{W}$ is Hurwitz (see e.g. Khalil, 2002) if the coefficients of this characteristic equation are strictly positive, i.e. if $\operatorname{tr}(\boldsymbol{W}) < 0$ and $\det(\boldsymbol{W}) > 0$. Since $v_t > 0$, $Y < 0$, and $\Delta > 2\,|X|\,/\,|Y|$ (this follows from (8.28)), the trace of $\boldsymbol{W}$ satisfies

$$\begin{aligned} \operatorname{tr}(\boldsymbol{W}) &= -\tfrac{v_t}{\Delta} + Y - \tfrac{X}{\Delta} \leq -\tfrac{v_t}{\Delta} - |Y| + \tfrac{|X|}{\Delta} \\ &\leq -\tfrac{v_t}{\Delta} - |Y| + \tfrac{1}{2}\,|Y| = -\tfrac{v_t}{\Delta} - \tfrac{1}{2}\,|Y| < 0, \end{aligned} \tag{B.3}$$

and the determinant of $\boldsymbol{W}$ satisfies

$$\det(\boldsymbol{W}) = -\frac{v_t}{\Delta}\left(Y - \frac{X}{\Delta}\right) - \frac{Xv_t}{\Delta^2} = -\frac{v_t}{\Delta}Y > 0. \tag{B.4}$$

The system matrix $\boldsymbol{W}$ of the linearized system (B.1) is therefore Hurwitz, which implies that the origin of the system (8.40) is ULES (see

Khalil (2002), Corollary 4.3). Since, by Lemma 8.2, the origin of (8.40) is also UGAS, Corollary 8.1 implies that the origin of (8.40) is globally $\mathcal{K}$-exponentially stable. This completes the proof of Lemma 8.3.

Appendix C

Low-pass Filtering Reference Models

In order to ensure that the state reference of a control system complies with the dynamical capabilities of the system, the commanded state of the system can be passed through a low-pass filter. The filter, which is called a *reference model*, keeps the output from the control system within the physical capabilities of the actuators of the system. In addition to the filtered state reference, the output from the filter typically also includes the derivatives of the state reference with respect to time, which are often needed in the control law of the system. A low-pass filtering reference model can for example be used in combination with PD-control of the angle of a mechanical joint. Since the derivative part of the PD-controller includes the derivative of the reference angle with respect to time, any steps in the reference angle would make the output from the PD-controller infinitly large. This infinite control output is avoided by using a reference model to low-pass filter the reference angle.

In the following, we present a *2nd order* and a *3rd order low-pass filtering reference model* for a control system with a single scalar state value. The commanded state reference is denoted by $r \in \mathbb{R}$ and the filtered state reference is denoted by $x_{\text{ref}} \in \mathbb{R}$. In other words, we want the system to reach the state r, but we apply the filtered state reference x_{ref} to the controller in order to ensure that the system is brought to r through smooth control actions. For control systems with multiple states, the reference model can be applied individually to each state. The following material is based on Chapter 5 in Fossen (2002).

C.1 A 2nd order low-pass filtering reference model

A reference model of 2nd order ensures smooth reference signals for x_{ref} and $\dot{x}_{\text{ref}}$, but not for $\ddot{x}_{\text{ref}}$. We define the reference model as a *mass-damper-spring* system with the transfer function

$$\frac{x_{\text{ref}}}{r} = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}, \tag{C.1}$$

where $\omega > 0$ and $\zeta > 0$ are, respectively, the *natural frequency* and the *relative damping ratio* of the mass-damper-spring system. The reference model, which satisfies

$$\lim_{t\to\infty} x_{\text{ref}} = r, \tag{C.2}$$

can be written

$$\ddot{x}_{\text{ref}} + 2\zeta\omega\dot{x}_{\text{ref}} + \omega^2 x_{\text{ref}} = \omega^2 r, \tag{C.3}$$

and can be implemented in a control system by defining the state vector

$$\boldsymbol{w} = \begin{bmatrix} x_{\text{ref}} \\ \dot{x}_{\text{ref}} \end{bmatrix} \tag{C.4}$$

and calculating the dynamics of this state according to

$$\dot{\boldsymbol{w}} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\zeta\omega \end{bmatrix} \boldsymbol{w} + \begin{bmatrix} 0 \\ \omega^2 \end{bmatrix} r. \tag{C.5}$$

C.2 A 3rd order low-pass filtering reference model

In order to ensure that x_{ref}, $\dot{x}_{\text{ref}}$, and $\ddot{x}_{\text{ref}}$ are all sufficiently smooth, the filter should be of 3rd order. This is achieved by cascading the mass-damper-spring system (C.1) with a 1st order low-pass filter, which gives the transfer function

$$\frac{x_{\text{ref}}}{r} = \frac{\omega^2}{(1+Ts)\left(s^2 + 2\zeta\omega s + \omega^2\right)}, \tag{C.6}$$

where $T = 1/\omega$ is the time constant of the 1st order low-pass filter. This transfer function can be written

$$\frac{x_{\text{ref}}}{r} = \frac{\omega^3}{s^3 + (2\zeta+1)\,\omega s^2 + (2\zeta+1)\,\omega^2 s + \omega^3}. \tag{C.7}$$

The reference model, which satisfies

$$\lim_{t\to\infty} x_{\text{ref}} = r, \tag{C.8}$$

can be written

$$x_{\text{ref}}^{(3)} + (2\zeta+1)\,\omega\ddot{x}_{\text{ref}} + (2\zeta+1)\,\omega^2\dot{x}_{\text{ref}} + \omega^3 x_{\text{ref}} = \omega^3 r, \tag{C.9}$$

and can be implemented in a control system by defining the state vector

$$\boldsymbol{w} = \begin{bmatrix} x_{\text{ref}} \\ \dot{x}_{\text{ref}} \\ \ddot{x}_{\text{ref}} \end{bmatrix} \tag{C.10}$$

and calculating the dynamics of this state according to

$$\dot{\boldsymbol{w}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\omega^3 & -(2\zeta+1)\,\omega^2 & -(2\zeta+1)\,\omega \end{bmatrix} \boldsymbol{w} + \begin{bmatrix} 0 \\ 0 \\ \omega^3 \end{bmatrix} r. \tag{C.11}$$