

Aksel Andreas Transeth

Modelling and Control of Snake Robots

Thesis for the degree of philosophiae doctor

Trondheim, September 2007

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



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

©Aksel Andreas Transeth

ISBN 978-82-471-4865-5 (printed ver.)
ISBN 978-82-471-4879-2 (electronic ver.)
ISSN 1503-8181

Theses at NTNU, 2008:2

Printed by Tapir Uttrykk

ITK Report 2007-3-W

Summary

Snake robots have the potential of contributing vastly in areas such as rescue missions, fire-fighting and maintenance where it may either be too narrow or too dangerous for personnel to operate. This thesis reports novel results within modelling and control of snake robots as steps toward developing snake robots capable of such operations.

A survey of the various mathematical models and motion patterns for snake robots found in the published literature is presented. Both purely kinematic models and models including dynamics are investigated. Moreover, different approaches to both biologically inspired locomotion and artificially generated motion patterns for snake robots are discussed.

Snakes utilize irregularities in the terrain, such as rocks and vegetation, for faster and more efficient locomotion. This motivates the development of snake robots that actively use the terrain for locomotion, i.e. obstacle aided locomotion. In order to accurately model and understand this phenomenon, this thesis presents a novel non-smooth (hybrid) mathematical model for 2D snake robots, which allows the snake robot to push against external obstacles apart from a flat ground. Subsequently, the 2D model is extended to a non-smooth 3D model. The 2D model offers an efficient platform for testing and development of planar snake robot motion patterns with obstacles, while the 3D model can be used to develop motion patterns where it is necessary to lift parts of the snake robot during locomotion. The framework of non-smooth dynamics and convex analysis is employed to be able to systematically and accurately incorporate both unilateral contact forces (from the obstacles and the ground) and spatial friction forces based on Coulomb's law using set-valued force laws. Snake robots can easily be constructed for forward motion on a flat surface by adding passive caster wheels on the underside of the snake robot body. However, the advantage of adding wheels suffers in rougher terrains. Therefore, the models in this thesis are developed for wheel-less snake robots to aid future development of motion patterns that do not rely on passive wheels.

For numerical integration of the developed models, conventional numerical solvers can not be employed directly due to the set-valued force laws and possible instantaneous velocity changes. Therefore, we show how to implement the models for simulation with a numerical integrator called the time-stepping method. This method helps to avoid explicit changes between equations during simulation even though the system is hybrid.

Both the 2D and the 3D mathematical models are verified through experiments. In particular, a back-to-back comparison between numerical simulations and experimental results is presented. The results compare very well for obstacle aided locomotion.

The problem of model-based control of the joints of a planar snake robot without wheels is also investigated. Delicate operations such as inspection and maintenance in industrial environments or performing search and rescue operations require precise control of snake robot joints. To this end, we present a controller that asymptotically stabilizes the joints of a snake robot to a desired reference trajectory. The 2D and 3D model referred to above are ideal for simulation of various snake robot motion pattern. However, it is also advantageous to model the snake robot based the standard equations of motion for the dynamics of robot manipulators. This latter modelling approach is not as suited for simulation of a snake robot due to its substantial number of degrees of freedom, but a large number of control techniques are developed within this framework and these can now be employed for a snake robot. We first develop a process plant model from the standard dynamics of a robot manipulator. Then we derive a control plant model from the process plant model and develop a controller based on input-output linearization of the control plant model. The control plant model renders the controller independent of the global orientation of the snake robot as this is advantageous for the stability analysis. Asymptotic stability of the desired trajectory of the closed-loop system is shown using a formal Lyapunov-based proof. Performance of the controller is, first, tested through simulations with a smooth dynamical model and, second, with a non-smooth snake robot model with set-valued Coulomb friction.

The three main models developed in this thesis all serve important purposes. First, the 2D model is for investigating planar motion patterns by effective simulations. Second, the 3D model is for developing motion patterns that require two degrees of freedom rotational joints on the snake robot. Finally, the control plant model is employed to investigate stability and to construct a model-based controller for a planar snake robot so that its joints are accurately controlled to a desired trajectory.

Preface

This thesis contains the results of my doctoral studies from August 2004 to September 2007 at the Department of Engineering Cybernetics (ITK) at the Norwegian University of Science and Technology (NTNU) under the guidance of Professor Kristin Ytterstad Pettersen. The research is funded by the Strategic University Program on Computational Methods in Nonlinear Motion Control sponsored by The Research Council of Norway.

I am grateful to my supervisor Professor Kristin Ytterstad Pettersen for the support and encouragement during my doctoral studies. She has been a mentor in how to do research and our meetings have always been joyful ones. I am thankful for her constructive feedback on my research results and publications which have taught me how to convey scientific results in a to-the-point manner. In addition, I am much obliged to her for introducing me to various very skilled people around the world, which allowed me to be a visiting researcher in Zürich and Santa Barbara.

I am thankful for the invitation of Dr. ir. habil. Remco I. Leine and Professor Christoph Glocker to visit them at the Center of Mechanics at the Eidgenössische Technische Hochschule (ETH) Zürich in Switzerland. The introduction to non-smooth dynamics given to me by Dr. ir. habil. Remco I. Leine together with the guidance I got during my stay there are invaluable. In addition, I would like to thank the rest of the people at ETH Center for Mechanics for making my stay there a pleasant one.

I thank Professor João Pedro Hespanha for having me as a visitor at the Center for Control, Dynamical systems, and Computation (CCDC) at the University of California Santa Barbara (UCSB) in the USA. I appreciate the valuable advice and ideas I got from him. In addition, I would like to thank Professor Nathan van de Wouw at the Eindhoven University of Technology (TU/e) for a fruitful collaboration and fun time together at UCSB together with the interesting time I had during my short visit at TU/e.

I appreciate the discussions with my fellow PhD-students at NTNU,

and I would particularly like to point out the conversations with my former office mate Svein Hovland and current office mate Luca Pivano. In addition, I greatly acknowledge the constructive debates and advice concerning all aspects of snake robots I have got from my friend Pål Liljebäck. Moreover, I thank Dr. Øyvind Stavdahl for sharing his ideas on and enthusiasm for snake robots.

I express my deepest gratitude to Dr. Alexey Pavlov for guiding me in the world of non-linear control and for his numerous constructive comments and valuable feedback on my thesis.

I thank all my colleagues at the department of Engineering Cybernetics for providing me with a good environment in which it was nice to do research. I thank Terje Haugen and Hans Jørgen Berntsen at the department workshop for building the snake robot employed in the experiments, and for sharing hands-on knowledge in the design phase. Also, I thank the students Kristoffer Nyborg Gregertsen and Sverre Brovoll who were both involved in the hardware and software design and implementation needed to get the snake robot working. Finally, I would like to thank Stefano Bertelli for always helping out with camcorders and movie production for presentations and Unni Johansen, Eva Amdahl and Tove K. B. Johnsen for taking care of all the administrative issues that arose during the quest for a PhD-degree.

Finally, I thank my parents for always believing in me, and I thank my girlfriend Bjørg Riibe Ramskjell for all her love and support.

Trondheim, September 2007

Aksel Andreas Transeth

Publications

The following is a list of publications produced during the work on this thesis.

Journal papers

- Transeth, A. A. and K. Y. Pettersen (2008). A survey on snake robot modeling and locomotion. *Robotica*. Submitted.
- Transeth, A. A., R. I. Leine, Ch. Glocker and K. Y. Pettersen (2008a). 3D snake robot motion: Modeling, simulations, and experiments. *IEEE Transactions on Robotics*. Accepted.
- Transeth, A. A., R. I. Leine, Ch. Glocker, K. Y. Pettersen and P. Liljebäck (2008b). Snake robot obstacle aided locomotion: Modeling, simulations, and experiments. *IEEE Transactions on Robotics*. Accepted.

Referred conference proceedings

- Transeth, A. A., R. I. Leine, Ch. Glocker and K. Y. Pettersen (2006a). Non-smooth 3D modeling of a snake robot with external obstacles. In: *Proc. IEEE Int. Conf. Robotics and Biomimetics*. Kunming, China. pp. 1189–1196.
- Transeth, A. A., R. I. Leine, Ch. Glocker and K. Y. Pettersen (2006b). Non-smooth 3D modeling of a snake robot with frictional unilateral constraints. In: *Proc. IEEE Int. Conf. Robotics and Biomimetics*. Kunming, China. pp. 1181–1188
- Transeth, A. A. and K. Y. Pettersen (2006). Developments in snake robot modeling and locomotion. In: *Proc. IEEE Int. Conf. Control, Automation, Robotics and Vision*. Singapore. pp. 1393–1400.

- Transeth, A. A., N. van de Wouw, A. Pavlov, J. P. Hespanha and K. Y. Pettersen (2007a), Tracking control for snake robot joints. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. San Diego, CA, USA. pp. 3539–3546.
- Transeth, A. A., P. Liljebäck and K. Y. Pettersen (2007b). 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. pp. 2582–2589.

Contents

Summary	i
Preface	iii
Publications	v
1 Introduction	1
1.1 Motivation and Background	1
1.2 Main Contributions of this Thesis	7
1.3 Organization of this Thesis	10
2 Developments in Snake Robot Modelling and Locomotion	13
2.1 Introduction	13
2.2 Biological Snakes and Inchworms	14
2.2.1 Snake Skeleton	14
2.2.2 Snake Skin	14
2.2.3 Locomotion – The Source of Inspiration for Snake Robots	15
2.3 Design and Mathematical Modelling	17
2.3.1 Kinematics	18
2.3.2 Dynamics	21
2.4 Snake Robot Locomotion	28
2.4.1 Planar Snake Robot Locomotion	30
2.4.2 3D Snake Robot Locomotion	34
2.5 Discussion and Summary	38
3 Non-smooth Model of a 2D Snake Robot for Simulation	41
3.1 Introduction	41
3.2 Summary of the Mathematical Model	42
3.3 Kinematics	44

3.3.1	Snake Robot Description and Reference Frames . . .	44
3.3.2	Gap Functions for Obstacle Contact	46
3.3.3	Bilateral Constraints: Joints	48
3.4	Contact Constraints on Velocity Level	49
3.4.1	Relative Velocity Between an Obstacle and a Link .	49
3.4.2	Tangential Relative Velocity	51
3.4.3	Bilateral Constraints: Joints	52
3.5	Non-smooth Dynamics	53
3.5.1	The Equality of Measures	53
3.5.2	Constitutive Laws for Contact Forces	56
3.6	Numerical Algorithm: Time-Stepping	61
3.6.1	Time Discretization	62
3.6.2	Solving for the Contact Impulsions	63
3.6.3	Constraint Violation	65
3.7	Summary	65
4	3D Snake Robot Modelling for Simulation	67
4.1	Introduction	67
4.2	Kinematics	68
4.2.1	Model Description, Coordinates and Reference Frames	68
4.2.2	Gap Functions for Ground Contact	70
4.2.3	Gap Functions for Contact with Obstacles	71
4.2.4	Gap Functions for Bilateral Constraints	73
4.3	Contact Constraints on Velocity Level	75
4.3.1	Unilateral Contact: Ground Contact	75
4.3.2	Unilateral Contact: Obstacle Contact	80
4.3.3	Bilateral Constraints: Joints	82
4.4	Non-smooth Dynamics	83
4.4.1	The Equality of Measures	83
4.4.2	Constitutive Laws for Contact Forces	85
4.4.3	Joint Actuators	88
4.5	Accessing and Control of Joint Angles	89
4.6	Numerical Algorithm: Time-stepping	91
4.6.1	Time Discretization	91
4.6.2	Solving for Contact Impulsions	93
4.6.3	Constraint Violation	95
4.7	Summary	97

5	Obstacle Aided Locomotion	99
5.1	Introduction	99
5.2	Motivation	99
5.3	Understanding Obstacle Aided Locomotion	100
5.4	Requirements for Intelligent Obstacle Aided Locomotion	102
5.5	Experimental Observation of Obstacle Aided Locomotion	105
6	Modelling and Control of a Planar Snake Robot	107
6.1	Introduction	107
6.2	Mathematical Modelling	109
6.2.1	Process Plant Model	110
6.2.2	Control Plant Model	116
6.3	Joints Control by Input-output Linearization	119
6.3.1	Control Plant Model Reformulation	119
6.3.2	Controller Design	120
6.3.3	Final Results on Input-output Linearization	121
6.4	Summary	130
7	Simulation and Experimental Results	131
7.1	Introduction	131
7.2	Simulation Parameters	132
7.2.1	A Description of Aiko and Model Parameters	132
7.2.2	Simulation Parameters and Reference Joint Angles	133
7.3	Simulations without Experimental Validation	136
7.3.1	3D Model: Drop and Lateral Undulation	137
7.3.2	3D Model: U-shaped Lateral Rolling	139
7.3.3	3D Model: Sidewinding	140
7.3.4	2D and 3D Model: Obstacle Aided Locomotion	141
7.3.5	Robot Model: Comparison of Controllers	147
7.4	Experimental Setup	152
7.5	Simulations with Experimental Validation	153
7.5.1	3D Model: Sidewinding	153
7.5.2	2D and 3D Model: Lateral Undulation with Isotropic Friction	155
7.5.3	2D and 3D Model: Obstacle Aided Locomotion	156
7.5.4	Discussion of the Experimental Validation	159
7.6	Summary	161
8	Conclusions and Further Work	163

Bibliography	167
A An additional Proof and a Theorem for Chapter 6	177
A.1 Theorem on Boundedness	177
A.2 Positive Definiteness of \mathbf{M}_V	178

Chapter 1

Introduction

1.1 Motivation and Background

The wheel is an amazing invention, but it does not roll everywhere. Wheeled mechanisms constitute the backbone of most ground-based means of transportation. On relatively smooth surfaces such mechanisms can achieve high speeds and have good steering ability. Unfortunately, rougher terrain makes it harder, if not impossible, for wheeled mechanisms to move. In nature the snake is one of the creatures that exhibit excellent mobility in various terrains. It is able to move through narrow passages and climb on rough ground. This mobility property is attempted to be recreated in robots that look and move like snakes – snake robots. These robots most often have a high number of degrees of freedom (DOF) and they are able to move without using active wheels or legs.

Snake robots may one day play a crucial role in search and rescue operations, fire-fighting and inspection and maintenance. The highly articulated body allows the snake robot to traverse difficult terrains such as collapsed buildings or the chaotic environment caused by a car collision in a tunnel as visualized in Figure 1.1. The snake robot could move within destroyed buildings looking for people while simultaneously bringing communication equipment together with small amounts of food and water to anyone trapped by the shattered building. A rescue operation involving a snake robot is envisioned in Miller (2002). Moreover, the snake robot can be used for inspection and maintenance of complex and possibly hazardous areas of industrial plants such as nuclear facilities. In a city it could inspect the sewer system looking for leaks or aid fire-fighters. Also, snake robots with one end fixed to a base may be used as robot manipulators that can

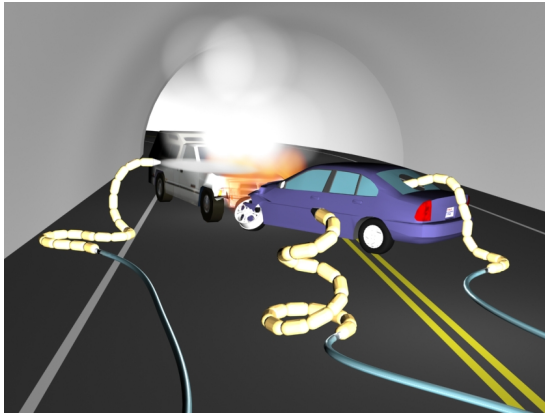


Figure 1.1: Fire-fighting snake robots to the rescue after a car accident in a tunnel. Courtesy of SINTEF (www.sintef.no/Snakefighter).

reach hard-to-get-to places.

Compared to wheeled and legged mobile mechanisms, the snake robot offers high stability and superior traversability. Moreover, the exterior of a snake robot can be completely sealed to keep dust and fluids out and this enhances its applicability. Furthermore, the snake robot is more robust to mechanical failures due to high redundancy and modularity. The downside is its limited payload capacity, poor power efficiency and a very large number of degrees of freedom that have to be controlled. For a more elaborate overview of numerous applications of snake robots, the reader is referred to Dowling (1997) and Choset *et al.* (2000).

The first qualitative research on snake locomotion was presented by Gray (1946) while the first working biologically inspired snake-like robot was constructed by Hirose almost three decades later in 1972 (Hirose, 1993). He presented a two-meter long serpentine robot with twenty revolute 1 DOF joints called the Active Cord Mechanism model ACM III, which is shown in Figure 1.2. Passive casters were put on the underside of the robot so that forward planar motion was obtained by moving the joints from side to side in selected patterns.

Since Hirose presented his Active Cord Mechanism model ACM III, many multi-link articulated robots intended for undulating locomotion have been developed and they have been called by many names. Some examples are: multi-link mobile robot (Wiriya-charoensunthorn and Laowattana, 2002), snake-like or snake robot (Kamegawa *et al.*, 2004; Lewis and

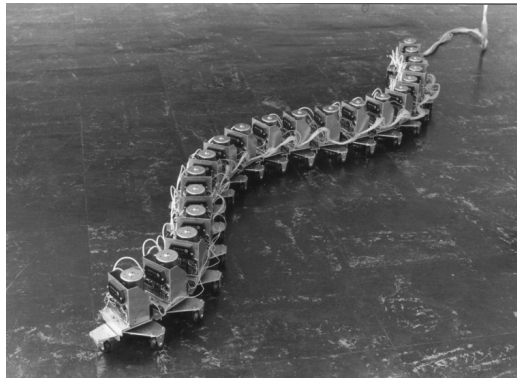


Figure 1.2: The Active Cord Mechanism model ACM III Hirose (1993). By permission of Oxford University Press.

Zehnpfennig, 1994; Lu *et al.*, 2003; Ma, 1999; Ma *et al.*, 2001*a*; Worst and Linnemann, 1996; Xinyu and Matsuno, 2003), hyper-redundant robot (Chirikjian and Burdick, 1994*b*) and G-snake (Krishnaprasad and Tsakiris, 1994). We employ the term ‘snake robot’ to emphasize that this thesis deals with robots whose motion mainly resembles the locomotion of snakes.

Research on snake robots has increased vastly during the past ten to fifteen years and the published literature is mostly focused on snake robot modelling and locomotion. In the remainder of this section we will present a summary of the previously published results most relevant to this thesis. A more thorough review on snake robot modelling and locomotion is given in Chapter 2.

The fastest and most common serpentine motion pattern used by biological snakes is called lateral undulation. In short, forward motion is obtained by this motion pattern by propagating waves from the front to the rear of the snake while exploiting roughness in the terrain. This has also been the most implemented motion pattern for snake robots. Note that by a ‘motion pattern’ or a ‘gait’ of a snake robot, we mean an (often repetitive) coordinated motion of the snake robot joints employed to move the snake robot in some direction. Snakes exploit irregularities in the terrain to push against to move forward by lateral undulation. This method of locomotion is attempted to be recreated for snake robots moving on a smooth surface by adding passive caster wheels (Ma, 2001; Ma *et al.*, 2003*a*; Ostrowski and Burdick, 1996; Wiriya Charoensunthorn and Laowattana, 2002; Ye *et al.*, 2004*a*) or metal skates (Saito *et al.*, 2002) on the underside of the

snake robot body. Relatively fast locomotion is obtained for snake robots with caster wheels travelling on a solid smooth surface. The dependency on the surface is important since the friction property of the snake robot links must be such that the links slide easier forward and backward than sideways for efficient snake robot locomotion by lateral undulation.

The dependency on the ground surface can be relaxed by mimicking biological snakes and utilizing external objects to move forward. This is the motivation for developing snake robots that exploit obstacles for locomotion. We define *obstacle aided locomotion* as the snake robot locomotion where the snake robot utilizes walls or other external objects, apart from the flat ground, for propulsion. Obstacle aided locomotion for snake robots was first investigated by Hirose in 1976 and experiments with a snake robot with passive caster wheels moving through a winding track were presented in Hirose and Umetani (1976) and Hirose (1993). More recently, Bayraktaroglu and Blazevic (2005) and Bayraktaroglu *et al.* (2006) elaborated on obstacle aided locomotion for wheel-less snake robots. The dynamics of such locomotion was simulated with the dynamic simulation software WorkingModel[®] in Bayraktaroglu and Blazevic (2005), where the rigid body contact (i.e. the contact between an obstacle and the snake robot) is represented by a spring-damper model (i.e. a compliant contact model).

Snake robots capable of 3D motion appeared more recently (Chirikjian and Burdick, 1993, 1995; Hirose and Morishima, 1990; Liljebäck *et al.*, 2005; Mori and Hirose, 2002) and, together with the robots, mathematical models of both the kinematics and the dynamics of snake robots were also developed. Purely kinematic 3D models were presented in Burdick *et al.* (1993), Chirikjian and Burdick (1995) and Ma *et al.* (2003*b*). In such models, friction is not considered in the contact between the snake robot and the ground surface. Instead, it is assumed in Ma *et al.* (2003*b*) that a snake robot link can slide without friction forward and backward. This can be achieved by adding passive caster wheels, which introduces non-holonomic constraints, to the underside of the snake robot. A different approach presented in Burdick *et al.* (1993) and Chirikjian and Burdick (1995) is to assume that the parts of the snake robot in contact with the ground are anchored to the ground surface. Then the other parts of the snake robot body may move without being subjected to friction forces.

A model of the motion dynamics is needed to describe the friction forces a snake robot is subjected to when moving over a surface. Most mathematical models that describe the dynamics of snake robot motion are limited to planar (2D) motion (Ma, 2001; Ma *et al.*, 2003*a*; McIsaac and Ostrowski,

2003b; Saito *et al.*, 2002) while 3D models of snake robots have only recently been developed (Liljebäck *et al.*, 2005; Ma *et al.*, 2004) where the contact between a snake robot and the ground surface is modelled as compliant. 3D models facilitate testing and development of 3D serpentine motion patterns such as sinus-lifting (similar to lateral undulation, but with an additional vertical wave) and sidewinding (a gait commonly employed by desert snakes). There exists various software to model and simulate dynamical systems. 3D simulations of a snake robot are presented in Dowling (1999, 1997) where such software is employed to define and simulate a 3D snake robot. However, details of this implementation are not presented and the simulations are thus not easily reproducible by the reader.

On a flat surface the ability of a snake robot to move forward is dependent on the friction between the ground surface and the body of the snake robot. Hence, the friction forces and the unilateral (upward) contact forces (which give rise to the friction forces) are important parts of the mathematical model of a snake robot. The friction forces are usually based on a Coulomb or viscous-like friction model (McIsaac and Ostrowski, 2003b; Saito *et al.*, 2002), and the Coulomb friction is most often modelled using the sign-function (Ma *et al.*, 2004; Saito *et al.*, 2002). The unilateral contact forces as a result of contact with the ground surface are modelled as a spring-damper system in Liljebäck *et al.* (2005), where the centre of gravity of each link is employed as the contact point.

We have focused mostly on modelling in the above review, and we now proceed to give a short comment on snake robot locomotion. Many authors base the choice and implementation of the most common serpentine motion pattern called lateral undulation on the serpenoid curve found in Hirose (1993). This is a curve that describes the motion of a biological snake while moving by lateral undulation. A snake robot can follow an approximation to this curve by setting its joint angles according to a sine-curve that is phase-shifted between adjacent joints. This approach to snake robot locomotion is widely implemented for snake robots that have either wheels (Ostrowski and Burdick, 1996; Prautsch and Mita, 1999) or a friction property such that each link of the snake robot glides easier forward and backward compared to transversal motion (Ma, 2001; Saito *et al.*, 2002). A no-slip constraint (i.e. a non-holonomic velocity constraint) on each wheel is sometimes introduced in the mathematical model and this results in that the snake robot links cannot slip sideways. Such an approach is presented in Prautsch and Mita (1999), where the no-slip constraint allows one to significantly reduce the model. A Lyapunov-based proof for this reduced model together with a

proposed controller shows that the snake robot is able to move to a position reference. A velocity controller for a wheel-less snake robot with the friction property described above is presented in Saito *et al.* (2002). The simulation and experimental results presented in that paper indicate that the snake robot is able to stay within a reasonable offset of a desired speed. Results on controllability based on a kinematic description of snake robot locomotion are presented in Kelly and Murray (1995) and Ma *et al.* (2003b).

From the review above we see that very interesting and useful results are published on snake robot modelling and locomotion. There are many important aspects to consider when modelling a snake robot. The friction model has to describe the frictional contact with the ground properly. Moreover, a model to describe a varying normal contact force with the ground surface is needed to simulate some motion patterns. Other approaches to snake robot locomotion rely on having objects to push against to move forward and in this case the contact forces between the snake robot and the obstacles have to be defined in some way. In addition, we need to define some sort of exoskeleton for the snake robot for contact with obstacles and the ground surface. In order to develop the previously published result on modelling of snake robots further, we note that only unidirectional Coulomb friction can be described by a sign-function since the direction of the friction force due to a snake robot link sliding on the ground surface will not point in the correct direction relative to the direction of the sliding velocity for spatial friction. Moreover, a sign-function with the commonly employed definition $\text{sign}(0) = 0$ (i.e. not set-valued) will not give a non-zero friction force for zero sliding velocity. However, such a non-zero friction force is sometimes acting on rigid bodies in the stick-phase of Coulomb friction. Hence, a snake robot on a slightly inclined surface will slowly slide downward even though it should stick to the surface due to the Coulomb friction. This is because the friction force becomes zero every time the velocity becomes zero. Hence, the above mentioned properties of Coulomb friction cannot be modelled with the previously published friction models. Furthermore, a very high spring coefficient is needed to model a hard obstacle (or ground surface) with a compliant contact model (which is employed in the previously published literature) and this leads to stiff differential equations which in turn are cumbersome to solve numerically. Moreover, it is not clear how to determine the dissipation parameters of the contact unambiguously when using a compliant model (Brogliato, 1999). Hence, there is a need for an alternative approach to describe Coulomb friction with an accurate stick-phase, and normal contact forces due to impacts with the

obstacles and the ground surface with a non-compliant model.

In addition, for 3D models of the dynamics of snake robots presented in such a way that the model is possible to re-create by the reader, only one point on each link (most often the centre of gravity) is considered as the contact point with the ground. Hence, rotational motion around the longitudinal axis of a snake robot link of such a point does not result in any translational motion since the point do not have a circumference. This makes the models unable to perform rolling gaits such as lateral rolling. Hence, there is a need for a well-described model of a snake robot with some sort of exoskeleton (instead of just a single stationary point) for contact with the ground surface.

For wheel-less snake robots it is not sufficient to consider a purely kinematic model for the motion pattern lateral undulation, as the friction between the snake robot and the ground surface is essential for locomotion. Therefore, the friction needs to be considered for wheel-less snake robots and this motivates including the dynamics in the development of model-based controllers for snake robots. Some controllers are developed for position control of a single link or a small selection of snake robot links. Moreover, controllers are proposed for velocity and heading control of a snake robot. However, for delicate operations such as inspection and maintenance in industrial plants, every point on the snake robot body need to be accurately controlled so that the snake robot do not unintentionally collide with objects on its path. Such operations require precise control of the snake robot joints and this can be realized through a model-based design. However, no such controller is found in the previously published literature.

1.2 Main Contributions of this Thesis

This thesis first presents a thorough review of snake robot modelling and locomotion. Then various mathematical models of a snake robot, together with a model-based control law are developed. Furthermore, we present simulation results of a number of motion patterns and some of these results are experimentally validated.

Review: During the last ten to fifteen years, the published literature on snake robots has increased vastly. This thesis gives an elaborate overview and comparison of the various mathematical models and locomotion principles for snake robots presented during this period. Both purely kinematic models and models including dynamics are investigated. Moreover, we provide an introduction to the source of

inspiration of snake robots: biologically inspired crawling locomotion. Furthermore, different approaches to both biologically inspired locomotion and artificially generated motion patterns for snake robots are discussed.

2D model: Propulsion by some sort of obstacle aided locomotion enables the snake robot to move forward in rough terrain. To this end, we present a novel non-smooth (hybrid) mathematical model for planar wheel-less snake robots that allows a snake robot to push against external obstacles apart from the flat ground. The framework of non-smooth dynamics and convex analysis allows us to systematically and accurately incorporate both unilateral contact forces (from the obstacles) and friction forces based on Coulomb's law using set-valued force laws. Hence, stick-slip transitions with the ground and impacts with the obstacles are modelled as instantaneous transitions. Moreover, the set-valued force law results in an accurate description of the Coulomb friction which is important for snake robot locomotion on a planar surface. Conventional numerical solvers can not be employed directly for numerical integration of this model due to the set-valued force laws and possible instantaneous velocity changes. Therefore, we show how to implement the model for numerical treatment with a numerical integrator called the time-stepping method. Even though we model the snake robot as a hybrid system, in this framework we avoid explicit switching between system equations (for example when a collision occurs). The description of the model and the method for numerical integration are presented in such a way that people who are new to the field of non-smooth dynamics can use this thesis and, in particular, Chapter 3 as an introduction to non-smooth modelling of robot manipulators with impacts and friction.

3D model: Many snake robot motion patterns require that the snake robot is able to perform more than just planar motion. Therefore, the 2D model is extended to a non-smooth 3D mathematical model of a snake robot (without wheels) where a particular choice of coordinates results in an effective way of writing the system equations. The same framework of non-smooth dynamics and convex analysis as in for the 2D model is employed for the 3D model. Hence, no explicit changes between equations are necessary when collisions occur. This is particularly advantageous for 3D motion since the snake robot links repeatedly collide with the ground surface during, for example, loco-

motion by sidewinding. Moreover, the set-valued force law employed to model the normal contact between the snake robot and the ground surface and obstacles is a beneficial alternative to the compliant (e.g. spring-damper) model employed in previously published works. We define for each link a cylindrical exoskeleton for contact with the ground surface and the external obstacles. Hence, a rolling motion will result in a sideways motion which would not be the case if only the centre of gravity is employed for contact with the environment. This enables us to simulate motion patterns such as lateral rolling.

Validation of models with experimental results: Both the 2D and the 3D mathematical model are verified through experiments. In particular, a back-to-back comparison between numerical simulations and experimental results is presented. Such a quantitative comparison is not found in any of the previously published works on snake robots. The simulations show that the numerical treatment of both models can handle both impacts with the floor and obstacles. The experiments are performed with the snake robot ‘Aiko’ depicted in Figure 1.3, which is a wheel-less snake robot with cylindrical links. Aiko has been developed at the NTNU/SINTEF Advanced Robotics Laboratory in Trondheim, Norway.

A ‘robot model’ and model-based control design: To be able to employ the model-based control techniques developed for robot manipulators, we employ the standard dynamics of a robot manipulator (found in, e.g., Spong and Vidyasagar (1989)) and develop a process plant model based on this framework for a planar snake robot. Such a representation is not favourable for simulation due to the high number of degrees of freedom of a snake robot so the 2D model mentioned earlier is still important for developing and testing new motion patterns. We employ one group of the techniques available for the robot manipulator model, called input-output linearization to develop a controller for tracking control of the snake robot joints. In order to make the stability analysis manageable, we develop a control plant model from the process plant model where the global orientation of the snake robot is removed from the model. This allows us to complete a formal Lyapunov-based proof which shows that the joint angles converge exponentially to their desired trajectory.

From the list of contributions given above, we see that several mathematical models of snake robots are presented in this thesis. All models



Figure 1.3: The snake robot ‘Aiko’ developed at the NTNU/SINTEF Advanced Robotics Laboratory. Aiko is used as a basis for the mathematical models.

describe the same snake robot. However, different aspects of snake robot locomotion is emphasized in the various models: The 2D model is for simulation of planar snake robot motion with or without obstacles to push against, while the 3D model is for simulation of 3D gaits like sidewinding or sinus-lifting where the use of obstacles can also be included. The 2D model is much faster to simulate and easier to implement since less degrees of freedom and contact points are considered. Hence, it should be employed as an effective test bed for the development of new planar motion patterns and snake robot controller testing. In addition, we have developed a 2D model of a snake robot based on the standard formulation of the dynamics of a robot manipulator. This model is denoted the ‘robot model’ and is in its analytical form is not favourable for simulations due to the intricate system matrices which arise from employing minimal generalized coordinates for a snake robot (with such a high number of degrees of freedom). However, the standard formulation provides us with a large range of tools for controller synthesis and stability analysis for the snake robot.

1.3 Organization of this Thesis

Chapter 2: The main developments in snake robot modelling and locomotion are reviewed.

Chapter 3: A 2D non-smooth model of a snake robot with obstacles to

push against is developed. The model is based on the framework of non-smooth dynamics and convex analysis. Instead of presenting the background theory in a separate chapter, we use the 2D model of a snake robot as an example and present the necessary theory when it is appropriate throughout this chapter.

Chapter 4: A 3D non-smooth model of a snake robot with obstacles to push against is developed. Those familiar with non-smooth dynamics and convex analysis can read this chapter independently from Chapter 3.

Chapter 5: The concept of obstacle aided locomotion for snake robots is elaborated on, and we provide an overview of how a biological snake moves by pushing against irregularities in the ground.

Chapter 6: A planar model of a snake robot based on the standard formulation of the dynamics of a robot manipulator is developed. Moreover, an input-output linearizing controller for asymptotic tracking for snake robot joints is designed and a formal stability proof is provided.

Chapter 7: Simulation results are given for all models. In addition, back-to-back comparisons between experimental results and simulation results with the 2D and 3D model are presented for a selection of serpentine motion patterns.

Chapter 8: Conclusions and suggestions to further work are presented.

Appendix A: An additional proof and a theorem employed in Chapter 6 are described.

Chapter 2

Developments in Snake Robot Modelling and Locomotion

2.1 Introduction

During the last ten to fifteen years, the published literature on snake robots has increased vastly and the purpose of this chapter is to provide an elaborate survey of the various mathematical models and locomotion principles of snake robots presented during this period. Both purely kinematic models and models including dynamics are investigated. Moreover, different approaches to both biologically inspired locomotion and artificially generated motion patterns for snake robots are discussed. We also provide an introduction to the source of inspiration for snake robot locomotion: serpentine and crawling locomotion of biological creatures. The specific choices of hardware for e.g. sensors and actuators for snake robots are beyond the scope of this chapter and will not be discussed. In addition, we do not consider snake robots with *active* wheels in this thesis. For more information on such snake robots see, for example, Kamegawa *et al.* (2004), Yamada and Hirose (2006a) and Masayuki *et al.* (2004).

Note that the notation in this chapter sometimes differs from the rest of this thesis. This is because in this chapter we strive to employ the same notation as used in the papers we discuss to ease the transition for those who want to explore these source papers further. However, we balance between using the exact same notation as in the various original papers and using a common notation for this chapter, so the correspondence is not

always one-to-one.

This chapter is based on Transeth and Pettersen (2006) and Transeth and Pettersen (2008) and it is arranged as follows: Section 2.2 gives a short introduction to snakes and serpentine and crawling locomotion of biological creatures. Various mathematical models of snake robots are presented in Section 2.3. Section 2.4 provides an overview of numerous motion patterns implemented on snake robots, while the survey presented in this chapter is discussed and summarized in Section 2.5.

2.2 Biological Snakes and Inchworms

Biological snakes, inchworms and caterpillars are the source of inspiration for most of the robots dealt with in this chapter. We will therefore start with a short introduction to snake physiology and snake locomotion. In addition, inchworm and caterpillar motion patterns are outlined. Unless otherwise specified, the contents in this section are based on Mattison (2002), Bauchot (1994) and Dowling (1997).

2.2.1 Snake Skeleton

The skeleton of a snake often consists of at least 130 vertebrae and can exceed 400 vertebrae. The range of movement between each joint is limited to between 10 and 20 degrees for rotation from side to side and to a few degrees of rotation when moving up and down. A large total curvature of the snake body is still possible because of the high number of vertebrae.

A very small rotation is also possible around the direction along the snake body. This property is employed when the snake moves sideways by sidewinding.

2.2.2 Snake Skin

Since snakes have no legs, the skin surface plays an important role in snake locomotion (Bauchot, 1994). The snake should experience little friction when sliding forward, but great friction when pushed backward. The skin is usually covered with scales with tiny indentations which facilitate forward locomotion. The scales form an edge to the belly during motion which results in that the friction between the underside of the snake and the ground is higher transversal to the snake body than along it (Hirose, 1993).

2.2.3 Locomotion – The Source of Inspiration for Snake Robots

Most motion patterns implemented for snake robots are inspired by locomotion of snakes. However, inchworms and caterpillars are also used as inspiration. The relevant motion patterns of all these creatures will be outlined in the following.

Lateral Undulation

Lateral undulation (also denoted serpentine crawling) is a continuous movement of the entire body of the snake relative to the ground. Locomotion is obtained by propagating waves from the front to the rear of the snake while exploiting roughness in the terrain. Every part of the body passes the same part of the ground ideally leaving a single sinus-like track as illustrated in Figure 2.1 (a). To prevent lateral slipping while moving forward, the snake ‘digs’ in to the ground with help of the edge described in Section 2.2.2. In addition, it may use contours such as rocks on the ground to push against. All the contact points with the ground constitute possible push-points for the snake and the snake needs at least three push-points to obtain a continuous forward motion. Two points are needed to generate forces and the third point is used to balance the forces such that they act forward.

The efficiency of lateral undulation is mainly based on two factors. 1) The contour of the ground. The more contoured the ground, the more efficient the locomotion. 2) The ratio between the length of the snake and its circumference. The fastest snakes have a length that is no longer than 10 to 13 times their circumference. Speeds up to 11 km/h have been observed in rough terrains.

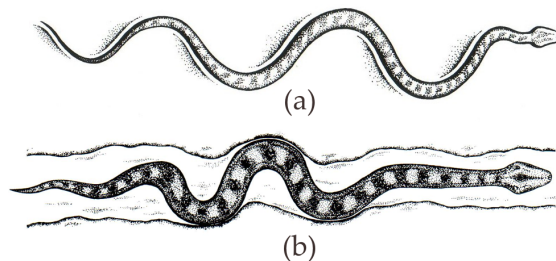


Figure 2.1: (a) Lateral undulation and (b) concertina locomotion (Mattison, 2002). By permission of Cassell Illustrated.

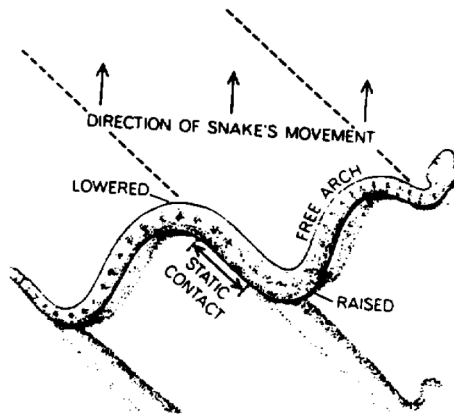


Figure 2.2: Sidewinding locomotion (Burdick *et al.*, 1993) © 1993 IEEE.

Concertina Locomotion

A concertina is a small accordion instrument. The name is used in snake locomotion to indicate that the snake stretches and folds its body to move forward. The folded part is kept at a fixed position while the rest of the body is either pushed or pulled forward as shown in Figure 2.1 (b). Then, the two parts switch roles. Forward motion is obtained when the force needed to push back the fixed part of the snake body is higher than the friction forces acting on the moving part of the body.

Concertina locomotion is employed when the snake moves through narrow passages such as pipes or along branches. If the path is too narrow compared to the diameter and curving capacity of the snake, the snake is unable to progress by this motion pattern.

Sidewinding Locomotion

Sidewinding is probably the most astonishing gait to observe and is mostly used by snakes in the desert. The snake lifts and curves its body leaving short, parallel marks on the ground while moving at an inclined angle as shown in Figure 2.2. Unlike lateral undulation there is a brief static contact between the body of the snake and the ground.

Sidewinding is usually employed on surfaces with low shear such as sand. The snakes can reach velocities up to 3 km/h during sidewinding locomotion.

Other Snake Gaits

Snakes also have gaits that are employed in special situations or by certain species. These are e.g. rectilinear crawling, burrowing, jumping, sinus-lifting, skidding, swimming and climbing. The latter four, which are or may be used for snake robots, are described as follows.

Sinus-lifting is a modification of lateral undulation where parts of the trunk are lifted to avoid lateral slippage and to optimise propulsive force (Hirose, 1993). This gait is employed for high speeds.

A variation of lateral undulation is called *skidding* (also denoted *slide-pushing*) and is employed when moving past low-friction surfaces. The snake rests its head on the ground and then sends a flexion wave down through its body. This is repeated in a zigzag pattern and is a very energy-inefficient way of locomotion.

Almost all snakes can swim. They move forward by undulating laterally like an eel.

Long and thin bodied snakes can climb trees by vertical lateral undulation. Parts of their body hang freely in the air, while branches are used as support.

Inchworm and Caterpillar Locomotion

An inchworm moves forward by grabbing the ground with its front legs while the rear end is pulled forward. The rear legs then grab the ground and the inchworm lifts its front legs and straightens its body. Caterpillars send a vertical travelling wave through their body from the end to the front in order to move forward. Small legs give the necessary friction force while on the ground.

2.3 Design and Mathematical Modelling

The mathematical model of a snake robot, of course, depends on its design. To categorize the different snake robot designs we recognize certain basic properties: 1) Type of joints, 2) number of degrees of freedom (DOF) and 3) with or without passive caster wheels. Most snake robots consist of links connected by revolute joints with one or two DOF. On some robots, the links are extensible (i.e. prismatic joints). To achieve the desired frictional property for lateral undulation mentioned in Section 2.2, some snake robots are equipped with passive wheels. When wheels are employed, the dynamics of the interaction between the robot and the ground surface is often ignored.

If no wheels are attached, this friction force needs to be considered for some, but not all, gaits (see Section 2.4).

In the following, the mathematical modelling of the different snake robots is divided into kinematics and dynamics.

2.3.1 Kinematics

The kinematics describes the geometrical aspect of motion. Different modelling techniques ranging from classical methods such as the Denavit-Hartenberg (D-H) convention (see e.g. Murray *et al.* (1994) for more on the D-H convention) to specialized methods for hyper-redundant structures (structures with a high number of DOF) have been employed. The following subsections will elaborate on the different modelling techniques.

The Denavit-Hartenberg Convention

The Denavit-Hartenberg convention is a well established method for describing the position and orientation of the links of a robot manipulator with respect to a (usually fixed) base frame. Different solutions are presented to deal with the fact that the base is not fixed on a snake robot (Liljebäck *et al.*, 2005; Poi *et al.*, 1998).

Poi *et al.* (1998) present a snake robot that is made of 9 equal modules. Each module consists of seven revolute 1 DOF joints which are connected by links of equal length. Three joints and four joints have the axis of rotation perpendicular to the horizontal and vertical plane, respectively, and the joints are placed alternately. Each module is parameterised with the D-H convention. A modification to the convention has been proposed by placing the base coordinate system on the closest motionless link to the part of structure which is in motion. Hence, the links in motion are described in an inertial frame. The snake robot described in Poi *et al.* (1998) moves only four or five modules simultaneously, so giving the position and orientation relative to the closest motionless link prevents traversing through the complete structure to obtain positions and orientations in an inertial frame (such traversing is usually necessary when employing the D-H convention since we are dealing with minimal coordinates).

The motion patterns employed in Liljebäck *et al.* (2005), sidewinding and lateral undulation, are based on constant joint movement so we have to traverse through the whole structure to obtain the inertial position and orientation of each snake robot link, and hence the previously presented approach (Poi *et al.*, 1998) will not simplify the mathematical structure.

Therefore, a *virtual structure for orientation and position* (VSOP) is introduced to be able to describe the kinematics of the snake robot in an inertial reference frame. Liljebäck *et al.* (2005) present a snake robot with 5 revolute 2 DOF joints. The VSOP describes the position and orientation of the rearmost link of the snake robot relative to an inertial reference frame by 3 orthogonal prismatic joints and 3 orthogonal revolute joints, respectively. These virtual joints are connected by links with no mass. By employing the VSOP in the Denavit-Hartenberg convention, the position and orientation of each joint is given in an inertial coordinate system.

A Backbone Curve

Instead of starting by finding the position and orientation of each joint directly as with the Denavit-Hartenberg convention, a curve that describes the shape of the ‘spine’ of the snake robot can be employed (Chirikjian, 1992; Chirikjian and Burdick, 1991, 1994b; Yamada and Hirose, 2006b). The Frenet-Serret apparatus (see Do Carmo, 1976) is employed in the classical handling of the geometry of curves (Chirikjian and Burdick, 1994b). However, this approach has some limitations (Chirikjian and Burdick, 1994b). First, the Frenet-Serret frames assigned along the curve are not defined for straight line segments. Second, the vector function describing the spatial curve requires the numerical solution of a cumbersome differential equation. The introduction of backbone curves (see e.g. Chirikjian and Burdick, 1994b) is a way of handling these limitations. The backbone curve is defined as ‘a piecewise continuous curve that captures the important macroscopic geometric features of a hyper-redundant robot’ (Chirikjian and Burdick, 1994b) and it typically runs through the spine of the snake robot. A set of orthonormal reference frames are found along the backbone curve to specify the actual snake robot configuration. The backbone curve parameterisation together with an associated set of orthonormal reference frames is called a *backbone curve reference set* and allows for snake robots built from both prismatic and revolute joints (Chirikjian and Burdick, 1991).

A recent alternative approach to the methods presented by Chirikjian and Burdick (Chirikjian and Burdick, 1991, 1994b) has been given by Yamada and Hirose (2006b). This approach is called the *bellows model* and is specifically designed for distinguishing explicitly between the twisting and bending of the body of the snake robot. This is advantageous since most snake robots are designed with joints capable of bending, but not twisting (for example snake robots with cardan joints). Hence, the ability to twist can simply be left out of the model of the snake robot. So far, there are no

published results on how to fit the continuous bellows model to an actual snake robot (which has a discrete morphology).

The problem of determining joint angles of a robot manipulator given the end-effector position is called the *inverse kinematics problem*. For hyper-redundant manipulators (such as snake robots) this is a very computationally demanding task. When the backbone curve is employed, the problem is reduced to determining the proper time varying behaviour of the backbone reference set and this approach has been employed for controlling a real snake robot in Chirikjian and Burdick (1994a).

The above methods described in this section are suitable for abstraction, understanding and development of the geometric aspects of snake robot motion planning (which may initially be quite complicated due to the high number of DOF), in particular when the dynamics may be neglected.

Non-holonomic Constraints and Snake Robots with Passive Caster Wheels

The key to snake robot locomotion is to continuously change the shape of the robot. This is achieved by rotation and/or elongation of its joints. Krishnaprasad and Tsakiris (1994) and Ostrowski and Burdick (1996) both present kinematic approaches on how to link the changes in internal configuration to the net position change of the robot. The relation is found by utilizing non-holonomic constraints and differential geometry. Ostrowski and Burdick (1996) employ Hirose's Active Cord Mechanism Model 3 (ACM III) as an example which will be explained in the following. The first three pair of wheels of the ACM III are illustrated in Figure 2.3. The five joint angles $\phi_1, \phi_2, \phi_3, \psi_1$ and ψ_3 are controlled inputs and θ is the absolute angle of the middle link. The kinematic non-holonomic constraints are realized by adding passive caster wheels on the snake robot and may be written in the form

$$\dot{\bar{x}}_i \sin(\bar{\phi}_i) - \dot{\bar{y}}_i \cos(\bar{\phi}_i) = 0, \quad (2.1)$$

where $(\dot{\bar{x}}_i, \dot{\bar{y}}_i)$ is the translational velocity of the centre of axle i (corresponding the numbering of $\phi_i, i = 1, 2, \dots$) and $\bar{\phi}_i$ is the absolute angle (with respect to a horizontal axis) of the two wheels connected to axle i . More on non-holonomic systems can be found in Kolmanovsky and McClamroch (1995) and Bloch *et al.* (2003). The wheels are assumed not to slip and therefore realize an ideal version of the frictional properties of the snake skin as mentioned in Section 2.2.2.

A *local form* \mathbb{A} of a *connection* provides the relation between the shape

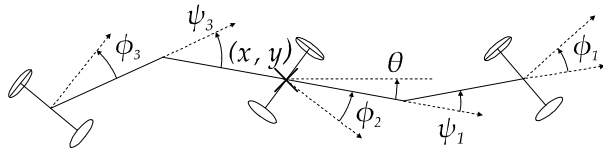


Figure 2.3: The first three links of the ACM III employed by Ostrowski and Burdick (1996).

changes of the snake robot and its net locomotion:

$$g^{-1}\dot{g} = -\mathbb{A}(r)\dot{r}, \quad (2.2)$$

where r is the shape variables and $g \in SE(2)$ gives the overall position and orientation of the snake robot (Ostrowski and Burdick, 1996). The connection provides understanding of how shape changes can generate locomotion and can even be used for controllability tests (Kelly and Murray, 1995; Ostrowski and Burdick, 1998). The simple form of (2.2) is dependent on the kinematic constraints breaking all the symmetries of the Lagrangian function which may raise dynamic constraints. This is achieved, with the ACM III as an example by using the first three segments to define the net motion of the snake robot. These segments define the path which is to be followed by the remaining segments due to the non-holonomic constraints on the wheels.

This modelling technique have also been used to include the dynamics, and this is described in Section 2.3.2.

2.3.2 Dynamics

The dynamics of the snake robots presented in the published literature has been derived by utilizing various modelling techniques such as the Newton-Euler formulation, Lagrange functions and geometric mechanics.

For snake robots without wheels, the friction between the snake robot and the ground affects the motion of the snake robot significantly. Thus, for these snake robots, the dynamics should be modelled for locomotion patterns such as lateral undulation. For snake robots with wheels, however, the wheels greatly reduce the friction, and hence, make it possible to utilize a purely kinematic model of the robot. The majority of results presented on modelling of the dynamics have therefore considered snake robots without wheels. In the following we will first give a short introduction to some of

the notation utilized below, then we give a brief overview of a selection of the results reported on the modelling of dynamics of wheeled snake robots and finally we present results on snake robots without wheels.

To ease the presentation of the mathematical models, a common notation for some of the material is presented, which is based in parts on Prautsch and Mita (1999) and Saito *et al.* (2002). Denote the mass m_i , length $2l_i$ and moment of inertia J_i for each link $i = 1, 2, \dots, n$. The snake robot moves in the xy -plane. Denote the angle θ_i between link i and the inertial (base) x -axis. Denote position of the centre of gravity (CG) of link i by (x_i, y_i) . Denote the unit vectors tangential $\mathbf{e}_t^{B_i} \in \mathbb{R}^2$ and normal $\mathbf{e}_n^{B_i} \in \mathbb{R}^2$ to the link i in the horizontal xy -plane. Hence, $\mathbf{e}_t^{B_i}$ points along link i and $\mathbf{e}_n^{B_i} \perp \mathbf{e}_t^{B_i}$. Denote the velocity $\mathbf{v}_i = [\dot{x}_i \ \dot{y}_i]^T \in \mathbb{R}^2$ of link i , and tangential and normal velocity of link i $\mathbf{v}_{i,t} = \mathbf{e}_t^{B_i} \left(\mathbf{e}_t^{B_i} \right)^T \mathbf{v}_i$ and $\mathbf{v}_{i,n} = \mathbf{e}_n^{B_i} \left(\mathbf{e}_n^{B_i} \right)^T \mathbf{v}_i$, respectively.

The friction forces that act on the CG of link i are denoted $\mathbf{f}_i = [f_{x_i} \ f_{y_i}]^T \in \mathbb{R}^2$ where f_{x_i} , f_{y_i} are friction forces between link i and the ground along the x - and y -direction of the inertial frame, respectively. The coefficients of friction tangential and normal to link i are $c_{i,t}^{(j)}$ and $c_{i,n}^{(j)}$, respectively, where j is used in this chapter to distinguish between the coefficients in the various friction models.

Snake Robots with Passive Caster Wheels

A desired property for moving by lateral undulation, is to keep the difference between lateral and longitudinal friction as high as possible. This property of friction can be obtained by attaching caster wheels to the belly of the snake robot. The equations of motion of a simplified version of the ACM III snake robot used by Hirose (1993) are presented by Prautsch and Mita (1999). The robot is the same as the ACM III shown in Figure 1.2 and Figure 2.3 except that the wheel axles are fixed. The dynamic model is derived to utilize acceleration-based control algorithms. It is assumed that the wheels do not slip sideways.

A snake robot (called the SR#2) has been presented and compared to the ACM III by Wiriyacharoensunthorn and Laowattana (2002). The Active Cord Mechanism (ACM) modelling assumes that the wheels do not slip. This non-slippage introduces non-holonomic constraints. The SR#2 model is based on *holonomic* framework and is hence without the no-slip condition. The argument used against assuming no slip is that it is difficult

to control the torques in the joints such that the assumption is satisfied. Simulations show the ACM III build up an error in position while following a circular path. This is not the case for SR#2, something which makes it a more accurate model for this scenario.

The above models have all described planar motion. Moreover, a 3D model of the dynamics of a snake robot with wheels that do not slip has been presented by Ma *et al.* (2004). In addition, a system equation for control of the height of the wheels is given and computer simulations are presented in that paper.

Snake Robots without Wheels

The use of wheels may decrease traversability (Saito *et al.*, 2002), thus wheel-less robots may have an advantage. As discussed earlier, friction plays a significant role for wheel-less snake robots, hence it is necessary to model the dynamics and not only the kinematics for relatively high speed motion (for certain motion patterns during low speed motion, a kinematic model may suffice). First, we will give an overview of the friction and contact models employed for snake robots, then a selection of dynamic models derived for snake robots without wheels will be presented.

Friction and Contact Models The friction models presented in literature on snake robots are based on a Coulomb or viscous-like friction model and such models are explained, for instance, in Egeland and Gravdahl (2002). For 3D models of snake robots, it is necessary to model the normal contact force due to impacts and sustained contact with the ground, in addition to the friction force. This force has been described as compliant by a spring-damper model in Liljebäck *et al.* (2005) as

$$f_{N_i} = \begin{cases} 0 & , z_i \geq 0 \\ -k \cdot z_i - d \cdot \dot{z}_i & , z_i < 0, \end{cases} \quad (2.3)$$

where $z_i \in \mathbb{R}$ is the height of the centre of gravity of link i , $\dot{z}_i = \frac{dz}{dt}$, $k \in \mathbb{R}^+$ is the constant spring coefficient of the ground, and $d \in \mathbb{R}^+$ is a constant damping coefficient that serves to damp out the oscillations induced by the spring. The friction force on link i , based on a simple, viscous-like model, is now written as

$$\mathbf{f}_i = -c_{i,t}^{(1)} |f_{N_i}| \mathbf{v}_{i,t} - c_{i,n}^{(1)} |f_{N_i}| \mathbf{v}_{i,n} \in \mathbb{R}^2. \quad (2.4)$$

The sum of forces acting on link i in the model presented by Liljebäck *et al.* (2005) is $\mathbf{f}_i^{3D} = [\mathbf{f}_i^T \quad f_{N_i}]^T \in \mathbb{R}^3$. The spring coefficient k needs to be

set very high to imitate a solid surface. Hence, the total system is stiff and requires a very small simulation step size to be simulated. However, the constitutive law (2.3) for the normal force provides an intuitive and simple approach to implementing the normal force. A friction model including both static and dynamic friction properties for a 3D dynamic model is given by Ma *et al.* (2004).

The 2D anisotropic viscous friction model used by Grabec (2002) can be derived from (2.4) by setting $f_{N_i} \equiv 1$. In this case the friction force is found from

$$\mathbf{f}_i = \mathbf{H}_i \mathbf{v}_i, \quad (2.5)$$

where

$$\mathbf{H}_i = c_{i,n}^{(2)} \left[\left(1 - \frac{c_{i,t}^{(2)}}{c_{i,n}^{(2)}} \right) \mathbf{e}_t^{B_i} \left(\mathbf{e}_t^{B_i} \right)^T - \mathbf{I}_{2 \times 2} \right], \quad (2.6)$$

and $\mathbf{I}_{2 \times 2} \in \mathbb{R}^{2 \times 2}$ is a unit matrix.

The effect of rotational motion of the links is introduced in the two 2D friction models, one with viscous and one with Coulomb friction, presented by Saito *et al.* (2002). Both models are derived by integrating the infinitesimal friction forces on a link. The translational part of the viscous friction model is given by (2.4) with $f_{N_i} = m_i$ (i.e. f_{N_i} is not an actual force). The total viscous friction torque due to rotational velocity around the centre of gravity of link i is found to be

$$\tau_i = -c_{i,n}^{(3)} \frac{m_i l_i^2}{3} \dot{\theta}_i \in \mathbb{R}. \quad (2.7)$$

For translational motion the friction force based on Coulomb's law is found for $\dot{\theta}_i = 0$ as (Saito *et al.*, 2002)

$$\mathbf{f}_i = -m_i g \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix} \begin{bmatrix} c_{i,t}^{(3)} & 0 \\ 0 & c_{i,n}^{(3)} \end{bmatrix} \text{sign} \left(\begin{bmatrix} \left(\mathbf{e}_t^{B_i} \right)^T \mathbf{v}_i \\ \left(\mathbf{e}_n^{B_i} \right)^T \mathbf{v}_i \end{bmatrix} \right). \quad (2.8)$$

The expression for the Coulomb friction force is slightly different for $\dot{\theta}_i \neq 0$ (Saito *et al.*, 2002), however we only include the case when $\dot{\theta}_i = 0$ here to simplify the presentation. Employing Coulomb's law of dry friction as the friction model results in a more complicated, but also a more accurate model for motion on non-lubricated surfaces. The viscous-like friction model (2.4) does not include dry friction and thus the relative high friction forces which may arise at low velocities are not modelled.

For most of the gaits simulated with the above friction models, the property $c_{i,t} < c_{i,n}$ has been implemented to realize the anisotropic friction property of a snake moving by lateral undulation. It may be difficult to design a snake robot with $c_{i,t} < c_{i,n}$ on a general surface. Sidewinding has been implemented with an isotropic friction model ($c_{i,n} = c_{i,t}$) by Liljebäck (2004) and as a purely kinematic case by Burdick *et al.* (1993). Special gaits for planar motion based on an isotropic friction model are detailed by Chernousko (2005, 2000).

Decoupled Dynamic Model A five link snake robot with 1 DOF joints is modelled and controlled in Saito *et al.* (2002). The robot is built and experiments performed to validate the theoretical results. Four parallel metal skates are put on the underside of each link to implement the anisotropic friction property $c_{t_i} < c_{n_i}$.

The dynamic model of the snake robot is developed from the Newton-Euler equations resulting in two sets of equations: one for translational motion of the centre of mass \mathbf{w} of the snake robot and another for the rotational motion of each link given in an inertial frame. The final equations of motion can be decoupled into two parts: shape motion and inertial locomotion. The shape motion maps the joint torques to joint angles while the inertial locomotion relates the joint angles to the inertial position and orientation of the snake robot. This simplifies the analysis and synthesis of locomotion of the snake robot. To achieve decoupling, a vector of relative angles $\phi \in \mathbb{R}^{n-1}$ where the i -th element of ϕ is $\phi_i = \theta_i - \theta_{i+1}$, and a quantity $\dot{\psi} \in \mathbb{R}$ which can be thought of as ‘an average angular momentum’ are introduced (Saito *et al.*, 2002). The expressions for shape motion and inertial motion, respectively, are found to be

$$\mathbf{h}_s(\ddot{\phi}, \theta, \dot{\theta}, \dot{\mathbf{w}}) = \mathcal{B}\mathbf{u}, \quad (2.9)$$

$$\mathbf{h}_i(\dot{\psi}, \ddot{\psi}, \theta, \dot{\mathbf{w}}, \ddot{\mathbf{w}}, \dot{\phi}) = 0, \quad (2.10)$$

where $\mathbf{h}_s(\cdot)$, $\mathbf{h}_i(\cdot) \in \mathbb{R}^n$ are certain functions, $\theta = [\theta_1 \ \dots \ \theta_n]^T$, \mathbf{u} are the joint torques and \mathcal{B} is an invertible matrix. Control of the snake robot is now performed in two steps: First, the joint torques \mathbf{u} control the shape of the robot and, second, the relative angles ϕ control the average angular momentum $\dot{\psi}$ and position \mathbf{w} . For someone who needs a 2D model of a snake robot and has a basic knowledge of classical mechanics, this is probably the easiest 2D model to implement for simulation due to the concise and comprehensive presentation of the model in the paper.

Quasi-Stationary Equations of Motion A 2D model based on the Newton-Euler formulation of a snake robot with 1 DOF revolute joints with the viscous friction model (2.5) is presented by Grabec (2002).

Non-dimensional variables are introduced to simulate the dynamics of the snake robot. The resulting system of second order non-linear equations, which constitute the non-dimensional model of the snake robot, may become unstable during simulation. To aid the numerical treatment, over-critical damping is introduced by setting accelerations to zero. The result is a set of ‘quasi-stationary’ first-order differential equations of motion. By employing the first-order equation for translational motion together with the friction model in short form (2.5) the velocity of the head of the snake robot is found to be

$$\mathbf{v}_{head} = - \left(\sum_{i=1}^n \mathbf{H}_i \right)^{-1} \sum_{i=1}^n \mathbf{H}_i \mathbf{v}_i^{(rel)}, \quad (2.11)$$

where $\mathbf{v}_i^{(rel)}$ is the velocity of link i with respect to the head and \mathbf{H}_i is found from (2.6). Saito *et al.* (2002) give the relationship between shape changes from joint angle deflection and the position of the CG of the snake robot (2.10). To investigate locomotion analytically, (2.11) offers an alternative approach where the direct connection between velocities of each link relative to head of the snake robot and the head velocity is given.

Creeping on an Inclined Plane A model of a snake robot with n links and 1 DOF rotational joints has been developed from the Newton-Euler equations by Ma (2001). The actual snake robot that is modelled has wheels. However, the friction between the underside of the snake robot and the ground surface is modelled as anisotropic Coulomb friction. Hence, the wheels do not constitute non-holonomic constraints (i.e. the wheels may slip) and that is why we have included the model in this section. The model of planar motion of the snake robot is extended to motion on an inclined plane where the angle of inclination effects the motion of the snake robot in Ma *et al.* (2003a, 2001b).

The mathematical model is presented in two ways (both for planar motion and the motion on an inclined plane). The first alternative is to write the model in a form where it is assumed that the joint angles together with the joint angular velocities and accelerations are given (this form is later employed for something called shape based control). From the specified data, the rotational and translational accelerations of the first link can then be found from the model. Thus, the motion of the snake robot in the plane is found. Moreover, the joint torques necessary to move the joint in

a predetermined way can be found from the model. Hence, it is possible to study the joint torques and e.g. how they change for a specified motion pattern for various friction scenarios.

The second alternative is the most common for snake robot models: how does the snake robot move given the commanded joint torques? By specifying the joint torques, the link angle accelerations are found. Then, the translational and rotational acceleration of the first link can be found and the necessary velocities and positions are found by integration.

Simulation results are given for both shape based and torque based control of the snake robot.

The Lagrangian Research on robots that resemble snakes is not only limited to land-based locomotion. Papers regarding anguilliform (eel-like) locomotion have also been published (Ayers *et al.*, 2000; McIsaac and Ostrowski, 1999, 2000, 2003b). A five link 2D snake robot (called the REEL II) with 1 DOF revolute joints, which will be used as an example in the following, is modelled and experimented with in McIsaac and Ostrowski (2003b). Motion planning for such a robot consists of first building up the momentum to the snake robot and then steering the robot to its desired location. Hence, it is convenient that the mathematical model includes an explicit expression for the momentum. The model is formulated from the Lagrangian of the system and is summarized in the following.

The fact that the energy of the system and the frictional forces acting on the system are invariant with respect to the position and orientation of the snake robot (the system exhibits Lie groups symmetries) is exploited to simplify the mathematical model. The assumption that the joint angles are controlled directly (the same as saying that the dynamics (2.9) is ignored) yields two sets of resulting equations. The first equation relates the velocity of the snake robot to its internal shape changes and is similar to (2.2) given in Section 2.3.1 except for the locked inertia tensor $\mathbb{I}(r)$ and generalized momentum vector p which have been added (we have a case of mixed constraints with both kinematic and dynamic constraints). The dynamics of the system is described by the generalized momentum equation which is the second set of resulting equations. The generalized momentum p is associated with the momentum along the directions allowed by the kinematic constraints. A thorough explanation of the equations is found in Bloch *et al.* (1995).

The Newton-Euler Algorithm A mathematical model of a snake robot with five 2 DOF revolute joints is presented by Liljebäck *et al.* (2005). A snake robot is also constructed for experiments. In addition to the actual snake robot, a *virtual structure of orientation and position* (VSOP, see Section 2.3.1) is included in the dynamic model. The VSOP together with the snake robot have generalized minimal coordinates $\mathbf{q} \in \mathbb{R}^{2(n-1)+6}$ and generalised forces $\boldsymbol{\tau} \in \mathbb{R}^{2(n-1)+6}$. The Newton-Euler formulation and the VSOP perspective is employed and the dynamic model is written as

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{ext}, \quad (2.12)$$

where \mathbf{M} is the mass and inertia matrix, \mathbf{C} is the Coriolis and centripetal matrix, $\mathbf{g}(\cdot)$ is the vector of gravitational forces and torques and $\boldsymbol{\tau}_{ext}$ is the vector including the external forces (friction and normal contact force). The matrices are detailed in Liljebäck (2004). The Newton-Euler algorithm has been employed to simulate the snake robot. Hence, the full analytical expressions for the system matrices do not need to be found explicitly. Instead, the necessary matrices and accelerations are found numerically with the recursive Newton-Euler algorithm. This is advantageous since the analytical expressions for the system matrices are extremely large for a large number of joints when minimal coordinates are employed. For a model with non-minimal coordinates found in Ma *et al.* (2004), analytical expressions for the joint torques and head configuration of a 3D snake robot model deducted from the Newton-Euler equations are shown.

The Lagrange and the Newton-Euler method are similar for rigid body dynamics in that the expression obtained by the Lagrange method is found by running through the Newton-Euler algorithm once. Since the Newton-Euler algorithm deals with the mathematical model as a recursive algorithm, it is a more efficient framework for simulation than the Lagrange method for large models (Egeland and Gravdahl, 2002).

A modification of the Newton-Euler algorithm has been presented by Boyer *et al.* (2006) to numerically evaluate a model of a *continuous* 3D underwater snake robot where the modelling approach is based on beam theory.

2.4 Snake Robot Locomotion

A variety of approaches on how to make a snake robot move has been proposed. In most of the motion patterns or ‘gaits’ used for locomotion, we find a distinct resemblance to the undulating locomotion of biological snakes

Gait	With passive wheels	Without wheels
Concertina		Shan and Koren (1993)
Lateral undulation	Ma (2001); Ma <i>et al.</i> (2003 <i>a,b</i>); Mori and Hirose (2002); Ohno and Hirose (2001); Ostrowski and Burdick (1996); Prautsch and Mita (1999); Wiriyacharoen-sunthorn and Laowattana (2002); Ye <i>et al.</i> (2004 <i>a</i>)	Bayraktaroglu <i>et al.</i> (1999); Grabec (2002); Liljebäck <i>et al.</i> (2005); Ma (1999); McIsaac and Ostrowski (1999, 2003 <i>b</i>); Saito <i>et al.</i> (2002)
Sidewinding		Burdick <i>et al.</i> (1993); Liljebäck <i>et al.</i> (2005)
Inchworm/ Caterpillar		Chirikjian and Burdick (1995); Ohno and Hirose (2001); Poi <i>et al.</i> (1998); Rincon and Sotelo (2003)
Climbing		Nilsson (1997 <i>b</i> , 1998)
Lateral rolling	Mori and Hirose (2002); Togawa <i>et al.</i> (2000); Yamada <i>et al.</i> (2005)	Chen <i>et al.</i> (2004); Dowling (1999); Ohno and Hirose (2001)

Table 2.1: Overview of gaits implemented for snake robots.

or worms. However, the motion patterns may be altered to compensate for the fact that the snake robots do not have the exact same anatomy as biological snakes, inchworms or caterpillars. For example, the snake robots are not as articulated as their biological counterpart, and this reduces movability. In addition, snakes use among other their skin to sense their immediate vicinity. Such a fine grid of sensors is difficult to implement on a snake robot.

Early studies of snake locomotion were given by Gray (1946). Later, a mathematical description of the serpentine motion of snakes was presented by Hirose (see e.g. Hirose, 1993). An overview of the most frequent gaits which have been implemented for snake robots is found in Table 2.1 and we see that lateral undulation is the most common motion pattern.

A description of a number of different motion patterns both for planar and 3D motion is given in the following.

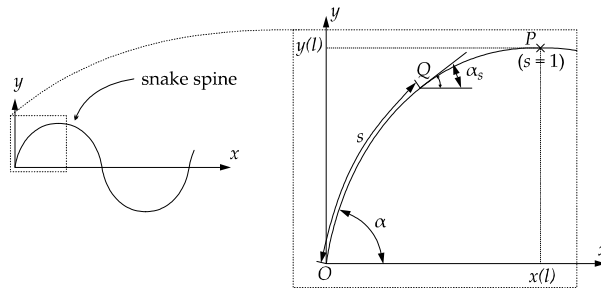


Figure 2.4: The serpenoid curve in Hirose (1993). By permission of Oxford University Press.

2.4.1 Planar Snake Robot Locomotion

Planar snake robot locomotion is usually performed with the motion pattern lateral undulation. First, we will present how to describe the motion pattern for a snake robot. Then some examples will be given together with other gaits for planar locomotion.

The Serpenoid Curve

A biological snake that moves by lateral undulation across a uniform surface displays a periodic creeping motion. The shape of the snake during this kind of locomotion has been studied by Hirose (see e.g. Hirose, 1993), and he has presented the ‘serpenoid curve’ as a way of describing the form of the gliding motion of a snake. The serpenoid curve is interesting for snake robot locomotion since the snake robot can move forward by moving its joints such that its body follows the trace of the serpenoid curve. The curve ensures that the curvature changes smoothly along the body, which is natural considering the contraction and relaxation of the muscles in a snake during locomotion (Hirose, 1993).

The serpenoid curve is illustrated in Figure 2.4 where s is the distance along the curve, l is the length of one quarter period of the curve and $\alpha_s(s, l)$ is the ‘winding angle’ along the curve. Denote the tangential $c_{i,t}$ and normal $c_{i,n}$ frictional coefficient between link i and the ground. The winding angle α_s for locomotion of a particular snake robot is determined by factors such as the link length, the bending angles between adjacent links and the ratio $c_{i,t}/c_{i,n}$ where the ratio also gives a lower bound for α_s . The winding angle α_s at an arbitrary point Q on the serpenoid curve is

found from

$$\alpha_s(s, l) = \alpha \cos\left(\frac{\pi}{2l}s\right), \quad (2.13)$$

where α is the initial winding angle as illustrated in Figure 2.4. Unless otherwise specified, the common assumption is $c_{i,t} < c_{i,n}$ which is necessary to move forward on a flat surface by lateral undulation. However, it needs to be mentioned that a snake robot will instead move slowly *backward* for $c_{i,t} = c_{i,n}$ (Nilsson, 2004).

An alternative to the serpenoid curve has been proposed by Ma (1999) and is called the serpentine curve. Ma (1999) has shown that a snake robot following this curve has higher locomotive efficiency. For a snake robot that does not slip, the locomotive efficiency is defined as the ratio between the forces that act along the snake robot and the forces that act perpendicular to the body (Ma, 1999). The serpenoid curve is nevertheless the most frequent employed basis for finding the desired joint angles for a snake robot. This is probably partly due to its simplicity compared to the expression for the serpentine curve.

Lateral Undulation for Snake Robots

The serpenoid curve cannot be exactly reproduced by a snake robot due to its discrete morphology. However, an approximation to the serpenoid curve can be recreated by a snake robot by setting its (relative) joint reference angle $\phi_{h_i,d}$ for joints $i = 1, \dots, n - 1$ as

$$\phi_{h_i,d} = A_h \sin(\omega_h t + (i - 1)\beta_h) + \gamma_h, \quad (2.14)$$

where A_h is the maximum amplitude of oscillation, β_h is the phase shift between adjacent links, γ_h determines the orientation of the snake robot and ω_h is the angular frequency of oscillation (see e.g. Saito *et al.*, 2002; Ye *et al.*, 2004a). The use of γ_h to change the heading of the snake is discussed in Ye *et al.* (2004a) and two alternative turning motions are also presented. The parameters ω_h and β_h are the two most significant parameters in order to control the speed of the serpentine wave that propagates down the body of the snake robot.

Sideways slip should be kept as low as possible when moving by lateral undulation. This is because such slip does not contribute to forward motion. For wheel-less snake robots, sideways slip can be reduced by changing the motion pattern parameters in (2.14) according to the ratio $c_{i,t}/c_{i,n}$. An example on how to do this has been presented by Saito *et al.* (2002), where the desired relative joint angles are found from (2.14) and the corresponding

experimental results are presented. In addition, Saito *et al.* (2002) show how to control the speed and heading of the snake robot to desired values by altering ω_h and γ_h in (2.14), respectively.

Another possibility to reduce sideways slip is to mechanically alter the snake robot by adding passive caster wheels to the belly. For the snake robot ACM III (see Figure 1.2 and Figure 2.3) used as an example by Ostrowski and Burdick (1996) a particular approach is employed to avoid wheel-slip while moving forward. To this end, the angles between adjoining links (ψ_1 and ψ_3) and the angles between the wheel axles and the links (ϕ_1 , ϕ_2 and ϕ_3) are found from expressions similar to (2.14) where β_h is 90 degrees larger for the joint angles than for the axle angles. A model similar to the ACM III, but with fixed wheel axles, is employed by Prautsch and Mita (1999). There a Lyapunov-based approach for position control of the head of the snake robot is proposed. In addition, it is shown how to minimize the control torques required for serpentine motion since low control torques reduce the risk of wheel-slip. Date *et al.* (2000) introduce an expression for the constraint forces caused by the wheels. In addition, a control approach based on dynamic manipulability is proposed to avoid sideways slip in the same paper. A control strategy to avoid wheel-slip is also presented in Date *et al.* (2001). The latter approach uses an automatic generation of the joint reference signals based on a velocity reference for the head of the snake robot while keeping a winding shape of the robot. The results given by Date *et al.* (2001) are extensions of the methods presented by Prautsch and Mita (1999) where the snake robot tended to a singular (straight) posture when the number of links was increased (Date *et al.*, 2000).

Results on controllability of snake robots (and other locomotion systems) are given by Ostrowski and Burdick (1998) and Kelly and Murray (1995).

McIsaac and Ostrowski (McIsaac and Ostrowski, 2000, 2003*a,b*) have derived and implemented closed-loop heading control based on image-based position feedback for an underwater snake robot, and they have presented a controller for stopping the underwater snake robot during forward and circular motion. The motion pattern for underwater locomotion is similar to lateral undulation.

Ma *et al.* (2003*a*) and Ma and Tadokoro (2006) study lateral undulation on an inclined surface. It is found that for the snake robot to be able to move up the plane, the upper limit for the initial winding angle α should be decreased and the lower limit should be increased for an increasing inclination angle of the plane.

While most of the approaches to locomotion presented above rely on the transversal/longitudinal friction property, others have explored alternative ways to locomote with either the friction model being isotropic ($c_{i,t} = c_{i,n}$) or a purely kinematic model being used. Biological snakes exploit obstacles such as rocks or contours on the ground to aid locomotion during lateral undulation as mentioned in Section 2.2.3. This form of locomotion has also been investigated for snake robots (Bayraktaroglu *et al.*, 2006; Bayraktaroglu and Blazevic, 2005; Hirose, 1993; Hirose and Umetani, 1976). A pioneering work on such locomotion has been presented by Hirose and Umetani (1976), where it is demonstrated how to utilize walls and large cylinders to generate propulsive forces for a wheeled snake robot. Hirose (1993) analyses the gliding configuration of a snake negotiating a corner within a maze. In addition, he presents a relationship between the angle of the corner, the friction coefficient between the snake and the walls of the maze and the distance between the contact points for contact between the snake and the maze. These analytical results can be employed to determine how a snake robot should move within a similar maze. Shan and Koren (1993) present an approach to locomotion where a snake robot is allowed to come in contact with obstacles and it is discussed how a snake robot can continue to be mobile while in contact with an obstacle. In this publication the links of the snake robot presented have ball casters to slide on and solenoids to push into the ground surface to increase friction when needed. With this snake robot, the change in number of degrees of freedom of each link while in contact with the obstacles is investigated and the obstacles are used as constraints so that the snake robot is able to slide along them. More recently, Bayraktaroglu and Blazevic (2005) have presented a wheel-less snake robot that employs push-points, such as pegs, to create the total propulsive force during a form of lateral undulation. Here, the ratio between lateral and longitudinal friction forces is disregarded, and this makes it possible to build and develop motion patterns for snake robots without the friction property $c_{i,t} < c_{i,n}$ and hence without wheels. The joints are not bent to push against the pegs; instead, the snake robot in Bayraktaroglu and Blazevic (2005) has linear actuators on each side of the links that push out from the link. This work has been extended by Bayraktaroglu *et al.* (2006) where the linear actuators are no longer necessary. Instead, the wheel-less snake robot localizes push-points with on/off switch sensors along the sides of its body. The snake robot then calculates a spline between two adjacent push-points and the joint angles are controlled so that its body is fitted to the spline and thus uses the push-points to be able to move

forward. It is assumed that there is always a push-point available within a reasonable proximity of the head of the snake robot. Experimental results are presented for this approach. The serpenoid curve has not been employed directly in the papers by Bayraktaroglu *et al.* (Bayraktaroglu *et al.*, 2006; Bayraktaroglu and Blazevic, 2005) due to the change of shape according to the placement of the push-points, however, the similarities to the shape of a biological snake moving by lateral undulation while using contours on the ground to push against are apparent.

It is important that a snake robot has sensors to detect contours on the ground that can be pushed against for faster locomotion. In addition, sensors can be used to make a map of the environment of the snake robot on-line in order to find a feasible path. This is shown in Choset and Lee (2001) and Henning *et al.* (1998). Moreover, sensor-based following of a light source is shown in Worst and Linnemann (1996).

Alternative Approaches to Locomotion

Planar locomotion for a 2-link, 3-link and a multi-link system with 1 DOF revolute joints (i.e. rotational joints) is studied by Chernousko (2005, 2000). In those papers, dry friction forces between the links and the ground are assumed. A combination of fast and slow movements of the joints is used to move the 2- and 3-link system to any point in the plane. Forward motion of the multi-link system is obtained by propagating a single wave consisting of 3 or 4 links from the tail to the front of the robot. Very small velocities and accelerations are necessary for such locomotion which depends on the stick-slip phases of dry friction. Subsequently, locomotion is obtained as long as the moving part of the snake robot is not subjected to a friction force larger than the non-moving part is able to counteract by lying still. This principle is similar the concertina locomotion (see Section 2.2.3) except that the snake robot does not anchor its body by pushing against something other than the flat ground surface such as, for example, the walls of a narrow passage.

2.4.2 3D Snake Robot Locomotion

For snake robots with 2 DOF revolute joints and capable of 3D motion, a second reference signal is needed to control the vertical wave (i.e. the lifting and lowering of the links). The vertical joint reference signal can be written as (Liljebäck *et al.*, 2005):

$$\phi_{v_i,d} = A_v \sin(\omega_v t + (i - 1)\beta_v + \beta_0), \quad (2.15)$$

where $\phi_{v_i,d}$ is the relative angle of joint i that controls the lift of the adjacent links. The phase difference between the horizontal and vertical wave is given by β_0 and the remaining parameters A_v , ω_v and β_v give the amplitude, angular frequency and phase-shift similar to the parameters in (2.14). The expressions (2.14)-(2.15) will be employed throughout this thesis to implement various motion patterns.

Sinus-lifting is also implemented for snake robots (Hirose and Mori, 2004; Ma *et al.*, 2004; Mori and Hirose, 2002). A 2D model incorporating a ground contact force as a function of the curvature of the snake body is used by Ma *et al.* (2004). It is shown that the snake robot moves forward faster by sinus-lifting than by lateral undulation. In addition, sinus-lifting is a way of reducing the risk of side-slip of the wheels. A control method for determining which links to lift based on a kinematic model has been proposed by Ohmameuda and Ma (2002). A wheeled snake robot able to move in 3D is presented by Ma *et al.* (2003b) and it is shown that for the robot to be controllable and observable, it requires that $4 \leq m \leq n - 2$ where m is the number of pairs of wheels touching the ground simultaneously. Sinus-lifting, like lateral undulation, is dependent on an anisotropic friction property ($c_{t,i} < c_{n,i}$) for efficient locomotion. However, there are also 3D motion patterns for moving with isotropic friction, and these will be elaborated on in the following three paragraphs.

Sidewinding was implemented with $c_{t,i} = c_{n,i}$ by Liljebäck (2004) using (2.14) and (2.15) to find the desired joint angles, and it was found that both γ_h and β_0 control the direction of locomotion. A slightly different approach to sidewinding locomotion is developed by Burdick *et al.* (1993) for 3D motion on a flat surface. In this latter work the snake robot lays its stationary parts flat on the ground while the moving parts are curved above the ground (see Section 2.2.3 for how sidewinding is performed). This results in a more natural gait compared to a sinus-like curve for the shape of a snake robot which is obtained by employing (2.14) and (2.15) to find the desired joint angles. The use of such a sinus-like curve results in the fact that the snake robot only touches the ground surface at isolated points when the joint angles are accurately controlled. Thus, the snake robot can more easily fall over since the contact surface with the ground is small. Sidewinding has also been performed without an explicit mathematical expression for the shape or joint angles such as e.g. (2.14)-(2.15). One such approach to develop a form of sidewinding uses genetic programming (Tanev *et al.*, 2005). Hence, the joint movement of the snake robot is determined based on some desired criteria. Moreover, the simulation results presented in Tanev

et al. (2005) demonstrate that the snake robot manages to move sideways despite several damaged joints. Such robustness to mechanical failure is important since the snake robot consists of many joints and it is probable that one or more of them may fail during challenging operations.

Inchworm-like and caterpillar-like motion patterns are shown with (Chirikjian and Burdick, 1995) and without (Chirikjian and Burdick, 1995; Poi *et al.*, 1998; Rincon and Sotelo, 2003) extensible joints (see Figure 2.5 (b) for the latter case). These gaits have been modelled by only considering the kinematics (and not the dynamics) of the snake robot. The model is nevertheless valid since it is assumed that the snake robot is moving slowly. While some results (Chirikjian and Burdick, 1995; Poi *et al.*, 1998) rely on slow speeds or ratchet wheels to avoid slipping, water can also be pumped between the links to add weight to the parts that are not moving (Rincon and Sotelo, 2003). This way, the ground contact force and hence the friction, on the moving parts are reduced.

Various kinds of rolling motion have also been proposed. These motion patterns are not directly similar to movement of biological snakes. Lateral rolling is a sub-group of these motion patterns, and one way of implementing lateral rolling consists of moving all the joints in phase while keeping the snake robot in an U-shape on the ground as depicted in Figure 2.5 (a) (Dowling, 1999). The reference joint angles for lateral rolling can be found from (2.14) and (2.15) by setting $A_h = A_v$, $\omega_h = \omega_v$, $\beta_h = \beta_v$ and $\beta_0 = \frac{\pi}{2}$ (Ohno and Hirose, 2001). Experimental results for this gait are given by e.g. Mori and Hirose (2002) and Hirose and Mori (2004). U-shaped lateral rolling is mainly performed with all links in contact with the ground surface. However, 2 DOF (cardan) joints are still needed so the snake robot has to be capable of 3D motion (the robot can also be equipped with alternating perpendicular 1 DOF joints instead of 2 DOF joints). Other forms of lateral rolling are described in Chen *et al.* (2004). Another way of moving proposed in the same paper is called the ‘smoke ring’ according to Dowling (1999) and is depicted in Figure 2.5 (e). The snake robot then shapes its body into a ring and subsequently curls around a vertical or horizontal pipe-like structure in order to ‘roll’ along the pipe. The snake robot can also move like a wheel as shown in Figure 2.5 (c), but it is challenging to keep the snake robot from falling over while moving. The shape of the snake robot is described analytically for several motion patterns based on rolling (or ‘twisting’) by Ye *et al.* (2004b). In the latter paper sinus-lifting and other common serpentine motion patterns have been combined with a twisting motion. Moreover, several of these motion patterns are tested

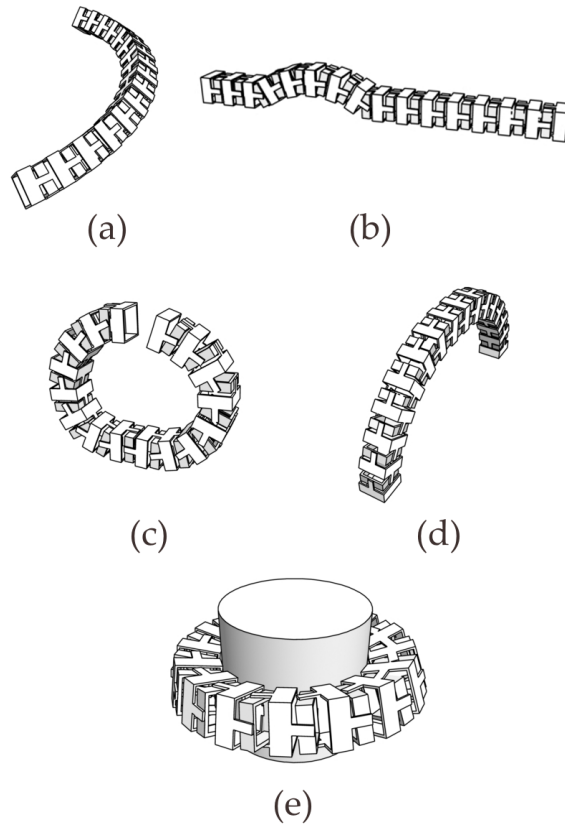


Figure 2.5: Various motion patterns implemented for snake robots: (a) U-shaped lateral rolling (moving towards the right), (b) travelling wave/caterpillar locomotion (moving the same way as the travelling wave, i.e. left or right), (c) the wheel (moving right or left), (d) bridge mode (moving right or left), and (e) 'smoke ring' (moving up- or downward).

by experiments and all tests show a mainly sideways motion of the snake robot. Erkmen *et al.* (2002) have also elaborated on the twisting mode of locomotion where an approach on how to make the snake robot walk using its two ends as feet is described. This form of locomotion is called ‘bridge mode’ and is illustrated in Figure 2.5 (d). In addition, a genetic algorithm is presented that handles how to move the snake robot from lying flat on the ground to standing on its two ends without falling over. Descriptive figures and explanations of the several gaits mentioned above, in addition to other motion patterns, are given by Dowling (1999), Ohno and Hirose (2001) and Mori and Hirose (2002). Moreover, movies of various snake robots during locomotion can be found on the web (Choset, 2007; Miller, 2007).

A snake robot, called the ACM-R5, with fins with wheels on its link and thus capable of both swimming and land-based locomotion is presented by Yamada *et al.* (2005). On the ground the snake robot moves by lateral rolling and lateral undulation. In the water it can move up and down, swim forward and turn. The details on the motion patterns are said to be reported at a later stage.

A snake robot needs to lift its body when, for example, climbing or moving over high obstacles. However, the joint actuators can only produce a limited torque. Hence, the snake robot links should be lifted in some ‘intelligent’ manner to keep the necessary joint torques as low as possible. To this end, Nilsson (1997*b*, 1998) has developed an algorithm for lifting several links with limited joint torque. In addition, the snake robot may utilize poles or other objects for climbing and a variant of the ‘smoke ring’ in Figure 2.5 (e) has been proposed by Nilsson (1997*a*).

2.5 Discussion and Summary

Research on snake robots has increased the past ten to fifteen years, but there are still many challenges to face both on modelling and control of snake robots in order to make them able to locomote intelligently through unknown terrain. In this chapter we have seen a variety of approaches to modelling and locomotion of snake robots and these will be summarized in the following.

Since snake robots have a high number of degrees of freedom, careful consideration has to be taken to decide what model to use for a snake robot depending on which aspect of snake robot locomotion is being studied. A planar and purely kinematic model is sometimes employed to describe lateral undulation for a snake robot with passive caster wheels that do not

slip. The use of a purely kinematic model simplifies both the model and the analysis of locomotion, and factors that contribute to locomotion are more easily highlighted. 3D models of the kinematics of snake robots without wheels are also developed and they are employed to study motion patterns which also requires lifting and lowering (in addition to sideways motion) of the snake robot links. The kinematic description of snake robot locomotion without wheels is justified by assuming low velocities and sometimes also certain friction properties (such as zero friction while gliding forward, but infinite friction when pushed backward).

Although passive wheels help to justify the non-slip assumption (on the wheels) of some kinematic models, it can still be difficult to control the joint torques such that the wheels do not slip. Moreover, locomotion of snake robots at high velocities require that the friction between the underside of the snake robot and the ground surface is considered. Hence, there is a need for models that include the dynamics of snake robot locomotion. 2D dynamical models are mainly used to describe lateral undulation with side-slip (i.e. without the idealized assumption of wheels that do not slip, or with no wheels at all). The friction force between the snake robot and the ground surface is described as viscous-like friction or Coulomb friction, and Coulomb friction is usually modelled with sign-functions. However, such a sign-function does not allow for a non-zero friction force at zero velocity (the stick-phase). Hence, a snake robot lying on an inclined plane will slide slowly downward. The directional friction property between the belly of the snake robot can not always be realized due to varying ground conditions. To this end, simulation results of a planar snake robot moving by a promising scheme which consists of the snake robot pushing against obstacles are presented, and the simulation results are obtained from the dynamic simulation software WorkingModel. However, the contact between the snake robot and the obstacles are described with a compliant model, and it requires a stiff spring coefficient to represent the contact between a (hard) obstacle and a (hard) snake robot link. This results in stiff differential equations which are cumbersome to solve numerically.

Mathematical models that include the dynamics of motion yield a more accurate description of the behaviour of the system. This is advantageous with respect to simulation, but the models may get very large and unmanageable. Thus, the simplicity of an analytical analysis suffers. In 2D models, certain properties of the system help to simplify the model, but not all of these properties persist in 3D models. However, moving in, for example, a shattered building is an inherently 3D experience and thus 3D

dynamical models are necessary to simulate such motion. The complexity of the models grows a lot from 2D to 3D, and few 3D models have been presented in the published literature. For the presented models, the contact force between a snake robot and the ground surface is modelled using a linear spring-damper model. This is a very intuitive approach and easy to implement. However, the large spring coefficient needed to resemble a hard ground surface results in a stiff mathematical model which requires small integration steps to avoid instability in the numerical integration. One of the 3D models have been simplified to describe planar movement with a varying ground contact force to study sinus-lifting.

By going through the published literature, we see that there are several aspects of snake robot locomotion. The main concern is to find some kind of (often repetitive) reference for the joint angles such that the snake robot moves in some direction. A large variety of preprogrammed reference signals are proposed resulting in different motion patterns. The preprogrammed motion patterns can be improved and extended by incorporating feedback from external measurements of, for example, velocity, heading, and/or position of the snake robot in the generation of the joint reference angles. To some extent, such results are presented in the published literature. Others focus on finding optimal preprogrammed motion patterns with respect to energy consumption and distance travelled. Some snake robots are equipped with sensors and the motion pattern is then generated based on continuously updated measurements from the sensors. This has been achieved for a snake robot that moves forward by sensing and pushing against external objects in an experimental setup for planar motion. A large variety of 3D motion patterns are implemented for real snake robots and some of these motion patterns are based on rolling. To simulate such motion, some sort of exoskeleton is needed for contact with the ground surface. However, the 3D dynamical models presented only define the contact as a single point for each snake robot link, and thus this renders such motion impossible.

The above approaches are important contributions to making snake robots navigate by themselves in an unknown environment. However, there are still numerous challenges to face before a snake robot can be employed to be able to search efficiently for people trapped in a collapsed building or aid fire-fighters on a dangerous mission. This motivates further research on snake robots.

Chapter 3

Non-smooth Model of a 2D Snake Robot for Simulation

3.1 Introduction

The fastest biological snakes exploit roughness in the terrain for locomotion. They may push against rocks, branches, or other obstacles to move forward more efficiently (Gray, 1946; Hirose, 1993). Snakes can also exploit the walls of narrow passages for locomotion. Their mobility properties are interesting for snake robot locomotion and give the motivation for investigating snake robot obstacle aided locomotion. We define *obstacle aided locomotion* as snake robot locomotion where snake robots utilize walls or other external objects, apart from the flat ground, for propulsion. In order to develop such an obstacle aided locomotion scheme, we need a mathematical model that includes the interaction between the snake robot and a terrain with external objects for the snake robot to push against.

In this chapter we present a non-smooth (hybrid) modelling approach particularly suitable for modelling snake robot motion in general and obstacle aided locomotion in particular. We use this approach to develop a 2D mathematical model of a snake robot that can push against external objects for locomotion. The mathematical model is described in the framework of non-smooth dynamics and convex analysis (Brogliato, 1999; Glocker, 2001; Leine and Nijmeijer, 2004). The tools available in this framework that are necessary for the mathematical modelling of a snake robot will be introduced and explained when necessary.

This chapter is based on Transeth *et al.* (2007b) and Transeth *et al.* (2008b) and it is organized as follows. Section 3.2 gives a short introduc-

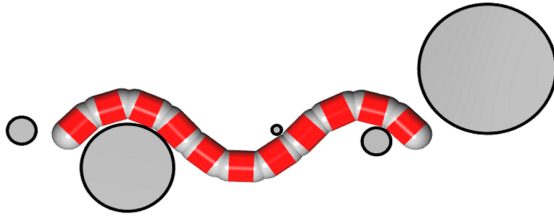


Figure 3.1: Planar snake robot with 11 links in an environment with obstacles that it may push against to move forward (the snake robot is moving towards the right in the illustration).

tion and motivation for the derivation of the mathematical model. The kinematics of the snake robot with obstacles is given in Section 3.3. Section 3.4 lays the foundation for finding the obstacle contact and ground friction forces. Section 3.5 describes the non-smooth dynamics of the snake robot, while Section 3.6 outlines the numerical treatment of the model. A summary of the most important results presented in this chapter is given in Section 3.7.

3.2 Summary of the Mathematical Model

This section contains a brief outline of how to derive a non-smooth mathematical model of a snake robot during planar motion. This preliminary section is meant to motivate and ease the understanding of the forthcoming deduction of the system equations.

The 2D model of the snake robot consists of n links connected by $n - 1$ one degree of freedom (DOF) rotational joints. An illustration of the planar snake robot in an environment with external circular objects (later referred to as obstacles) is given in Figure 3.1. Let $\mathbf{u} \in \mathbb{R}^{3n}$ be a vector containing the translational and rotational velocities of all the links of the snake robot (the structure of the snake robot together with the coordinates and reference frames are described further in Section 3.3). Let the differential measures $d\mathbf{u}$ and dt be loosely described for now as a ‘possible differential change’ in \mathbf{u} and the time t , respectively, while a more precise definition is given in Section 3.5.1. The use of differential measures allows for instantaneous changes in velocities which occur for impacts with obstacles. The system equations for the snake robot can now be written as

$$\mathbf{M}d\mathbf{u} - \mathbf{h}dt - d\mathbf{R} = \boldsymbol{\tau}dt, \quad (3.1)$$

where (3.1) is termed the *equality of measures* (Moreau, 1988). Here $\mathbf{M} \in \mathbb{R}^{3n \times 3n}$ is the mass matrix, $\boldsymbol{\tau} \in \mathbb{R}^{3n}$ contains all the joint actuator torques, $\mathbf{h} \in \mathbb{R}^{3n}$ represents all the smooth forces (for example gravity, note that $\mathbf{h} = \mathbf{0}_{3n \times 1}$ for the 2D model) and $d\mathbf{R} \in \mathbb{R}^{3n}$ accounts for the normal contact forces/impulses from the obstacles, the Coulomb friction forces/impulses and the bilateral constraint forces/impulses in the joints. Note: We allow in this chapter for *instantaneous* changes in velocities usually associated with collisions. Hence, the (normal contact/friction/constraint) ‘forces’ are not always defined due to the infinite accelerations. In these cases we have impulses instead of forces. The non-smooth equality of measures (3.1) allows us to formulate in a uniform manner both the smooth and non-smooth phases of motion. This is achieved partly by representing the contact forces/impulses as *contact impulse measures* (these are included by $d\mathbf{R}$ in the equality of measures).

A substantial part of the beginning of this chapter is devoted to deducting the force measure $d\mathbf{R}$. Hence, let us briefly look at how to derive the contribution of the normal contact impulse measure between an obstacle j and the first link, in $d\mathbf{R}$. Let $d\mathbf{R}_1 \in \mathbb{R}^3$ be the sum of contact impulse measures that directly affect link 1 (i.e. the first three elements of $d\mathbf{R}$). Then

$$d\mathbf{R}_1 = \mathbf{w}_H dP_H + \left\{ \begin{array}{l} \text{friction and joint constraint} \\ \text{impulse measures} \end{array} \right\}, \quad (3.2)$$

where $dP_H \in \mathbb{R}$ is the normal contact impulse measure from obstacle j on link 1 and $\mathbf{w}_H \in \mathbb{R}^3$ is the corresponding generalized force direction (subscripts ‘ j ’ and ‘1’ are omitted for brevity).

Let $g_H \in \mathbb{R}$ be a function giving the shortest distance between link 1 and the obstacle. Such a function is called a *gap function* (Glocker and Studer, 2005). The gap function is the starting point for the systematic approach of finding the impulse measures. The link and obstacle are separated if $g_H > 0$, are in contact if $g_H = 0$ and are penetrating each other if $g_H < 0$. Hence, the relative velocity between link 1 and the obstacle along the shortest line between the two objects can be defined as $\gamma_H := \dot{g}_H$, and it is written, by calculating \dot{g}_H , on the form

$$\gamma_H = \mathbf{w}_H^T \mathbf{u}_1, \quad (3.3)$$

where \mathbf{u}_1 is the velocity of link 1. Hence, by inspecting the expression for \dot{g}_H and formulating it as (3.3) we find an expression for \mathbf{w}_H in (3.2). The normal contact impulse measure dP_H is related to the relative velocity γ_H by employing a *set-valued force law* (see Section 3.5). The set-valuedness

of the force laws allows us to write each constitutive law (force law) with a single equation and avoid explicit switching between equations (for example when a collision occurs). In addition, this formulation provides an accurate description of the planar Coulomb friction (see Section 3.5.2). In the following three sections, we will elaborate on how to derive the elements that constitute (3.1), that is the various gap functions, relative velocities and finally the forces/impulses included in the equality of measures.

3.3 Kinematics

In this section we present the kinematics of the snake robot in an environment with obstacles. From the kinematics, we develop *gap functions* both for obstacle contact detection and for conforming to the bilateral joint constraints. The gap functions are later used as a basis for calculating the normal contact forces and the joint constraint forces.

First, we give an overview of the coordinates used to describe the position and orientation of the snake robot. The coordinates are chosen such that the mass matrix becomes constant. This is advantageous for the numerical treatment. Subsequently, the gap functions for describing the distance between the snake robot and the obstacles, and the ‘gaps’ in the joints, are found.

3.3.1 Snake Robot Description and Reference Frames

The model of a planar snake robot consists of n equal links connected by $n - 1$ one degree of freedom rotational joints. A snake robot with 3 links is depicted in Figure 3.2. Link i is shaped as a rectangle of length $2L_{GS_i}$ and width $2L_{SC_i}$. It is assumed that there is a massless semicircle of radius L_{SC_i} connected to the ends of each link. The semicircles together with the straight sides define the surface (or contour) of a link used for contact with the external objects. The distance between two adjacent joints is L_i .

Let the inertial reference frame be approximated by an earth-fixed frame $I = \{O, \mathbf{e}_x^I, \mathbf{e}_y^I\}$ with the origin O attached to the ground surface, and where the plane $(\mathbf{e}_x^I, \mathbf{e}_y^I)$ constitutes the horizontal ground surface which the snake robot can move on. A general notation used throughout this chapter is that a vector from the origin of frame I to a point A is given by $\mathbf{r}_A \in \mathbb{R}^2$ and a vector from point A to point B is written \mathbf{r}_{AB} . Let a vector \mathbf{r}_A described in frame I be written as ${}_I\mathbf{r}_A$. Denote the point G_i as the centre of gravity (CG) of link i where the CG is assumed to be in the

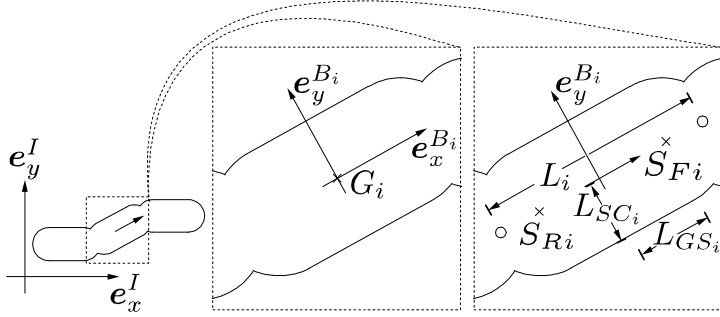


Figure 3.2: Snake robot with three links. The positions of the joints are given by the two small circles in the rightmost drawing.

middle of the rectangle. Denote the body-fixed frame $B_i = \{G_i, e_x^{B_i}, e_y^{B_i}\}$, where the origin is at the point G_i , the axis $e_x^{B_i}$ is pointing along the centre line of link i towards link $i + 1$ and $e_y^{B_i}$ is pointing transversal to the link following the right-hand convention. The centre point of the front and rear sphere which constitutes the ends of link i is denoted S_{Fi} and S_{Ri} , respectively.

The earth-fixed position and orientation of link i is

$$\mathbf{q}_i = \begin{bmatrix} {}_I \mathbf{r}_{G_i} \\ \theta_i \end{bmatrix}, \quad (3.4)$$

where ${}_I \mathbf{r}_{G_i} \in \mathbb{R}^2$ is the position of the CG of link i and $\theta_i \in S^1$ is the angle between the axes e_x^I and $e_x^{B_i}$. The translational and rotational velocities of link i are

$$\mathbf{u}_i = \begin{bmatrix} {}_I \mathbf{v}_{G_i} \\ \omega_i \end{bmatrix}, \quad (3.5)$$

where ${}_I \mathbf{v}_{G_i} := {}_I \dot{\mathbf{r}}_{G_i}$ and $\omega_i := \dot{\theta}_i$ when they exist (i.e. for impact free motion). All the positions and orientations, and velocities, are gathered in the two vectors

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \vdots \\ \mathbf{q}_n \end{bmatrix} \quad \text{and} \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_n \end{bmatrix}. \quad (3.6)$$

The transformation of a vector \mathbf{r} between reference frames is given by

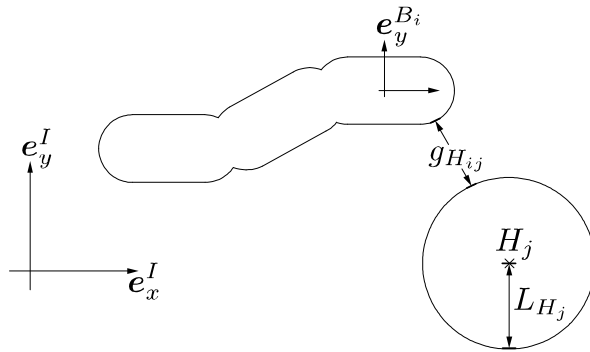


Figure 3.3: Snake robot with three links depicted with obstacle $j = 1$ closest to link $i = 3$.

${}^I \mathbf{r} = \mathbf{R}_{B_i}^I {}^{B_i} \mathbf{r}$ where the rotation matrix is

$$\mathbf{R}_{B_i}^I = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix}. \quad (3.7)$$

External fixed objects are included in the model for the snake robot to push against. Each obstacle j , $j = 1, 2, \dots, \nu$, is shaped as a circle in the plane with radius L_{H_j} and midpoint H_j , see Figure 3.3. The gap function $g_{H_{ij}}$ in the figure is described in the following.

3.3.2 Gap Functions for Obstacle Contact

We begin this section by giving an example of what a gap function is used for, then we introduce the gap functions for contact between the snake robot and the obstacles.

Example 3.1 (Gap Function) *Consider two convex rigid bodies 1 and 2. Now, we want to determine whether Body 1 is in contact with Body 2. This can be achieved by determining whether the minimal distance between the two bodies is equal to zero. The function that gives us the minimal distance is called a gap function (Glocker and Studer, 2005) and the point on Body 1 that is closest to Body 2 is called a (possible) contact point. Apart from determining if two rigid bodies are in contact, the gap function will also be used as a basis for calculating the direction and size of the forces involved in the contact.*

The gap function $g_{H_{ij}}$ gives the shortest distance between link i and obstacle j as illustrated in Figure 3.3 and is used to detect whether the two bodies are in contact. In addition, this gap function will be used later as a basis for calculating the forces involved in the contact.

Link i can, at any time-instance, only touch a convex obstacle j at a single point, resulting in a contact point that may move across the entire surface of a link. The surface (or contour) of a link consists of three parts: the rectangle, the front semicircle and the rear semicircle. Hence, we see that we need to find two separate kinds of gap functions for each link depending on which part of the link is closest to the obstacle. In particular, we need to find the shortest distance between a rectangle and a circle, and between two circles. This approach provides us with a *moving* contact point. Hence, the obstacle can come in contact with the snake robot at any point and the effect the contact force has on a snake robot link is therefore modelled accurately.

Distance Between Rectangle Part and Obstacle

When the rectangle-shaped part of link i is closest to obstacle j , we need to investigate the distance between a line (the $e_x^{B_i}$ -axis) and the centre of the obstacle H_j . The shortest distance between the $e_x^{B_i}$ -axis and H_j is

$$d_{ij} = ({}_I\mathbf{r}_{G_i} - {}_I\mathbf{r}_{H_j})^T {}_I\mathbf{e}_y^{B_i}. \quad (3.8)$$

Hence, the gap function when the rectangle part of link i is closest to obstacle j is

$$g_{H_{ij}} = |d_{ij}| - (L_{H_j} + L_{SC_i}). \quad (3.9)$$

Distance Between Circle Part and Obstacle

The minimal distance between the circle part of link i and obstacle j is found from the gap function

$$g_{H_{ij}} = \|\mathbf{r}_{H_j S_i}\| - (L_{H_j} + L_{SC_i}) \quad (3.10)$$

where

$$\mathbf{r}_{H_j S_i} = \mathbf{r}_{G_i} + \mathbf{r}_{G_i S_{Fi}} - \mathbf{r}_{H_j} \quad \text{or} \quad (3.11)$$

$$\mathbf{r}_{H_j S_i} = \mathbf{r}_{G_i} + \mathbf{r}_{G_i S_{Ri}} - \mathbf{r}_{H_j} \quad (3.12)$$

depending on whether the front or rear part of the link is closest to the obstacle, respectively. In addition, $\mathbf{r}_{G_i S_{Fi}} = L_{GS_i} \mathbf{e}_x^{B_i}$, $\mathbf{r}_{G_i S_{Ri}} = -L_{GS_i} \mathbf{e}_x^{B_i}$ and we define $\|\cdot\| := \|\cdot\|_2$.

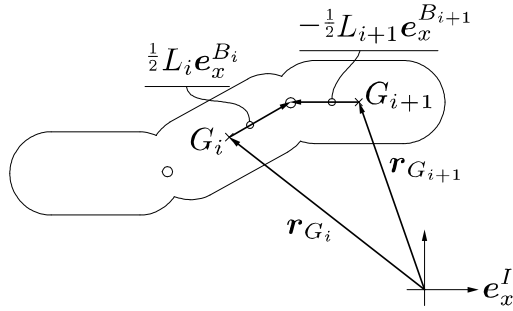


Figure 3.4: Illustration of how the joint gap function $g_{J_{i\chi}}$ is found. We see that for the snake (robot with 3 links) in the picture, the vectors are drawn for $i = 2$.

A Vector of All Obstacle Gap functions

We now gather the gap functions $g_{H_{ij}}$ for the combination of all n links and ν obstacles in the vector:

$$\mathbf{g}_H = [g_{H_{11}} \ \cdots \ g_{H_{n1}} \ g_{H_{12}} \ \cdots \ g_{H_{n2}} \ \cdots \ g_{H_{1\nu}} \ \cdots \ g_{H_{n\nu}}]^\top, \quad (3.13)$$

where $g_{H_{ij}}$ is found from either (3.9) or (3.10) depending on which part of link i is closest to obstacle j .

3.3.3 Bilateral Constraints: Joints

Each joint introduces two bilateral constraints in the model. These constraints keep the ‘gap’ in a joint equal to zero. An expression for this ‘gap’, a gap function, needs to be found in order to calculate the constraint forces. To find the gap function, we need to relate the position of joint i between link i and $i + 1$ to both adjacent links. The position of joint i going via link i and $i + 1$ is given by $I \mathbf{r}_{J_{Fi}} = I \mathbf{r}_{G_i} + \frac{1}{2} L_i I \mathbf{e}_x^{B_i}$ and $I \mathbf{r}_{J_{Ri+1}} = I \mathbf{r}_{G_{i+1}} - \frac{1}{2} L_{i+1} I \mathbf{e}_x^{B_{i+1}}$, respectively. Figure 3.4 depicts how these position vectors are found. Now, the gap functions for the gaps in the joints are

$$g_{J_{i\chi}} = (I \mathbf{e}_\chi^I)^\top (I \mathbf{r}_{J_{Fi}} - I \mathbf{r}_{J_{Ri+1}}), \quad (3.14)$$

for $\chi = x, y$ and $i = 1, \dots, n - 1$.

3.4 Contact Constraints on Velocity Level

In this section the relative velocities between the snake robot, and the obstacles and the ground, are found by taking the time-derivative (when they exist) of the corresponding gap functions. The relative velocities are employed to calculate the normal contact forces involved in the contact between the snake robot and the obstacles (Le Saux *et al.*, 2005; Leine and Nijmeijer, 2004). Also, the relative velocities are employed to find the friction forces between the ground and the snake robot, and the bilateral constraint forces in the joints. This is shown in Section 3.5.2.

3.4.1 Relative Velocity Between an Obstacle and a Link

The relative velocity between a link i and an obstacle j is defined as $\gamma_{H_{ij}} := \dot{g}_{H_{ij}}$ (when the time-derivative exists). As for the gap functions, the expression for the relative velocity depends on which part of the link is closest to the obstacle.

Rectangle Part Closest to Obstacle

If the rectangle-shaped part of link i is closest to obstacle j , then the relative velocity between these two convex objects $\gamma_{H_{ij}} := \dot{g}_{H_{ij}}$ is found, by employing (3.9), as

$$\gamma_{H_{ij}} = \frac{d}{dt} |d_{ij}| \implies \gamma_{H_{ij}} = \mathbf{w}_{H_{ij}}^T \mathbf{u}_i, \quad (3.15)$$

where d_{ij} is given in (3.8) and the generalized force direction is

$$\mathbf{w}_{H_{ij}}^T = \text{sign}(d_{ij}) \left[\left({}_I \mathbf{e}_y^{B_i} \right)^T \quad \left({}_I \mathbf{r}_{H_j} - {}_I \mathbf{r}_{G_i} \right)^T \begin{bmatrix} \cos(\theta_i) \\ \sin(\theta_i) \end{bmatrix} \right] \mathbf{u}_i, \quad (3.16)$$

for $i = 1, \dots, n$ and $j = 1, \dots, \nu$.

Circle Part Closest to Obstacle

The gap function (3.10) is employed to define the relative velocity $\gamma_{H_{ij}} := \dot{g}_{H_{ij}}$ when one of the circle-shaped parts of link i is closest to obstacle j . By definition, the relative velocity is

$$\gamma_{H_{ij}} = \frac{d}{dt} \left\| {}_I \mathbf{r}_{H_j S_i} \right\| \implies \gamma_{H_{ij}} = \mathbf{w}_{H_{ij}}^T \mathbf{u}_i, \quad (3.17)$$

where the generalized force direction is, for $i = 1, \dots, n$ and $j = 1, \dots, \nu$,

$$\mathbf{w}_{H_{ij}}^T = \frac{I \mathbf{r}_{H_j S_i}}{\|I \mathbf{r}_{H_j S_i}\|} \left[\mathbf{I}_{2 \times 2} \quad \mathbf{R}_{B_i}^I \mathbf{D}_{B_i} \mathbf{r}_{G_i S_i} \right], \quad (3.18)$$

$$\mathbf{D} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{I}_{2 \times 2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (3.19)$$

and $I \mathbf{r}_{H_j S_i}$ is given by (3.11) or (3.12) and $\mathbf{r}_{G_i S_i}$ is equal to either ${}_{B_i} \mathbf{r}_{G_i S_{F_i}}$ or ${}_{B_i} \mathbf{r}_{G_i S_{R_i}}$ depending on whether the front (F_i) or rear (R_i) part of the link is closest to the obstacle as described in Section 3.3.2. To find the expression for $\mathbf{w}_{H_{ij}}^T$ in (3.18) we have employed the following identities:

$$\frac{d}{dt} \|I \mathbf{r}_{H_j S_i}\| = \frac{I \mathbf{r}_{H_j S_i}}{\|I \mathbf{r}_{H_j S_i}\|} \frac{d}{dt} I \mathbf{r}_{H_j S_i}, \quad (3.20)$$

$$\frac{d}{dt} \mathbf{R}_{B_i}^I = \mathbf{R}_{B_i}^I \mathbf{D}. \quad (3.21)$$

Notice that we have employed the same notation for $\mathbf{w}_{H_{ij}}$ in (3.16) and (3.18) to reduce the number of subscripts.

Vector of All Relative Velocities for Obstacles

We gather the relative velocities $\gamma_{H_{ij}}$ for all n links and ν obstacles in the vector

$$\boldsymbol{\gamma}_H = \mathbf{W}_H^T \mathbf{u} \in \mathbb{R}^{n\nu}, \quad (3.22)$$

where

$$\boldsymbol{\gamma}_H = [\gamma_{H_{11}} \quad \cdots \quad \gamma_{H_{n1}} \quad \gamma_{H_{12}} \quad \cdots \quad \gamma_{H_{n2}} \quad \cdots \quad \gamma_{H_{1\nu}} \quad \cdots \quad \gamma_{H_{n\nu}}]^T, \quad (3.23)$$

$$\mathbf{W}_H = [\mathbf{W}_{H_1} \quad \cdots \quad \mathbf{W}_{H_\nu}] \in \mathbb{R}^{3n \times n\nu}, \quad (3.24)$$

and

$$\mathbf{W}_{H_j} = \begin{bmatrix} \mathbf{w}_{H_{1j}} & \mathbf{0}_{3 \times 1} & \cdots & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} & \mathbf{w}_{H_{2j}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} & \cdots & \mathbf{0}_{3 \times 1} & \mathbf{w}_{H_{nj}} \end{bmatrix} \in \mathbb{R}^{3n \times n}. \quad (3.25)$$

Depending on which part of the link is closest to the obstacle, $\mathbf{w}_{H_{ij}}$ is obtained either from (3.16) or (3.18).

3.4.2 Tangential Relative Velocity

The tangential relative velocities between a link i and the ground is presented here. For the case of the planar snake robot, these velocities are the velocities of the centre of gravity G_i for each link with respect to the earth-fixed coordinate axis $(\mathbf{e}_x^I, \mathbf{e}_y^I)$. The relative velocities will later be employed to find the friction forces between the link and the ground. We start by finding the tangential relative velocities parallel to the \mathbf{e}_x^I and \mathbf{e}_y^I -axis. These are denoted for link i by $\gamma'_{T_{ix}}$ and $\gamma'_{T_{iy}}$, respectively. However, in order to easily find the friction forces longitudinal and lateral to each link, we need to derive the tangential relative velocities longitudinal $\gamma_{T_{ix}}$ and lateral $\gamma_{T_{iy}}$ to each link.

The tangential relative velocities parallel to the \mathbf{e}_x^I and \mathbf{e}_y^I -axis are

$$\gamma'_{T_{i\chi}} = (\mathbf{I} \mathbf{e}_\chi^I)^\top \mathbf{I} \mathbf{v}_{G_i} \implies \gamma'_{T_{i\chi}} = (\mathbf{w}'_{T_{i\chi}})^\top \mathbf{u}_i, \quad (3.26)$$

where the generalized force direction is

$$(\mathbf{w}'_{T_{i\chi}})^\top = [(\mathbf{I} \mathbf{e}_\chi^I)^\top \quad 0], \quad (3.27)$$

for $\chi = x, y$ and $i = 1, \dots, n$. The two relative velocities are combined in one vector so that

$$\gamma'_{T_i} = (\mathbf{W}'_{T_i})^\top \mathbf{u}_i, \quad (3.28)$$

where

$$\gamma'_{T_i} = \begin{bmatrix} \gamma'_{T_{ix}} \\ \gamma'_{T_{iy}} \end{bmatrix} \in \mathbb{R}^2, \quad \mathbf{W}'_{T_i} = \begin{bmatrix} \mathbf{w}'_{T_{ix}} & \mathbf{w}'_{T_{iy}} \end{bmatrix} \in \mathbb{R}^{3 \times 2}. \quad (3.29)$$

The tangential relative velocities along the $\mathbf{e}_x^{B_i}$ and $\mathbf{e}_y^{B_i}$ -axis can now be found as follows. Define

$$\gamma_{T_i} = \begin{bmatrix} \gamma_{T_{ix}} \\ \gamma_{T_{iy}} \end{bmatrix}. \quad (3.30)$$

Then, by employing the rotation matrix $\mathbf{R}_{B_i}^I$ in (3.7), we have that $\gamma'_{T_i} = \mathbf{R}_{B_i}^I \gamma_{T_i}$. Hence,

$$\gamma_{T_i} = \mathbf{W}_{T_i}^\top \mathbf{u}_i, \quad (3.31)$$

where

$$\mathbf{W}_{T_i} = \mathbf{W}'_{T_i} \mathbf{R}_{B_i}^I, \quad (3.32)$$

for $i = 1, \dots, n$.

We gather all the tangential relative velocities γ_{T_i} as

$$\gamma_T = \mathbf{W}_T^T \mathbf{u}, \quad (3.33)$$

where

$$\gamma_T = \begin{bmatrix} \gamma_{T_1} \\ \vdots \\ \gamma_{T_n} \end{bmatrix} \in \mathbb{R}^{2n}, \quad \mathbf{W}_T = \begin{bmatrix} \mathbf{W}_{T_1} & \mathbf{0}_{3 \times 2} & \cdots & \mathbf{0}_{3 \times 2} \\ \mathbf{0}_{3 \times 2} & \ddots & & \vdots \\ \vdots & & \ddots & \mathbf{0}_{3 \times 2} \\ \mathbf{0}_{3 \times 2} & \cdots & \mathbf{0}_{3 \times 2} & \mathbf{W}_{T_n} \end{bmatrix} \in \mathbb{R}^{3n \times 2n}, \quad (3.34)$$

and \mathbf{W}_{T_i} is found from (3.32).

3.4.3 Bilateral Constraints: Joints

We need the relative velocities of the ‘gaps’ in the joints to find the forces involved in the bilateral constraint forces acting on the links and imposed by the joints. The bilateral constraint forces ensure that adjacent links only moves relative to each other in ways allowed by the 1 DOF joint that connects the links. The relative joint velocities for joint i along the \mathbf{e}_x^I - and \mathbf{e}_y^I -axis are defined as $\gamma_{J_{ix}} := \dot{J}_{ix}$ and $\gamma_{J_{iy}} := \dot{J}_{iy}$ (when the time-derivatives exist). Hence, by employing (3.14) we find that

$$\gamma_{J_{i\chi}} = \mathbf{w}_{J_{i\chi}}^T \begin{bmatrix} \mathbf{u}_i \\ \mathbf{u}_{i+1} \end{bmatrix}, \quad (3.35)$$

for $\chi = x, y$ and $i = 1, \dots, n-1$, where

$$\mathbf{w}_{J_{ix}}^T = \left[(I \mathbf{e}_x^I)^T \quad \frac{-L_i}{2} \sin(\theta_i) \quad - (I \mathbf{e}_x^I)^T \quad \frac{-L_{i+1}}{2} \sin(\theta_{i+1}) \right], \quad (3.36)$$

$$\mathbf{w}_{J_{iy}}^T = \left[(I \mathbf{e}_y^I)^T \quad \frac{L_i}{2} \cos(\theta_i) \quad - (I \mathbf{e}_y^I)^T \quad \frac{L_{i+1}}{2} \cos(\theta_{i+1}) \right]. \quad (3.37)$$

By defining $\gamma_{J_i} = [\gamma_{J_{ix}} \quad \gamma_{J_{iy}}]^T$, we gather all the relative joint velocities as

$$\gamma_J = \mathbf{W}_J^T \mathbf{u}, \quad (3.38)$$

where $\gamma_J = [\gamma_{J_1}^T \quad \cdots \quad \gamma_{J_{n-1}}^T]^T \in \mathbb{R}^{2(n-1)}$,

$$\mathbf{W}_J^T = \begin{bmatrix} \mathbf{W}_{J_1}^T & \mathbf{0}_{2 \times 3} & \cdots & \mathbf{0}_{2 \times 3} \\ \mathbf{0}_{2 \times 3} & \mathbf{W}_{J_2}^T & & \mathbf{0}_{2 \times 3} \\ \vdots & & \ddots & \mathbf{0}_{2 \times 3} \\ \mathbf{0}_{2 \times 3} & \cdots & \mathbf{0}_{2 \times 3} & \mathbf{W}_{J_{n-1}}^T \end{bmatrix} \in \mathbb{R}^{3n \times 2(n-1)}, \quad (3.39)$$

and $\mathbf{W}_{J_i} = [\mathbf{w}_{J_{ix}} \quad \mathbf{w}_{J_{iy}}]$ for $i = 1, \dots, n-1$.

3.5 Non-smooth Dynamics

The starting point for describing the dynamics of the snake robot is the *equality of measures* as introduced in Moreau (1988). The equality of measures includes both the equations of motion for impact free motion and the impact equations. The impact equations give rise to impulsive behaviour (Glocker and Studer, 2005). In this section we employ all the results from Section 3.3 and Section 3.4 to find the various components of the equality of measures.

3.5.1 The Equality of Measures

The equality of measures describes the dynamics of the snake robot within the context of non-smooth dynamics. The non-smoothness allows for instantaneous velocity jumps, usually associated with impacts. The velocity $\mathbf{u}(t)$ is assumed to admit a right \mathbf{u}^+ and left \mathbf{u}^- limit for all t in the (short) time-interval $I_t = [t_A, t_E]$ (Moreau, 1988) and its time-derivative $\dot{\mathbf{u}}$ exists for almost all $t \in I_t$. The differential measure $d\mathbf{u}$ is assumed to be decomposed into two parts to be able to obtain \mathbf{u} from integration:

$$d\mathbf{u} = \dot{\mathbf{u}}dt + (\mathbf{u}^+ - \mathbf{u}^-) d\eta, \quad (3.40)$$

where dt denotes the Lebesgue measure and $d\eta$ denotes the atomic measure where $\int_{t_i} d\eta = 1$ if $\mathbf{u}^+(t_i) - \mathbf{u}^-(t_i) \neq 0$. The total increment of \mathbf{u} over a compact sub-interval $[t_1, t_2]$ of the time-interval I_t is

$$\int_{[t_1, t_2]} d\mathbf{u} = \mathbf{u}^+(t_2) - \mathbf{u}^-(t_1) \quad (3.41)$$

and is due to a continuous change (i.e. impact free motion) stemming from $\dot{\mathbf{u}}$ as well as possible discontinuities in \mathbf{u} within the time-interval $[t_1, t_2]$. Equation (3.41) is even valid when the time-interval $[t_1, t_2]$ reduces to a singleton $\{t_1\}$ (i.e. $t_2 = t_1$):

$$\int_{\{t_1\}} d\mathbf{u} = \mathbf{u}^+(t_1) - \mathbf{u}^-(t_1), \quad (3.42)$$

due to the atomic measure $d\eta$. Hence, if a velocity jump occurs for $t = t_1$ then (3.41) gives a non-zero result. Note that if the obstacles are removed from the model (thus, no impacts will occur), then we have $\mathbf{u}^+(t) = \mathbf{u}^-(t) \forall t$, and (3.40) becomes $d\mathbf{u} = \dot{\mathbf{u}}dt$.

The Newton-Euler equations as an equality of measures can be written in a general form as

$$\mathbf{M}(\mathbf{q}, t) d\mathbf{u} - \mathbf{h}(\mathbf{q}, \mathbf{u}, t) dt - d\mathbf{R} = \boldsymbol{\tau} dt, \quad (3.43)$$

where $\mathbf{M}(\mathbf{q}, t)$ is the mass matrix, $\mathbf{h}(\mathbf{q}, \mathbf{u}, t)$ is the vector of smooth forces, $d\mathbf{R}$ is the force measure of possibly atomic impact impulsions and $\boldsymbol{\tau}$ is the vector of applied torques. For the planar snake robot, the equality of measures reduces to

$$\mathbf{M}d\mathbf{u} - d\mathbf{R} = \boldsymbol{\tau} dt, \quad (3.44)$$

where $d\mathbf{R}$ and $\boldsymbol{\tau}$ will be described in the following and we note that $\mathbf{h}(\mathbf{q}, \mathbf{u}, t) = \mathbf{0}_{3n}$. The mass matrix is

$$\mathbf{M}(\mathbf{q}, t) = \mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & \mathbf{0}_{3 \times 3} & \cdots & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \ddots & & \vdots \\ \vdots & & \ddots & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \cdots & \mathbf{0}_{3 \times 3} & \mathbf{M}_n \end{bmatrix} \in \mathbb{R}^{3n \times 3n}, \quad (3.45)$$

where

$$\mathbf{M}_i = \begin{bmatrix} m_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & \Theta_{\psi_i} \end{bmatrix} \in \mathbb{R}^{3 \times 3}, \quad i = 1, \dots, n, \quad (3.46)$$

and $m_i \in \mathbb{R}$ and $\Theta_{\psi_i} \in \mathbb{R}$ are the mass and moment of inertia of link i , respectively. Hence, the mass matrix \mathbf{M} is diagonal and constant.

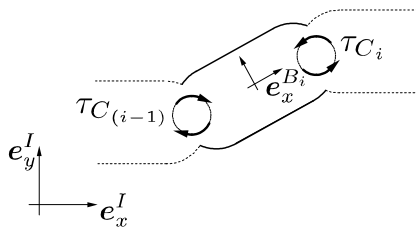
The force measure $d\mathbf{R}$ accounts for all contact forces and impulses. Let \mathcal{H} be the set of all active contacts with the obstacles

$$\mathcal{H}(t) = \{b | g_{H_b}(\mathbf{q}(t)) = 0\} \subseteq \{1, 2, \dots, n\nu\}, \quad (3.47)$$

where g_{H_b} is the b -th element of the vector \mathbf{g}_H in (3.13). The force measure can now be written as

$$d\mathbf{R} = \mathbf{W}_T d\mathbf{P}_T + \mathbf{W}_J d\mathbf{P}_J + \sum_{b \in \mathcal{H}} (\mathbf{W}_H)_b dP_{H_b}, \quad (3.48)$$

where dP_{H_b} is the normal contact impulse measure between a link and an obstacle, $d\mathbf{P}_T$ is the tangential contact impulse measure (the friction) between the ground and the links and $d\mathbf{P}_J$ is the contact impulse measures due to the bilateral constraints in the joints. The order of the elements in the vectors $d\mathbf{P}_T$ and $d\mathbf{P}_J$ corresponds to the vectors of relative velocities (3.33) and (3.38), respectively. The notation $(\mathbf{W}_H)_b$ denotes the b -th

Figure 3.5: Control torques on link i .

column of the \mathbf{W}_H -matrix in (3.24). The contact impulse measures are, similarly to the velocity measure, decomposed into two parts. One part consists of the Lebesgue-measurable force while the other part is a purely atomic impact impulse. This is written for dP_{H_b} as

$$dP_{H_b} = \lambda_{H_b} dt + \Lambda_{H_b} d\eta, \quad (3.49)$$

where λ_{H_b} is the Lebesgue-measurable force due to a *continuous contact* between a link and an obstacle, and Λ_{H_b} is a purely atomic impact impulse caused by a *collision* between the two objects. The same decomposition can be performed for the other contact impulse measures. The Lebesgue measurable force and the purely atomic impact impulse for the normal contact with the obstacles and the friction are found from the force laws given in Section 3.5.2.

The $n - 1$ joint actuators are modelled as applied torques. The total torque applied to link i (and thus every third element of the vector $\boldsymbol{\tau}$) is

$$\tau_{3 \cdot i} = \tau_{C_i} - \tau_{C_{i-1}}, \quad (3.50)$$

for $i = 1, \dots, n$ where $\tau_{C_i} \in \mathbb{R}$ is the torque applied to joint i and $\tau_{C_0} = \tau_{C_n} = 0$ (there are no actuators at the two ends of the snake robot). The torques applied to link i are depicted in Figure 3.5. The desired torques are found with a PD-controller:

$$\tau_{C_i} = -k_P (\phi_{h_i,d} - \phi_{h_i}) - k_D (\dot{\phi}_{h_i,d} - \omega_{J_i}), \quad (3.51)$$

where $k_P \in \mathbb{R}^+$, $k_D \in \mathbb{R}^+$ are positive constants, the relative joint angle ϕ_{h_i} and angular velocity are

$$\phi_{h_i} = \theta_{i+1} - \theta_i, \quad (3.52)$$

$$\omega_{J_i} = \omega_{i+1} - \omega_i, \quad (3.53)$$

respectively, and $\phi_{h_i,d}$ is the desired joint angle for joint i for $i = 1, \dots, n-1$. The joint torques are included in the equality of measures (3.44) through the vector

$$\boldsymbol{\tau} = [\mathbf{0}_{1 \times 2} \quad \tau_3 \quad \mathbf{0}_{1 \times 2} \quad \tau_6 \quad \cdots \quad \mathbf{0}_{1 \times 2} \quad \tau_{3n}]^T \in \mathbb{R}^{3n}. \quad (3.54)$$

3.5.2 Constitutive Laws for Contact Forces

In this section we will introduce set-valued force laws for normal contact with the obstacles and for Coulomb friction. The set-valuedness of the force laws allows us to write each constitutive law with a single equation thus avoiding explicit switching between equations in the numerical treatment. These laws will all be formulated on velocity level using the relative contact velocities $\boldsymbol{\gamma}$ given by (3.22) and (3.33). Subsequently, the set-valued force laws are formulated as equalities in the end of this section using the so-called ‘proximal point function’ in order to include the force laws in the numerical simulation (Leine and Nijmeijer, 2004).

Normal Contact Force

The normal contact between a link and an obstacle is described by the unilateral constraint

$$g_H \geq 0, \quad \lambda_H \geq 0, \quad g_H \lambda_H = 0, \quad (3.55)$$

which is known as Signorini’s law (Leine and Nijmeijer, 2004). Here, λ_H is the normal contact force and g_H is the gap function. The subscripts i and j are temporarily removed for brevity. This set-valued force law states that the contact is impenetrable, $g_H \geq 0$, the contact can only transmit pressure forces $\lambda_H \geq 0$ and the contact force λ_H does not produce work $g_H \lambda_H = 0$. The force law can be expressed on different kinematic levels: displacement level (3.55), velocity level and acceleration level. In the following we express all force laws for a closed contact (i.e. when two objects are in contact) on velocity level, while all forces vanish for open contacts (i.e. when two objects are not in contact). Then, by employing concepts of convex analysis, the relationship between the relative velocity and the Lebesgue measurable normal contact force (not an impulse) may be written for a closed contact $g_H = 0$ as an inclusion on velocity level

$$-\boldsymbol{\gamma}_H \in N_{C_H}(\lambda_H), \quad (3.56)$$

where $\boldsymbol{\gamma}_H \in \mathbb{R}$ the relative velocity in (3.15) or (3.17), the convex set $C_H = \{\lambda_H \mid \lambda_H \geq 0\} = \mathbb{R}^+$ is the set of admissible contact forces and N_{C_H}

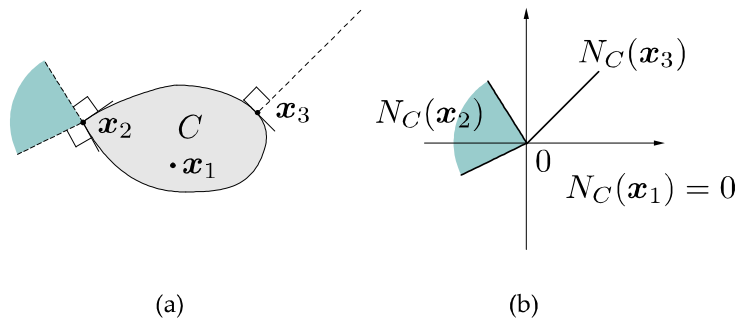


Figure 3.6: (a) Three points in and on the boundary of the set C , and (b) the normal cones to the set C at the points (Illustration inspired by Leine and Nijmeijer (2004)).

is the normal cone to the set C_H (Leine and Nijmeijer, 2004). The concept of normal cones will be illustrated shortly by two examples. For the time being, we note that the inclusion (3.56) is equivalent to the condition

$$\gamma_H \geq 0, \lambda_H \geq 0, \quad \gamma_H \lambda_H = 0, \quad (3.57)$$

for a closed contact $g_H = 0$. The force law (3.56) describes the impenetrability of sustained contact, i.e. $g_H = 0$ and $\gamma_H = 0$, as well as detachment: $\gamma_H > 0 \Rightarrow \lambda_H = 0$. However, (3.56) does not cover impacts (where we have impulses instead of forces). For impacts we need a similar *impact law* described at the end of this subsection. We formalize the above comment on impenetrability and detachment by giving the following two examples:

Example 3.2 (Normal cone) *From the definition of a normal cone $N_C(\mathbf{x})$ to a convex set C at the point $\mathbf{x} \in \mathbb{R}^n$ (Leine and Nijmeijer, 2004; Rockafellar, 1970), we have that $N_C(\mathbf{x}) = \{0\}$ for $\mathbf{x} \in \text{int } C$, and $N_C(\mathbf{x}) = \{\emptyset\}$ for $\mathbf{x} \notin C$. If \mathbf{x} is on the boundary of C , then $N_C(\mathbf{x})$ is, somewhat loosely described, the set of all vectors $\mathbf{y} \in \mathbb{R}^n$ that are normal to $\mathbf{x} \in C$. See Figure 3.6 for an illustration of various normal cones.*

Example 3.3 (Normal cone) *The boundary of the set $C_H = \mathbb{R}^+$ in (3.56) is the single point $\lambda_H = 0$, and the interior of C_H is $\mathbb{R}^+ \setminus \{0\}$. Hence, we have $N_{C_H}(0) = \mathbb{R}^- \setminus \{0\}$ and $N_{C_H}(2) = \{\emptyset\}$.*

The force law (3.56) only covers finite-valued contact efforts during impulse free motion, i.e. all velocities are absolutely continuous in time. When

a collision occurs in a rigid-body setting, then the velocities will be locally discontinuous in order to prevent penetration. The velocity jump is accompanied by an impact impulse Λ_H , for which we will set up an impact law. The relative velocity admits, similarly to the velocities \mathbf{u} , a right γ_H^+ and a left γ_H^- limit. The impact law for a completely inelastic impact at a closed contact can now be written as

$$-\gamma_H^+ \in N_{C_H}(\Lambda_H), \quad C_H = \{\Lambda_H \mid \Lambda_H \geq 0\} = \mathbb{R}^+, \quad (3.58)$$

which is equivalent to the condition

$$\gamma_H^+ \geq 0, \quad \Lambda_H \geq 0, \quad \gamma_H^+ \Lambda_H = 0. \quad (3.59)$$

Notice that the force law (3.56) and the impact law (3.58) is on the same form. We have earlier stated that there is no need for an explicit switch between equations when e.g. an impact occurs in the numerical treatment of the model. This becomes evident in Section 3.6 with the introduction of *contact impulses* (which include both forces and possible impulses) for the discretization of the system dynamics.

Coulomb Friction Force

Similarly to the force law (3.56) for normal contact, we describe the constitutive description for friction using an inclusion on a normal cone. The friction force $\boldsymbol{\lambda}_T \in \mathbb{R}^2$ between the ground and a link (we omit subscript ‘ i ’ for brevity), in the two-dimensional tangent plane to the contact point, is modelled with a Coulomb friction law

$$-\boldsymbol{\gamma}_T \in N_{C_T}(\boldsymbol{\lambda}_T), \quad (3.60)$$

where $\boldsymbol{\gamma}_T \in \mathbb{R}^2$ is the relative sliding velocity for link i given in (3.31), C_T is the (convex) set of admissible friction forces and N_{C_T} is the normal cone to the set C_T (see Figure 3.7 for an illustration of the force law). In the following, we show how to describe isotropic and anisotropic spatial Coulomb friction with (3.60) by changing the set C_T . First, isotropic friction will be elaborated on. Then we extend the force law (3.60) to describe a particular form of anisotropic friction, called orthotropic friction. However, before we begin, consider the following example that describes the force law (3.60):

Example 3.4 *Please refer to Figure 3.7 (a) and Example 3.2, and observe that the force law (3.60) distinguishes between two cases: if the friction force is in the interior of the set C_{T_1} , or if it is on the boundary of the set.*

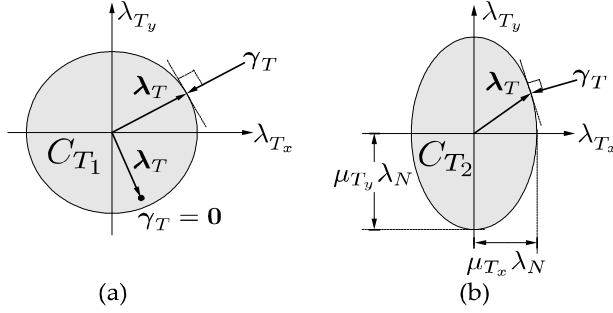


Figure 3.7: Relationship between tangential relative velocity and friction force for (a) isotropic friction with the set C_{T_1} (in grey) given by (3.61) and (b) anisotropic friction where C_{T_2} is found from (3.63).

If $\lambda_T \in \text{int } C_{T_1}$, then it holds that $N_{C_{T_1}}(\lambda_T) = \mathbf{0}$ from which we conclude that $\gamma_T = \mathbf{0}$. Obviously, this corresponds to the stick phase of the friction law. If the friction force is on the boundary of C_{T_1} , then the normal cone $N_{C_{T_1}}(\lambda_T)$ consists of the outward normal ray on the circle C_{T_1} at the point λ_T . This corresponds to that the link is sliding.

Isotropic friction indicates that the friction forces are not dependent on the orientation of the snake robot links while sliding. We employ the force law (3.60) to describe isotropic friction by choosing the convex set C_T as

$$C_{T_1} = \{\lambda_T \mid \|\lambda_T\| \leq \mu_T \lambda_N\}, \quad (3.61)$$

where $\mu_T > 0$ is the friction coefficient and $\lambda_N = mg$. The resulting relationship between the tangential relative velocities and the friction forces is illustrated in Figure 3.7 (a) where a possible resulting friction force for $\gamma_T = \mathbf{0}$ is included. The set-valued force law (3.60) with (3.61) contains the cases of stick and slip:

$$\begin{aligned} \text{stick : } & \gamma_T = \mathbf{0}, \quad \|\lambda_T\| \leq \mu_T \lambda_N, \\ \text{slip : } & \gamma_T \neq \mathbf{0}, \quad \lambda_T = -\mu_T \lambda_N \frac{\gamma_T}{\|\gamma_T\|}. \end{aligned} \quad (3.62)$$

The advantage of formulating the friction law as the inclusion (3.60) now becomes apparent. A spatial friction law such as (3.60), which is equivalent to (3.62) for $C_T = C_{T_1}$, can not properly be described by a set-valued sign-function. Some authors model the spatial contact with two sign-functions for the two components of the relative sliding velocity using

two friction coefficients μ_{T_x} and μ_{T_y} (Ma *et al.*, 2003a; Saito *et al.*, 2002). This results however directly in an anisotropic friction law, as the friction force and the sliding velocity do no longer point in opposite directions (even for $\mu_{T_x} = \mu_{T_y}$). Then such a double-sign-function force law corresponds to (3.60) with C_T being a rectangle with length $2\mu_{T_x}\lambda_N$ and width $2\mu_{T_y}\lambda_N$. Also, the (set-valued) sign function can be approximated with a smooth function, for example some arctangent function. This results in a very steep slope of the friction curve near zero relative velocity. Such an approach is very cumbersome for two reasons. First of all, stiction can not properly be described: an object on a slope will with a smoothed friction law always slide. Secondly, the very steep slope of the friction curve causes the differential equations of motion to become numerically stiff. Summarising, we see that (3.60) or (3.62) describes spatial Coulomb friction taking isotropy and stiction properly into account. We prefer using (3.60) instead of (3.62), because the latter becomes not well conditioned for very small γ_T when used in numerics. Note also that the force laws for Coulomb friction (3.60) and normal contact with the obstacles (3.56) have the same mathematical form. Moreover, the inclusion (3.60) is much more general since we can easily change the convex set C_T to get a different (and hence anisotropic) friction model.

Orthotropic friction (which is a particular form of anisotropic friction) can be described by the force law (3.60) with the set C_T being the equal to the set

$$C_{T_2} = \left\{ \boldsymbol{\lambda}_T \mid \frac{\lambda_{T_x}^2}{(\mu_{T_x}\lambda_N)^2} + \frac{\lambda_{T_y}^2}{(\mu_{T_y}\lambda_N)^2} \leq 1 \right\}, \quad (3.63)$$

where $\boldsymbol{\lambda}_T = [\lambda_{T_x} \ \lambda_{T_y}]^T$, and $\mu_{T_x}, \mu_{T_y} > 0$ are directional friction coefficients along the $\mathbf{e}_x^{B_i}$ - and $\mathbf{e}_y^{B_i}$ -axis, respectively. The relationship between the direction of the tangential relative velocity and the direction of the corresponding friction force is illustrated in Figure 3.7 (b) for $\mu_{T_x} < \mu_{T_y}$. Note that for $\mu_{T_x} = \mu_{T_y} = \mu_T$, the set C_{T_2} is equal to C_{T_1} .

While the force law (3.60) only describes the Coulomb friction during impulse free motion, we also need a force law for impact impulses $\mathbf{\Lambda}_T$. These are found from the exact same form as (3.60) by replacing $\boldsymbol{\gamma}_T$ with its right limit $\boldsymbol{\gamma}_T^+$ and inserting $\mathbf{\Lambda}_T$ instead of $\boldsymbol{\lambda}_T$ both in (3.60) and in the convex set C_T .

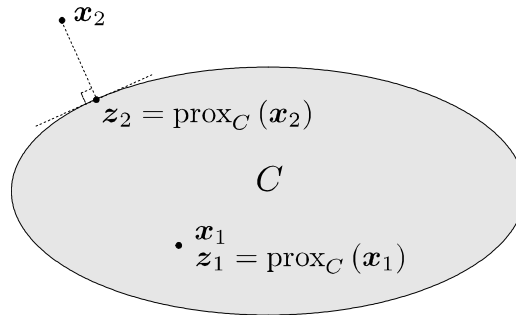


Figure 3.8: Illustration of the prox-function for the set C marked in grey. Notice that $x_1 = z_1$ since $x_1 \in C$.

Constitutive Laws as Projections

An inclusion can not be directly employed in numerical calculations. Hence, we transform the force laws (3.56) and (3.60), which have been stated as an inclusion to a normal cone, into an equality. This is achieved through the so-called proximal point function $\text{prox}_C(\mathbf{x})$, which equals \mathbf{x} if $\mathbf{x} \in C$ and equals the closest point to \mathbf{x} on the boundary of C if $\mathbf{x} \notin C$. This is illustrated in Figure 3.8. The set C must be convex. Using the proximal point function, we transform the force laws into implicit equalities (Leine and Nijmeijer, 2004):

$$-\gamma_\kappa \in N_{C_\kappa}(\lambda_\kappa) \iff \lambda_\kappa = \text{prox}_{C_\kappa}(\lambda_\kappa - r_\kappa \gamma_\kappa), \quad (3.64)$$

where $r_\kappa > 0$ for $\kappa = H, T$.

3.6 Numerical Algorithm: Time-Stepping

The numerical solution of the equality of measures (3.44) is found with an algorithm introduced in Moreau (1988) (Glocker and Studer, 2005; Leine and Nijmeijer, 2004) called the time-stepping method described in the following. By employing this method together with set-valued force laws we avoid having to explicitly switch between equations when impacts occur. Sections 3.6.1 and 3.6.2 are based on Le Saux *et al.* (2005), except for the novel direct calculation of the bilateral contact impulses which we have introduced in Transeth *et al.* (2006a,b).

3.6.1 Time Discretization

Let t_l denote the time at time-step $l = 1, 2, 3, \dots$ where the step size is $\Delta t = t_{l+1} - t_l$. Consider the (usually very short) time interval $I_t = [t_A, t_E]$ and let $t_A = t_l$. Define $\mathbf{q}_A = \mathbf{q}(t_A)$, $\mathbf{u}_A = \mathbf{u}(t_A)$ which are admissible with respect to both the unilateral and bilateral constraints. Our goal is now to find $\mathbf{q}_E = \mathbf{q}(t_E)$ ¹. We use the states of the system at the *mid-point* $t_M = t_A + \frac{1}{2}\Delta t$ of the time-interval I_t to decide which contact points are active (i.e. whether or not any links are touching an obstacle). The coordinates (positions and orientations) at t_M are found from

$$\mathbf{q}_M = \mathbf{q}_A + \frac{\Delta t}{2} \mathbf{u}_A. \quad (3.65)$$

The approximation of the matrices \mathbf{W}_Ξ , where $\Xi = H, T, J$, on the time-interval I_t is given as $\mathbf{W}_{\Xi M} := \mathbf{W}_\Xi(\mathbf{q}_M)$. A numerical approximation of the equality of measures (3.44) over the time-interval I_t can now be written as

$$\mathbf{M}(\mathbf{u}_E - \mathbf{u}_A) - \mathbf{S} - \mathbf{W}_{JM} \mathbf{P}_J = \tau \Delta t, \quad (3.66)$$

where

$$\mathbf{S} = \sum_{b \in \mathcal{H}_M} (\mathbf{W}_{HM})_b P_{H_b} + \mathbf{W}_{TM} \mathbf{P}_T, \quad (3.67)$$

and P_{H_b} , \mathbf{P}_T and \mathbf{P}_J are the contact impulses during the time-interval I_t . They consist of forces λ acting *during* I_t and possible impulses Λ acting *in* the time-interval I_t . To find the positions and orientations \mathbf{q}_E at the end of the time-interval, we need to solve (3.66) for \mathbf{u}_E and the contact impulses. The contact impulses associated with obstacle and ground frictional contact are found using the prox-functions described in Section 3.5.2 for the set of active contact points \mathcal{H}_M found by (3.47) with $t = t_M$. Hence, the constitutive laws (3.64) for the obstacle contact and friction impulses may now be written as

$$P_{H_b} = \text{prox}_{C_H} (P_{H_b} - r_H \gamma_{HE_b}), \quad b \in \mathcal{H}_M, \quad (3.68)$$

$$\mathbf{P}_{T_a} = \text{prox}_{C_T} (\mathbf{P}_{T_a} - r_T \gamma_{TE_a}), \quad a = 1, \dots, n, \quad (3.69)$$

¹In order to clarify the use of notation, we note that the subscript letters A and E refer to the German words ‘Anfang’ which means ‘beginning’, and ‘Ende’ which means ‘end’.

where $r_H, r_T > 0$, γ_{HE_b} is the b -th element of the vector γ_{HE} , γ_{TE_a} is the vector of the $(2a - 1)$ -th and $2a$ -th element of γ_{TE} and

$$\gamma_{HE} := \gamma_H(\mathbf{q}_M, \mathbf{u}_E) = \mathbf{W}_{HM}^T \mathbf{u}_E, \quad (3.70)$$

$$\gamma_{TE} := \gamma_T(\mathbf{q}_M, \mathbf{u}_E) = \mathbf{W}_{TM}^T \mathbf{u}_E. \quad (3.71)$$

The constitutive laws (3.68)-(3.71) are valid for completely inelastic impacts.

The constraints on the joints are bilateral and it therefore holds that

$$\gamma_{JE} := \gamma_J(\mathbf{q}_E, \mathbf{u}_E) = \mathbf{W}_{JM}^T \mathbf{u}_E = \mathbf{0}, \quad \forall t. \quad (3.72)$$

This allows us to directly compute the associated contact impulses \mathbf{P}_J by employing (3.66) and (3.72):

$$\mathbf{P}_J = -(\mathbf{W}_{JM}^T \mathbf{M}^{-1} \mathbf{W}_{JM})^{-1} \mathbf{W}_{JM}^T [\mathbf{u}_A + \mathbf{M}^{-1}(\mathbf{S} + \tau \Delta t)]. \quad (3.73)$$

The equations (3.66)-(3.71) and (3.73) constitute a set of non-smooth equations where the number of unknowns \mathbf{u}_E , P_{H_b} , \mathbf{P}_T and \mathbf{P}_J , equals the number of equations. An algorithm to solve this set of equations is described in Section 3.6.2. After having solved for \mathbf{u}_E we find the position at the end of the time-step as

$$\mathbf{q}_E = \mathbf{q}_M + \frac{\Delta t}{2} \mathbf{u}_E. \quad (3.74)$$

Notice from (3.66)-(3.74) that no explicit change between equations are needed (when for example an impact between the snake robot and an obstacle occurs) since both the non-impulsive forces and the impact impulses are covered by the same constitutive laws (3.68)-(3.69).

3.6.2 Solving for the Contact Impulsions

The numerical integration algorithm used in this chapter is called a time-stepping method which allows for a simultaneous treatment of both impulsive and non-impulsive forces during a time-step. The frictional contact problem, defined by (3.66)-(3.71) and (3.73), needs to be solved for each time-step t_l . A Modified Newton Algorithm (Alart and Curnier, 1991) has been chosen, because of its simplicity, to solve the non-linear problem iteratively. Let the superscript (k) denote the current iteration of the Modified Newton Algorithm and initialize all contact impulses (for active contacts) with the value they had the last time their corresponding contact point was active. Now, the algorithm may be written as:

Algorithm 3.1 (Modified Newton Algorithm for Planar Model)

1. Solve

$$\mathbf{P}_J^{(k)} = - \left(\mathbf{W}_{JM}^T \mathbf{M}^{-1} \mathbf{W}_{JM} \right)^{-1} \mathbf{W}_{JM}^T \left[\mathbf{u}_A + \mathbf{M}^{-1} \left(\mathbf{S}^{(k)} + \tau \Delta t \right) \right], \quad (3.75)$$

where

$$\mathbf{S}^{(k)} = \mathbf{W}_{TM} \mathbf{P}_T^{(k)} + \sum_{b \in \mathcal{H}_M} (\mathbf{W}_{HM})_b P_{H_b}^{(k)}. \quad (3.76)$$

2. Solve $\mathbf{u}_E^{(k+1)}$ from

$$\mathbf{M} \left(\mathbf{u}_E^{(k+1)} - \mathbf{u}_A \right) - \mathbf{S}^{(k)} - \mathbf{W}_{JM} \mathbf{P}_J^{(k)} = \tau \Delta t. \quad (3.77)$$

3. Solve for $a = 1, \dots, n$ and $b \in \mathcal{H}_M$

$$P_{H_b}^{(k+1)} = \text{prox}_{C_H} \left(P_{H_b}^{(k)} - r_H \gamma_{HE_b}^{(k+1)} \right), \quad (3.78)$$

$$\mathbf{P}_{T_a}^{(k+1)} = \text{prox}_{C_T} \left(\mathbf{P}_{T_a}^{(k)} - r_T \gamma_{TE_a}^{(k+1)} \right), \quad (3.79)$$

where

$$\gamma_{HE}^{(k+1)} = \mathbf{W}_{HM}^T \mathbf{u}_E^{(k+1)}, \quad \gamma_{TE}^{(k+1)} = \mathbf{W}_{TM}^T \mathbf{u}_E^{(k+1)}. \quad (3.80)$$

Repeat steps 1. to 3. until

$$\| \mathbf{P}_H^{(k+1)} - \mathbf{P}_H^{(k)} \| + \| \mathbf{P}_T^{(k+1)} - \mathbf{P}_T^{(k)} \| < \epsilon, \quad (3.81)$$

where $\epsilon > 0$ is a user-defined tolerance. Subsequently, \mathbf{q}_E is calculated from (3.74) and the calculation of the time-step is finished. Usually, a higher value of the parameters r_H, r_T yields a faster convergence rate at the risk of divergence. However, a general convergence result for the Modified Newton Algorithm does not exist. Note that the inverse of the mass matrix \mathbf{M} needs to be calculated in step 2. in the algorithm in order to solve for $\mathbf{u}_E^{(k+1)}$. However, the inverse only needs to be calculated once since \mathbf{M} is constant. This is advantageous for the numerical treatment. We give the following remark on the implementation of Algorithm 3.1:

Remark 3.1 *Note that for the implementation of the Modified Newton Algorithm (Algorithm 3.1) it is beneficial to set an upper bound on the number of iterations to avoid an infinite loop. If the algorithm does not converge within e.g. 10 000 iterations, then this suggests that the parameters r_H and r_T should be altered.*

The constitutive laws (3.78)-(3.80) used to describe the contact impulses are given on *velocity level*. This means that the bilateral constraints on position level are in general not satisfied. A solution to this problems is suggested in the following.

3.6.3 Constraint Violation

The contact impulses \mathbf{P}_J are calculated such that the bilateral constraints are satisfied on velocity level. In order to prevent that adjacent links drift apart, we project the positions of the links so that the bilateral constraints are satisfied on position level after the Modified Newton Algorithm has converged and \mathbf{q}_E has been found from (3.74). The projection is performed by keeping the position of link 1 fixed together with the orientations of all links. Then the links are moved so that all the gap functions (3.14) associated with the joints are equal to zero, i.e. $g_{J_{i\chi}} = 0$ for $\chi = x, y$ and $i = 1, \dots, n - 1$, and the vector \mathbf{q}_E is updated with the new positions of the links.

The projection performed to satisfy the bilateral constraints on position level adds additional computations to the numerical treatment of the snake robot model. An alternative approach is to reduce the number of generalized coordinates (from a non-minimal description) by expressing the bilateral constraints analytically and then incorporating them implicitly into the model. However, some sort of projection algorithm will in some cases still be necessary to avoid that the links drift apart depending on the choice of generalized coordinates and the accuracy of the numerical integrator. Another approach would be to develop the model from a set of minimal coordinates directly, but then the system matrices will grow very large when the number of links is increased. With the modelling approach described in this chapter, we avoid having to rewrite the model (from a non-minimal set of coordinates) to a new (minimal) set of coordinates and the system matrices are kept simple. Hence, we have the positions of all links directly available and the appearance of the model makes it easy to understand. Moreover, it is less demanding to extend the model to 3D where an implicit incorporation of the bilateral constraints in the model would prove cumbersome.

3.7 Summary

A novel mathematical 2D model based on the framework of non-smooth dynamics and convex analysis is developed for a snake robot in an envi-

ronment with external objects in this chapter. The model describes planar snake robot motion where the snake robot can push against external obstacles over the entire surface of each link. Hence, the effect of the contact forces between the obstacle and each link is modelled properly and motion patterns which are based on that the snake robot pushes against external objects can be tested with even very small objects. It has been shown how to easily incorporate the normal contact and friction forces into the system dynamics (i.e. the equality of measures) by set-valued force laws. The modelling framework allowed for accurately describing rigid body contact and collisions in a hybrid manner without having to explicitly switch between system equations. Moreover, it has been shown how to include correct directional Coulomb friction. Hence, the model can be employed to simulate various planar motion patterns both with and without external objects to push against. In addition, the concise description of the model and its numerical treatment give others the opportunity to implement the model for simulation.

In addition to presenting the 2D model of the snake robot, this chapter serves as an introduction to the various modelling techniques in the framework of non-smooth dynamics and convex analysis that have been necessary to employ in this thesis. In the following chapters, we will utilize and refer to the general theory given in this chapter.

Chapter 4

3D Snake Robot Modelling for Simulation

4.1 Introduction

Biological snakes may lift parts of their body during locomotion by a form of lateral undulation (called sinus-lifting) to move faster and more efficient. In addition, snakes that travel on a surface with uniform friction may employ a serpentine motion pattern called sidewinding for locomotion. Both of these motion patterns require that parts of the snake robot body are lifted. For this reason, a model of a snake robot capable of such motion patterns is needed to include the possibility of 3D motion.

In this chapter we extend the non-smooth 2D model given in Chapter 3 to 3D motion. We model the snake robot with 2 degrees of freedom (DOF) cardan joints and cylindrical links. The snake robot is set in an environment with vertical cylindrical obstacles that it may push against for locomotion. The general procedure for developing the model is similar to the development of the 2D model. However, different coordinates are chosen and additional constitutive laws need to be added (e.g. a varying normal force between the snake robot and the ground). The summary of the 2D model presented in Section 3.2 can be read as an introduction to the development of the 3D model also since the general approach is the same.

This chapter is based on Transeth *et al.* (2006a), Transeth *et al.* (2006b) and Transeth *et al.* (2008a) and it is organized as follows. The kinematics of the snake robot with the ground surface as a unilateral constraint is described in Section 4.2. Then the groundwork for finding the friction and ground contact forces is laid in Section 4.3. The non-smooth dynamics is

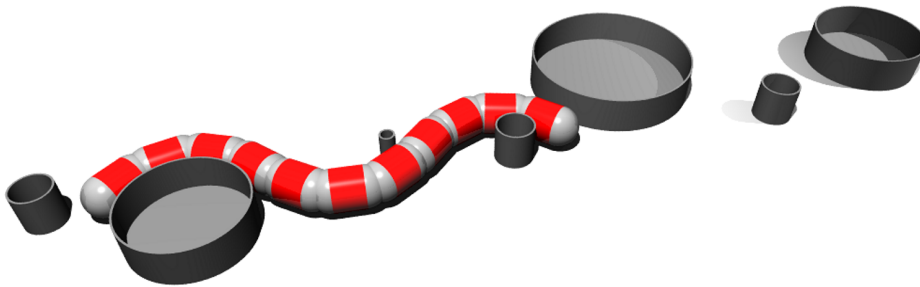


Figure 4.1: 3D model of snake robot among cylindrical obstacles.

presented in Section 4.4. How to calculate and control the joint angles is shown in Section 4.5. The numerical treatment of the model is described in Section 4.6 and a summary of the chapter is given in Section 4.7.

4.2 Kinematics

In this section we describe the kinematics of a 3D model of the snake robot. We employ the kinematics to develop *gap functions* for ground contact detection and obstacle contact detection. These functions are also needed for calculating the directions of the contact forces.

This section will first give an overview of the coordinates used to describe the position and orientation of the snake robot. Subsequently, the gap functions will be presented.

4.2.1 Model Description, Coordinates and Reference Frames

The 3D model of the snake robot ‘Aiko’ consists of n cylindrical links that are connected by $n - 1$ cardan joints, each having two degrees of freedom (DOF). Figure 4.1 illustrates the 3D model with obstacles for the snake robot to push against for locomotion. The distance L_i between two adjacent cardan joints equals the length of link i and the radius of each link is L_{SC_i} . Each link is modelled as a cylinder of length $2L_{GS_i}$ with two spheres of radius L_{SC_i} attached to the ends the link. Half of each sphere is covered by the cylinder. See Figure 4.2 for an illustration of link i together with parts of its two adjacent links. We denote the earth-fixed coordinate frame $I = (O, e_x^I, e_y^I, e_z^I)$, see Figure 4.2, as an approximation to an inertial frame where its origin O is fixed to the ground surface and the e_z^I -axis

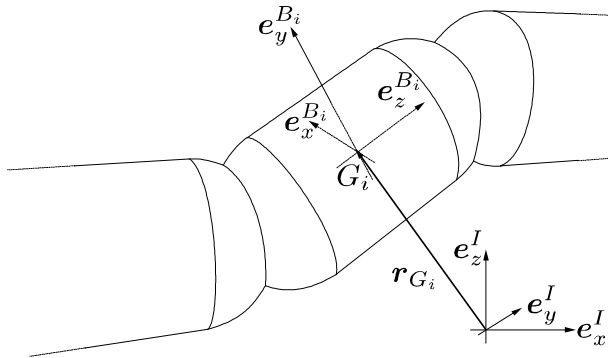


Figure 4.2: Link i with two adjacent links and reference frames.

is pointing in the opposite direction of the acceleration of gravity vector $\mathbf{g} = [0 \ 0 \ -g]^T$. The remaining coordinate axes are defined according to the right-hand convention where plane $(\mathbf{e}_x^I, \mathbf{e}_y^I)$ constitutes the horizontal ground surface. We denote the body-fixed frame $B_i = (G_i, \mathbf{e}_x^{B_i}, \mathbf{e}_y^{B_i}, \mathbf{e}_z^{B_i})$, where G_i is the centre of gravity of link i (which coincides with the geometric centre of the link-cylinder) and $\mathbf{e}_z^{B_i}$ points along the centre line of link i towards link $i+1$. We define $\mathbf{e}_y^{B_i}$ to be parallel to \mathbf{e}_z^I when the link is lying flat on the ground surface and has not rolled sideways. The remaining coordinate axis $\mathbf{e}_x^{B_i}$ is defined according to the right-hand convention.

The environment of the snake robot consists of a horizontal ground surface together with external objects. The external objects (referred to as ‘obstacles’) are included in the model. Each obstacle $j = 1, \dots, \nu$ is modelled as a vertical cylinder of radius L_{H_j} and infinite height with its centre-line parallel to \mathbf{e}_z^I . The vector from O to the point H_j where the mid-line of obstacle j intersects with the $(\mathbf{e}_x^I, \mathbf{e}_y^I)$ -plane is denoted by \mathbf{r}_{H_j} . We see from the above description of the model with obstacles, that the 2D model presented in Chapter 3 is a projection of the 3D model onto the $(\mathbf{e}_x^I, \mathbf{e}_y^I)$ -plane when lying flat on the ground surface. The only difference is the change in definition of the body-fixed coordinate axes (compare Figure 3.2 on page 45 and Figure 4.2).

The position and orientation of link i are described by the *non-minimal absolute* coordinates

$$\mathbf{q}_i = \begin{bmatrix} {}^I \mathbf{r}_{G_i} \\ \mathbf{p}_i \end{bmatrix} \in \mathbb{R}^7, \quad (4.1)$$

where ${}^I \mathbf{r}_{G_i} = [x_i \ y_i \ z_i]^T \in \mathbb{R}^3$ is the position of the centre of gravity of

link i and the vector $\mathbf{p}_i = [e_{i_0} \ \mathbf{e}_i^T]^T$, where $\mathbf{e}_i^T = [e_{i_1} \ e_{i_2} \ e_{i_3}]$, contains the four Euler parameters used to describe the orientation of link i . The Euler parameters form a unit quaternion vector with the constraint $\mathbf{p}_i^T \mathbf{p}_i = 1$. The coordinates are *non-minimal* because each link is described with a non-minimal set of coordinates and *absolute* because the position and orientation of link i is given directly relative to frame I . The velocity of link i is given by

$$\mathbf{u}_i = \begin{bmatrix} {}^I \mathbf{v}_{G_i} \\ {}_{B_i} \boldsymbol{\omega}_{IB_i} \end{bmatrix} \in \mathbb{R}^6, \quad (4.2)$$

where ${}^I \mathbf{v}_{G_i} = [v_{i_x} \ v_{i_y} \ v_{i_z}]^T \in \mathbb{R}^3$ is the translational velocity of the CG of link i which is ${}^I \mathbf{v}_{G_i} = {}^I \dot{\mathbf{r}}_{G_i}$ when it exists (i.e. for impact free motion). Moreover, ${}_{B_i} \boldsymbol{\omega}_{IB_i}$ is the angular velocity of frame B_i relative to frame I , given in frame B_i . The transformation ${}^I \mathbf{r} = \mathbf{R}_{B_i}^I \mathbf{r}$ can be performed with the rotation matrix

$$\mathbf{R}_{B_i}^I = \mathbf{H}_i \bar{\mathbf{H}}_i^T, \quad (4.3)$$

where

$$\mathbf{H}_i = [-\mathbf{e}_i \ \tilde{\mathbf{e}}_i + e_{i_0} \mathbf{I}], \quad \bar{\mathbf{H}}_i = [-\mathbf{e}_i \ -\tilde{\mathbf{e}}_i + e_{i_0} \mathbf{I}], \quad (4.4)$$

and the superscript $\tilde{\cdot}$ denotes the skew-symmetric form of a vector throughout this chapter, i.e.

$$\tilde{\mathbf{e}}_i = \begin{bmatrix} 0 & -e_{i_3} & e_{i_2} \\ e_{i_3} & 0 & -e_{i_1} \\ -e_{i_2} & e_{i_1} & 0 \end{bmatrix}. \quad (4.5)$$

The time-derivative of the rotation matrix is found from Egeland and Gravdahl (2002) as

$$\dot{\mathbf{R}}_{B_i}^I = \mathbf{R}_{B_i}^I {}_{B_i} \tilde{\boldsymbol{\omega}}_{IB_i} = {}^I \tilde{\boldsymbol{\omega}}_{IB_i} \mathbf{R}_{B_i}^I. \quad (4.6)$$

The coordinates (positions and orientation) and velocities of all links are gathered in the vectors

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_1 \\ \vdots \\ \mathbf{q}_n \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_n \end{bmatrix}. \quad (4.7)$$

4.2.2 Gap Functions for Ground Contact

The gap functions for contact between the snake robot links and the ground surface give the minimal distance between the floor and the front and rear

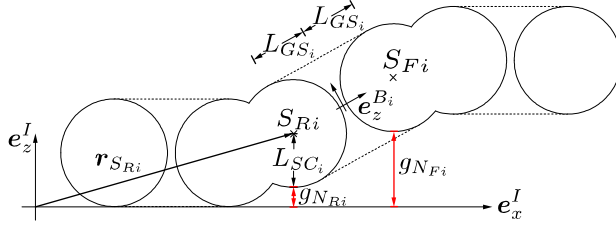


Figure 4.3: Surfaces (solid-drawn circles) on snake robot that constitute the contact between the robot and the ground.

part of each link. We also saw in Example 3.1 on page 46 that the gap functions are used as a basis for calculating the forces involved in the contact.

The contact surfaces between a link and the ground are modelled as two spheres at the ends of the link as illustrated for a three-link robot in Figure 4.3. Hence, we do not consider the cylindrical part of the link for ground contact. Moreover, note that with this assumption the cylindrical part is never penetrated by the ground surfaces since it is assumed that the ground surface is planar.

We denote the distance between the centre of the two spheres that belong to link i by $2L_{GS_i}$ and the radius of the spheres by L_{SC_i} . The position of the centre of the front and rear spheres are denoted by $\mathbf{r}_{S_{Fi}}$ and $\mathbf{r}_{S_{Ri}}$, respectively. The shortest distance (i.e. the gap functions) between the ground and the points on the front and rear spheres closest to the ground are denoted by $g_{N_{Fi}}$ and $g_{N_{Ri}}$, respectively. The distances are found from

$$g_{N_{Fi}} = (\mathbf{r}_{S_{Fi}})^T \mathbf{e}_z^I - L_{SC_i}, \quad g_{N_{Ri}} = (\mathbf{r}_{S_{Ri}})^T \mathbf{e}_z^I - L_{SC_i}, \quad (4.8)$$

where $\mathbf{r}_{S_{Fi}} = \mathbf{r}_{G_i} + L_{GS_i} \mathbf{e}_z^{B_i}$ and $\mathbf{r}_{S_{Ri}} = \mathbf{r}_{G_i} - L_{GS_i} \mathbf{e}_z^{B_i}$.

The gap functions are gathered in the vector

$$\mathbf{g}_N = [g_{N_{F1}} \quad \cdots \quad g_{N_{Fn}} \quad g_{N_{R1}} \quad \cdots \quad g_{N_{Rn}}]^T. \quad (4.9)$$

4.2.3 Gap Functions for Contact with Obstacles

For contact between the snake robot and the obstacles, we consider the entire surface of the link as depicted in Figure 4.2, and not only the two end-spheres employed for ground contact. This results in a contact point that may move across the entire surface of a link. The shortest distance

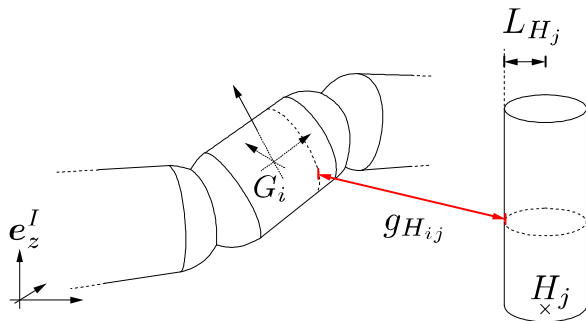


Figure 4.4: Illustration of the shortest distance between link i and obstacle j . The cylindrical part of the link is closest to the obstacle in the picture.

between link i and obstacle j is found with the gap function $g_{H_{ij}}$. Similarly to the 2D model, link i can only touch a convex obstacle j at one point at each time-instance. Hence, by inspecting the shape of the surface of a link, we need to consider the distance between two cylinders, and the distance between a cylinder and a sphere to calculate the shortest distance (i.e. the gap function) between the link and the obstacle, see Figure 4.4.

Cylinder - Cylinder Contact

If the part of link i which is shaped as a cylinder is closest to the obstacle (also being a cylinder), then the gap function for object contact is defined as the distance between these two cylinders. Since it is already established that it is the cylinder part of the link that is closest to the obstacle, we can regard the link cylinder to be of infinite length. The shortest distance between two infinitely long cylinders equals the shortest distance between their centre-lines minus the sum of their respective radii. The normal vector to the centre-lines of both the obstacle j and the link i cylinder is

$$\mathbf{n}_{ij} = \frac{\mathbf{e}_z^I \times \mathbf{e}_z^{B_i}}{\|\mathbf{e}_z^I \times \mathbf{e}_z^{B_i}\|}. \quad (4.10)$$

To find the shortest distance between the two centre-lines, we simply project a vector going from some point on the obstacle centre-line, for example H_j , to some point on the link centre-line, for example G_i , onto the normal vector \mathbf{n}_{ij} in (4.10). Subsequently, the shortest distance between the two centre-lines can be found as

$$d_{ij} = (\mathbf{r}_{G_i} - \mathbf{r}_{H_j})^T \mathbf{n}_{ij}. \quad (4.11)$$

Hence, the gap function for cylinder-cylinder contact is

$$g_{H_{ij}} = |d_{ij}| - (L_{H_j} + L_{SC}). \quad (4.12)$$

Cylinder - Sphere Contact

Consider one of the two hemispheres of link i as closest to obstacle j , then the gap function between the link and the obstacle is

$$g_{H_{ij}} = \|\mathbf{r}_{H_j S_i}\| - (L_{H_j} + L_{SC_i}), \quad (4.13)$$

where $\mathbf{r}_{H_j S_i}$ is a vector from H_j to the *projection of either the point S_{Fi} or the point S_{Ri} onto the $(\mathbf{e}_x^I, \mathbf{e}_y^I)$ -plane* depending on which end of the link is closest to the obstacle. Hence, it holds that

$$\mathbf{r}_{H_j S_i} = \begin{cases} \mathbf{A}_{xy} (\mathbf{r}_{G_i} + \mathbf{r}_{G_i S_{Fi}}) - \mathbf{r}_{H_j}, & \text{front part of link } i \text{ is closest,} \\ \mathbf{A}_{xy} (\mathbf{r}_{G_i} + \mathbf{r}_{G_i S_{Ri}}) - \mathbf{r}_{H_j}, & \text{rear part of link } i \text{ is closest,} \end{cases} \quad (4.14)$$

where

$$\mathbf{r}_{G_i S_{Fi}} = L_{GS_i} \mathbf{e}_z^{B_i}, \quad \mathbf{r}_{G_i S_{Ri}} = -L_{GS_i} \mathbf{e}_z^{B_i}, \quad \mathbf{A}_{xy} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (4.15)$$

Vector of Gap functions for Contact with External Objects

We now gather the gap functions $g_{H_{ij}}$ for all n links and ν vector

$$\mathbf{g}_H = [g_{H_{11}} \quad \cdots \quad g_{H_{n1}} \quad g_{H_{12}} \quad \cdots \quad g_{H_{n2}} \quad \cdots \quad g_{H_{1\nu}} \quad \cdots \quad g_{H_{n\nu}}]^\top, \quad (4.16)$$

where the elements $g_{H_{ij}}$ are found from either (4.12) or (4.13) depending on which part of link i is closest to obstacle j .

4.2.4 Gap Functions for Bilateral Constraints

Each 2 DOF cardan joint introduces four bilateral constraints. To include the bilateral constraint forces in the model, we first need to develop gap functions that describe the ‘gap’ in the joint.

To find the translational ‘gap’ in the joints, we need to relate the position of the joint between link i and $i+1$ to both of these links. Then we will get a non-zero value from the gap function if the adjacent links drift apart. Let the position of the joint between link i and $i+1$ be written relative to these two links as ${}^I \mathbf{r}_{J_{Fi}} = {}^I \mathbf{r}_{G_i} + \frac{1}{2} L_i {}^I \mathbf{e}_z^{B_i}$ and ${}^I \mathbf{r}_{J_{Ri+1}} = {}^I \mathbf{r}_{G_{i+1}} - \frac{1}{2} L_{i+1} {}^I \mathbf{e}_z^{B_{i+1}}$,

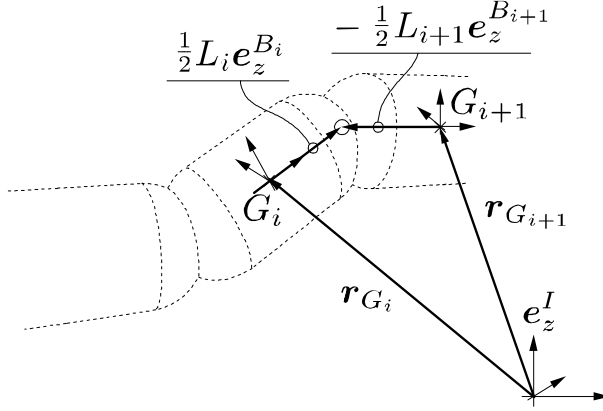


Figure 4.5: Illustration of how the translational gaps in the joint between link i and link $i + 1$ is found.

respectively. We see from Figure 4.5 that $\mathbf{r}_{J_{F_i}} = \mathbf{r}_{J_{R_{i+1}}}$ when the two adjacent links are in a position and orientation relative to each other that is allowed by the cardan joint which connects the links. The gap functions for the translational gaps can now be found from

$$g_{J_{i\chi}} = \left(I \mathbf{r}_{J_{F_i}} - I \mathbf{r}_{J_{R_{i+1}}} \right)^T I \mathbf{e}_{\chi}^I, \quad (4.17)$$

for $i = 1, \dots, n - 1$, and $\chi = x, y, z$. Hence, from (4.17) we found that

$$g_{J_{i\chi}} = \left(I \mathbf{r}_{G_i} - I \mathbf{r}_{G_{i+1}} \right)^T I \mathbf{e}_{\chi}^I + \frac{1}{2} \left(L_i \mathbf{R}_{B_i B_i}^I \mathbf{e}_z^{B_i} + L_{i+1} \mathbf{R}_{B_{i+1} B_{i+1}}^I \mathbf{e}_z^{B_{i+1}} \right)^T I \mathbf{e}_{\chi}^I, \quad (4.18)$$

for $i = 1, \dots, n - 1$, and $\chi = x, y, z$.

The next gap function provides the ‘gap’ in rotation around the axis that the cardan joint is not able to rotate around (one can only perform pitch and yaw movement with a cardan joint and not roll). Let $\mathbf{e}_y^{B_i}$ and $\mathbf{e}_x^{B_{i+1}}$ be the axes of rotation for the cardan joint between link i and link $i + 1$, then

$$\left(\mathbf{e}_y^{B_i} \right)^T \mathbf{e}_x^{B_{i+1}} = 0. \quad (4.19)$$

Hence, a measure for the rotational ‘gap’ can be defined from (4.19) as the gap function

$$g_{J_{i\phi}} = \left(\mathbf{R}_{B_i B_i}^I \mathbf{e}_y^{B_i} \right)^T \left(\mathbf{R}_{B_{i+1} B_{i+1}}^I \mathbf{e}_x^{B_{i+1}} \right). \quad (4.20)$$

4.3 Contact Constraints on Velocity Level

In this section we calculate the relative velocities between the snake robot and the ground from the gap functions. The relative velocities are needed to set up the set-valued contact forces for the closed contacts (Le Saux *et al.*, 2005).

4.3.1 Unilateral Contact: Ground Contact

Contact between the snake robot and the ground surface involves (vertical) normal forces, which guarantee the unilaterality of the contact, and (horizontal) tangential contact forces, which are due to friction and are dependent on the normal contact forces and the relative sliding velocities. We assumed that the normal contact forces between the snake robot and the ground surface were constant in the 2D model in Chapter 3. For the 3D model, this normal contact force is varying and we need to find the corresponding relative velocities to calculate it.

Relative Velocities Along e_z^I

The relative velocities between the front and rear part of link i and the ground along the e_z^I -axis are defined as $\gamma_{N_{F_i}} := \dot{g}_{N_{F_i}}$ and $\gamma_{N_{R_i}} := \dot{g}_{N_{R_i}}$ (when they exist), respectively, and they are used later to find the normal contact forces. Before we proceed, note that $\gamma_{N_{F_i}}$ (or $\gamma_{N_{R_i}}$) should not be found directly by taking the time-derivative of the expression for $g_{N_{F_i}}$ (or $g_{N_{R_i}}$) in (4.8). This is the case since the expressions (4.8) have already been simplified due to the fact that expressions have been inserted for the various body-fixed vectors which constitute the gap functions. Instead, for the relative velocities, we must consider the velocity of the *body-fixed* point P_{F_i} which at time-instant t coincides with the point C_{F_i} on the front sphere closest to the ground (the same principle applies for the rear part of the link). Note that the position vectors for P_{F_i} and C_{F_i} will be the same instantaneously. However, the *differentials* will be different. This is shown in the following. Let C_{F_i} be a point on the front sphere that moves on the sphere such that the point is always closest to the ground surface. Then the position of this point is

$$\mathbf{r}_{C_{F_i}} = \mathbf{r}_{G_i} + \mathbf{r}_{G_i S_{F_i}} + \mathbf{r}_{S_{F_i} C_{F_i}}. \quad (4.21)$$

The velocity of the point \tilde{C}_{Fi} is obtained by differentiation of (4.21):

$$I\mathbf{v}_{C_{Fi}} = I\mathbf{v}_{G_i} - \underbrace{(I\tilde{\mathbf{r}}_{G_i S_{Fi}} + I\tilde{\mathbf{r}}_{S_{Fi} C_{Fi}})}_{I\mathbf{v}_{P_{Fi}}} \mathbf{R}_{B_i B_i}^I \boldsymbol{\omega}_{IB_i} + \mathbf{R}_{B_i B_i}^I \dot{\mathbf{r}}_{S_{Fi} C_{Fi}}, \quad (4.22)$$

where we now can use that

$$\mathbf{r}_{S_{Fi} C_{Fi}} = -L_{SC_i} \mathbf{e}_z^I, \quad (4.23)$$

and we have employed (4.6) and $B_i \dot{\mathbf{r}}_{G_i S_{Fi}} = 0$, together with the identities $I\tilde{\boldsymbol{\omega}}_{IB_i} I\mathbf{r}_{G_i S_{Fi}} = -I\boldsymbol{\omega}_{IB_i} I\tilde{\mathbf{r}}_{G_i S_{Fi}}$ and $I\tilde{\boldsymbol{\omega}}_{IB_i} I\mathbf{r}_{S_{Fi} C_{Fi}} = -I\boldsymbol{\omega}_{IB_i} I\tilde{\mathbf{r}}_{S_{Fi} C_{Fi}}$. We see from (4.22) the expression for the velocity $I\mathbf{v}_{P_{Fi}}$ of the body-fixed point P_{Fi} which at time-instance t coincides with the point C_{Fi} . This velocity will be employed to find both the relative velocities along \mathbf{e}_z^I and the tangential relative velocities. The equivalent velocity $I\mathbf{v}_{P_{Ri}}$ on the rear part of the sphere is found similarly to $I\mathbf{v}_{P_{Fi}}$ in (4.22) by replacing Fi by Ri in (4.22)-(4.23).

Now, the relative velocities along \mathbf{e}_z^I for the front and rear part of link i can be found as

$$\gamma_{N_{Qi}} = (I\mathbf{e}_z^I)^T I\mathbf{v}_{P_{Qi}} \implies \gamma_{N_{Qi}} = (\mathbf{w}_{N_{Qi}})^T \mathbf{u}_i, \quad (4.24)$$

where

$$\mathbf{w}_{N_{Qi}} = [(I\mathbf{e}_z^I)^T \quad -(I\mathbf{e}_z^I)^T (I\tilde{\mathbf{r}}_{G_i S_{Qi}} + I\tilde{\mathbf{r}}_{S_{Qi} C_{Qi}}) \mathbf{R}_{B_i}^I]^T, \quad (4.25)$$

for $Q = F, R$ with $\mathbf{r}_{G_i S_{Fi}} = L_{GS_i} \mathbf{e}_z^{B_i}$ and $\mathbf{r}_{G_i S_{Ri}} = -L_{GS_i} \mathbf{e}_z^{B_i}$. The motivation to use the form (4.24) is that $\mathbf{w}_{N_{Qi}}$ gives the generalized direction of the contact force between the ground and the front and rear part of link i .

A vector gathering all $\gamma_{N_{Fi}}$ and $\gamma_{N_{Ri}}$ is

$$\boldsymbol{\gamma}_N = \mathbf{W}_N^T \mathbf{u}, \quad (4.26)$$

where

$$\boldsymbol{\gamma}_N = [\gamma_{N_{F1}} \quad \cdots \quad \gamma_{N_{Fn}} \quad \gamma_{N_{R1}} \quad \cdots \quad \gamma_{N_{Rn}}]^T, \quad (4.27)$$

$$\mathbf{W}_N = [\mathbf{W}_{NF} \quad \mathbf{W}_{NR}] \in \mathbb{R}^{6n \times 2n}, \quad (4.28)$$

and

$$\mathbf{W}_{N_Q} = \begin{bmatrix} \mathbf{w}_{N_{Q1}} & \mathbf{0}_{6 \times 1} & \cdots & \mathbf{0}_{6 \times 1} \\ \mathbf{0}_{6 \times 1} & \ddots & & \vdots \\ \vdots & & \ddots & \mathbf{0}_{6 \times 1} \\ \mathbf{0}_{6 \times 1} & \cdots & \mathbf{0}_{6 \times 1} & \mathbf{w}_{N_{Qn}} \end{bmatrix} \in \mathbb{R}^{6n \times n}, \quad (4.29)$$

for $Q = F, R$.

Tangential Relative Velocities

The friction forces between a link and the ground depend on the relative velocities in the $(\mathbf{e}_x^I, \mathbf{e}_y^I)$ -plane between the snake robot links and the ground. These velocities are termed tangential relative velocities. First, the relative velocities between the front part of link i and the ground along the \mathbf{e}_x^I - and \mathbf{e}_y^I -axis, $\gamma'_{T_{Fiz}}$ and $\gamma'_{T_{Fiy}}$, will be deducted. Subsequently, the relative velocities for the front part of link i along the projection of the longitudinal axis of the link onto the $(\mathbf{e}_x^I, \mathbf{e}_y^I)$ -plane $\gamma_{T_{Fiz}}$ and transversal to the link $\gamma_{T_{Fiy}}$, will be deducted from $\gamma'_{T_{Fiz}}$ and $\gamma'_{T_{Fiy}}$ respectively. The same type of notation applies for the rear part of link i : $\gamma'_{T_{Riz}}$, $\gamma'_{T_{Riy}}$, $\gamma_{T_{Riz}}$ and $\gamma_{T_{Riy}}$. The tangential relative velocities are found much in the same way as for $\gamma_{N_{Fi}}$ and $\gamma_{N_{Ri}}$. Consequently, we find the velocities of the points P_{Fi} and P_{Ri} along the \mathbf{e}_x^I - and \mathbf{e}_y^I -axis. Hence, by looking at (4.24), we see that the tangential relative velocities of the contact points on the front part of the link can be found as

$$\gamma'_{T_{Fi\zeta}} = ({}_I \mathbf{e}_\zeta^I)^\top {}_I \mathbf{v}_{P_{Fi}} \implies \gamma'_{T_{Fi\zeta}} = (\mathbf{w}'_{T_{Fi\zeta}})^\top \mathbf{u}_i, \quad (4.30)$$

for $\zeta = x, y$, where

$$\mathbf{w}'_{T_{Fi\zeta}} = [({}_I \mathbf{e}_\zeta^I)^\top \quad -({}_I \mathbf{e}_\zeta^I)^\top ({}_I \tilde{\mathbf{r}}_{G_i S_{Fi}} + {}_I \tilde{\mathbf{r}}_{S_{Fi} C_{Fi}}) \mathbf{R}_{B_i}^I]^\top. \quad (4.31)$$

The relative velocities between the rear part of the link and the ground are found by exchanging ${}_I \mathbf{v}_{P_{Fi}}$ with ${}_I \mathbf{v}_{P_{Ri}}$ in (4.30):

$$\gamma'_{T_{Ri\zeta}} = ({}_I \mathbf{e}_\zeta^I)^\top {}_I \mathbf{v}_{P_{Ri}} \implies \gamma'_{T_{Ri\zeta}} = (\mathbf{w}'_{T_{Ri\zeta}})^\top \mathbf{u}_i, \quad (4.32)$$

for $\zeta = x, y$, where

$$\mathbf{w}'_{T_{Ri\zeta}} = [({}_I \mathbf{e}_\zeta^I)^\top \quad -({}_I \mathbf{e}_\zeta^I)^\top ({}_I \tilde{\mathbf{r}}_{G_i S_{Ri}} + {}_I \tilde{\mathbf{r}}_{S_{Ri} C_{Ri}}) \mathbf{R}_{B_i}^I]^\top. \quad (4.33)$$

The tangential relative velocities for the front and rear part of the link i are combined in vectors:

$$\gamma'_{T_{Qi}} = \mathbf{W}'_{T_{Qi}}{}^\top \mathbf{u}_i, \quad (4.34)$$

for $Q = F, R$, where $\gamma'_{T_{Qi}} = [\gamma'_{T_{Qix}} \quad \gamma'_{T_{Qiy}}]^\top$ and

$$\mathbf{W}'_{T_{Qi}} = [\mathbf{w}'_{T_{Qix}} \quad \mathbf{w}'_{T_{Qiy}}] \in \mathbb{R}^{6 \times 2}. \quad (4.35)$$

Until now the relative velocities have been given in the directions along the \mathbf{e}_x^I and \mathbf{e}_y^I axes. In order to calculate the friction forces longitudinal and

transversal to a link, we need to know the corresponding relative velocities in these directions in the (e_x^I, e_y^I) -plane. To calculate these velocities we introduce for each link a frame Π_i with axes $(e_x^{\Pi_i}, e_y^{\Pi_i}, e_z^{\Pi_i})$, where ${}_I e_z^{\Pi_i} = {}_I e_z^I$, and

$${}_I e_x^{\Pi_i} = \frac{\mathbf{A}_{xy} {}_I e_z^{B_i}}{\|\mathbf{A}_{xy} {}_I e_z^{B_i}\|}, \quad {}_I e_y^{\Pi_i} = \frac{{}_I e_z^I \times {}_I e_x^{\Pi_i}}{\|{}_I e_z^I \times {}_I e_x^{\Pi_i}\|}, \quad (4.36)$$

where $\mathbf{A}_{xy} = \text{diag}([1, 1, 0])$. Hence, it holds that

$$\mathbf{R}_{\Pi_i}^I = [{}_I e_x^{\Pi_i} \quad {}_I e_y^{\Pi_i} \quad {}_I e_z^{\Pi_i}]. \quad (4.37)$$

Notice that ${}_I e_x^{\Pi_i} = {}_I e_z^{B_i}$ and ${}_I e_y^{\Pi_i} = {}_I e_x^{B_i}$ when link i is lying flat on the ground with ${}_I e_y^{B_i} = {}_I e_z^I$. Since only the motion in the (e_x^I, e_y^I) -plane is of interest, we define

$$\bar{\mathbf{R}}_{\Pi_i}^I = \mathbf{D}^T \mathbf{R}_{\Pi_i}^I \mathbf{D}, \quad \mathbf{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}^T. \quad (4.38)$$

Since we now have that $\gamma'_{T_{Q_i}} = \bar{\mathbf{R}}_{\Pi_i}^I \gamma_{T_{Q_i}}$, $Q = F, R$, the relative velocity between the floor and the front part of link i , with respect to frame Π_i , can be found from (4.34) as

$$\gamma_{T_{Q_i}} = \mathbf{W}_{T_{Q_i}}^T \mathbf{u}_i, \quad (4.39)$$

where $\gamma_{T_{Q_i}} = [\gamma_{T_{Q_{ix}}} \quad \gamma_{T_{Q_{iy}}}]^T$, $\mathbf{W}_{T_{Q_i}} = \mathbf{W}'_{T_{Q_i}} \bar{\mathbf{R}}_{\Pi_i}^I$, for $Q = F, R$.

A vector that gathers all $\gamma_{T_{F_i}}$ and $\gamma_{T_{R_i}}$ is found from

$$\gamma_T = \mathbf{W}_T^T \mathbf{u}, \quad (4.40)$$

where

$$\gamma_T = \begin{bmatrix} \gamma_{T_{F1}} \\ \vdots \\ \gamma_{T_{F_n}} \\ \gamma_{T_{R1}} \\ \vdots \\ \gamma_{T_{R_n}} \end{bmatrix} \in \mathbb{R}^{4n}, \quad \mathbf{W}_T = [\mathbf{W}_{T_F} \quad \mathbf{W}_{T_R}] \in \mathbb{R}^{6n \times 4n}, \quad (4.41)$$

and

$$\mathbf{W}_{T_Q} = \begin{bmatrix} \mathbf{W}_{T_{Q1}} & \mathbf{0}_{6 \times 2} & \cdots & \mathbf{0}_{6 \times 2} \\ \mathbf{0}_{6 \times 2} & \ddots & & \vdots \\ \vdots & & \ddots & \mathbf{0}_{6 \times 2} \\ \mathbf{0}_{6 \times 2} & \cdots & \mathbf{0}_{6 \times 2} & \mathbf{W}_{T_{Q_n}} \end{bmatrix} \in \mathbb{R}^{6n \times 2n}, \quad \text{for } Q = F, R. \quad (4.42)$$

Relative Rolling Velocities

Up until now we have only considered translational relative motion of the snake robot links. However, we also need to consider rotational relative motion to add a damping effect on the rotational motion in the form of rotational friction. In order to do this, we introduce the relative rolling velocity as

$$\boldsymbol{\gamma}_{V_{Q_i}} = \mathbf{D}^T ({}^I \mathbf{r}_{C_{Q_i} S_{Q_i}} \times {}^I \boldsymbol{\omega}_{B_i}), \quad (4.43)$$

for $Q = F, R$, where $\boldsymbol{\gamma}_{V_{Q_i}} = [\gamma_{V_{Q_ix}} \quad \gamma_{V_{Q_iy}}]^T$, ${}^I \mathbf{r}_{C_{Q_i} S_{Q_i}} = L_{SC_i} {}^I \mathbf{e}_z^I$ is the vector pointing (upward) from the contact point C_{Q_i} on the link end-sphere closest to the ground, toward the centre S_{Q_i} of the end-sphere and \mathbf{D} is given in (4.38). By employing the identity

$${}^I \mathbf{r}_{C_{Q_i} S_{Q_i}} \times {}^I \boldsymbol{\omega}_{B_i} = I \tilde{\mathbf{r}}_{C_{Q_i} S_{Q_i}} \mathbf{R}_{B_i B_i}^I \boldsymbol{\omega}_{B_i},$$

we find that the relative rolling velocity can be written

$$\boldsymbol{\gamma}_{V_{Q_i}} = \mathbf{W}_{V_{Q_i}}^T \mathbf{u}_i, \quad (4.44)$$

where

$$\mathbf{W}_{V_{Q_i}}^T = [\mathbf{O}_{2 \times 3} \quad \mathbf{D}^T I \tilde{\mathbf{r}}_{C_{Q_i} S_{Q_i}} \mathbf{R}_{B_i}^I], \quad (4.45)$$

for $Q = F, R$.

We gather all the relative rolling velocities as

$$\boldsymbol{\gamma}_V = \mathbf{W}_V \mathbf{u}, \quad (4.46)$$

where

$$\boldsymbol{\gamma}_V = \begin{bmatrix} \gamma_{V_{F1}} \\ \vdots \\ \gamma_{V_{Fn}} \\ \gamma_{V_{R1}} \\ \vdots \\ \gamma_{V_{Rn}} \end{bmatrix} \in \mathbb{R}^{4n}, \quad \mathbf{W}_V = [\mathbf{W}_{V_F} \quad \mathbf{W}_{V_R}] \in \mathbb{R}^{6n \times 4n}, \quad (4.47)$$

and \mathbf{W}_{V_F} , \mathbf{W}_{V_R} are found similarly to (4.42) by replacing $\mathbf{W}_{T_{F_i}}$ and $\mathbf{W}_{T_{R_i}}$ in (4.42) by $\mathbf{W}_{V_{F_i}}$ and $\mathbf{W}_{V_{R_i}}$, respectively.

4.3.2 Unilateral Contact: Obstacle Contact

Contact between link i and obstacle j is modelled as a unilateral contact. To calculate the contact force involved in the contact, the relative velocity $\gamma_{H_{ij}} := \dot{g}_{H_{ij}}$ between obstacle j and the point on link i closest to the obstacle needs to be found. This is obtained from the gap functions defined in Section 4.2.3. Equivalently to the calculations of the gap functions, the relative velocity $\gamma_{H_{ij}}$ is dependent on which part of the link is closest to the obstacle.

Cylinder - Cylinder Contact

If the cylinder part of link i is closest to obstacle j , then the gap function is given by (4.12) and the relative velocity $\gamma_{H_{ij}} := \dot{g}_{H_{ij}}$ can be written as

$$\gamma_{H_{ij}} = \left({}_I \mathbf{v}_{G_i}^T \mathbf{n}_{ij} + (\mathbf{r}_{G_i} - \mathbf{r}_{H_j})^T \dot{\mathbf{n}}_{ij} \right) \text{sign}(d_{ij}), \quad (4.48)$$

where \mathbf{n}_{ij} and d_{ij} are found from (4.10) and (4.11), respectively. Define $\|\cdot\| := \|\cdot\|_2$ and consider

$${}_I \dot{\mathbf{n}}_{ij} = - \frac{\frac{d}{dt} \|{}_I \mathbf{e}_z^I \times {}_I \mathbf{e}_z^{B_i}\|}{\|{}_I \mathbf{e}_z^I \times {}_I \mathbf{e}_z^{B_i}\|^2} ({}_I \mathbf{e}_z^I \times {}_I \mathbf{e}_z^{B_i}) + \frac{\frac{d}{dt} ({}_I \mathbf{e}_z^I \times {}_I \mathbf{e}_z^{B_i})}{\|{}_I \mathbf{e}_z^I \times {}_I \mathbf{e}_z^{B_i}\|}, \quad (4.49)$$

where

$$\frac{d}{dt} ({}_I \mathbf{e}_z^I \times {}_I \mathbf{e}_z^{B_i}) = -{}_I \tilde{\mathbf{e}}_z^I \mathbf{R}_{B_i B_i}^I \tilde{\mathbf{e}}_z^{B_i} \boldsymbol{\omega}_{IB_i}, \quad (4.50)$$

$$\frac{d}{dt} \|{}_I \mathbf{e}_z^I \times {}_I \mathbf{e}_z^{B_i}\| = \frac{-({}_I \mathbf{e}_z^{B_i})^T \mathbf{A}_{xy} \mathbf{R}_{B_i B_i}^I \tilde{\mathbf{e}}_z^{B_i} \boldsymbol{\omega}_{IB_i}}{\|{}_I \mathbf{e}_z^I \times {}_I \mathbf{e}_z^{B_i}\|}, \quad (4.51)$$

and $\tilde{\mathbf{e}}_z^I$ denotes the skew-symmetric form of \mathbf{e}_z^I similarly to (4.5) on page 70. Substitution of (4.49)-(4.51) into (4.48) yields

$$\gamma_{H_{ij}} = \mathbf{w}_{H_{ij}}^T \mathbf{u}_i, \quad (4.52)$$

where

$$\mathbf{w}_{H_{ij}} = \text{sign}(d_{ij}) \cdot \left[({}_I \mathbf{n}_{ij})^T \quad \mathbf{c}_{ij}^T \mathbf{R}_{B_i B_i}^I \tilde{\mathbf{e}}_z^{B_i} \right]^T, \quad (4.53)$$

and

$$\mathbf{c}_{ij}^T = \frac{({}_I \mathbf{r}_{G_i} - {}_I \mathbf{r}_{H_j})^T}{\|{}_I \mathbf{e}_z^I \times {}_I \mathbf{e}_z^{B_i}\|} \left(\frac{{}_I \mathbf{n}_{ij} ({}_I \mathbf{e}_z^{B_i})^T \mathbf{A}_{xy}}{\|{}_I \mathbf{e}_z^I \times {}_I \mathbf{e}_z^{B_i}\|} - {}_I \tilde{\mathbf{e}}_z^I \right). \quad (4.54)$$

The motivation to write the relative velocity in the form (4.52) is that the vector $\mathbf{w}_{H_{ij}} \in \mathbb{R}^6$ constitutes the generalized force direction of the contact force between obstacle j and link i .

Cylinder - Sphere Contact

The gap function in (4.13) is employed to define the relative velocity $\gamma_{H_{ij}} := \dot{g}_{H_{ij}}$ when one of the two hemispheres on link i is closest to the obstacle j . In order to find $\gamma_{H_{ij}}$, consider the vector ${}^I\mathbf{r}_{H_j S_i}$ in (4.14) which is a vector from H_j to the projection of either the point S_{Fi} or the point S_{Ri} onto the $(\mathbf{e}_x^I, \mathbf{e}_y^I)$ -plane depending on which end of the link is closest to the obstacle. The time-derivative of ${}^I\mathbf{r}_{H_j S_i}$ is

$${}^I\dot{\mathbf{r}}_{H_j S_i} = \mathbf{A}_{xy} ({}^I\mathbf{v}_{G_i} - \mathbf{A}_{xy} \mathbf{R}_{B_i}^I \tilde{\mathbf{r}}_{G_i S_i} {}_{B_i}\boldsymbol{\omega}_{IB_i}),$$

where ${}_{B_i}\tilde{\mathbf{r}}_{G_i S_i}$ denotes the skew-symmetric form of ${}_{B_i}\mathbf{r}_{G_i S_i}$ and

$$\mathbf{r}_{G_i S_i} = \begin{cases} \mathbf{r}_{G_i S_{Fi}} & , \text{ front part of link } i \text{ is closest,} \\ \mathbf{r}_{G_i S_{Ri}} & , \text{ rear part of link } i \text{ is closest.} \end{cases} \quad (4.55)$$

We now find the relative velocity between obstacle j and the point closest to the obstacle on the front or rear sphere of link i as

$$\gamma_{H_{ij}} = \mathbf{w}_{H_{ij}}^T \mathbf{u}_i, \quad (4.56)$$

where

$$\mathbf{w}_{H_{ij}}^T = \frac{{}^I\mathbf{r}_{H_j S_i}^T}{\|{}^I\mathbf{r}_{H_j S_i}\|} [\mathbf{A}_{xy} \quad -\mathbf{A}_{xy} \mathbf{R}_{B_i}^I \tilde{\mathbf{r}}_{G_i S_i}]. \quad (4.57)$$

Vector of Relative Velocities for the Obstacle Contact

We now gather the relative velocities $\gamma_{H_{ij}}$ for all n links and ν obstacles in the vector

$$\boldsymbol{\gamma}_H = \mathbf{W}_H^T \mathbf{u}, \quad (4.58)$$

where

$$\boldsymbol{\gamma}_H = [\gamma_{H_{11}} \quad \cdots \quad \gamma_{H_{n1}} \quad \gamma_{H_{12}} \quad \cdots \quad \gamma_{H_{n2}} \quad \cdots \quad \gamma_{H_{1\nu}} \quad \cdots \quad \gamma_{H_{n\nu}}]^T, \quad (4.59)$$

$$\mathbf{W}_H = [\mathbf{W}_{H_1} \quad \cdots \quad \mathbf{W}_{H_\nu}] \in \mathbb{R}^{6n \times n\nu}, \quad (4.60)$$

and

$$\mathbf{W}_{H_j} = \begin{bmatrix} \mathbf{w}_{H_{1j}} & \mathbf{0}_{6 \times 1} & \cdots & \mathbf{0}_{6 \times 1} \\ \mathbf{0}_{6 \times 1} & \ddots & & \vdots \\ \vdots & & \ddots & \mathbf{0}_{6 \times 1} \\ \mathbf{0}_{6 \times 1} & \cdots & \mathbf{0}_{6 \times 1} & \mathbf{w}_{H_{nj}} \end{bmatrix} \in \mathbb{R}^{6n \times n}. \quad (4.61)$$

Depending on which part of the link is closest to the obstacle, $\mathbf{w}_{H_{ij}}$ is obtained from either (4.53) or (4.57).

4.3.3 Bilateral Constraints: Joints

The bilateral constraints introduced by the cardan joint between two adjacent links prohibit relative motion between the links in four DOF. The relative velocities between the links along these DOF need to be found in order to calculate the bilateral constraint forces in the joints. These relative velocities are found from the gap functions $g_{J_{i\chi}}$, $\chi = x, y, z$, in (4.18) and $g_{J_{i\phi}}$ in (4.20) for $i = 1, \dots, n - 1$.

The relative velocities for the translational gap between link i and link $i + 1$ are defined as $\gamma_{J_{i\chi}} := \dot{g}_{J_{i\chi}}$ for $i = 1, \dots, n - 1$ where $\chi = x, y, z$. By employing (4.18), we find that

$$\gamma_{J_{i\chi}} = \mathbf{w}_{J_{i\chi}}^T \begin{bmatrix} \mathbf{u}_i \\ \mathbf{u}_{i+1} \end{bmatrix}, \quad (4.62)$$

where

$$\mathbf{w}_{J_{i\chi}} = \begin{bmatrix} (I \mathbf{e}_\chi^I) \\ - \left((I \mathbf{e}_\chi^I)^T \frac{L_i}{2} \mathbf{R}_{B_i B_i}^I \tilde{\mathbf{e}}_z^{B_i} \right)^T \\ - (I \mathbf{e}_\chi^I) \\ - \left((I \mathbf{e}_\chi^I)^T \frac{L_{i+1}}{2} \mathbf{R}_{B_{i+1} B_{i+1}}^I \tilde{\mathbf{e}}_z^{B_{i+1}} \right)^T \end{bmatrix}. \quad (4.63)$$

The relative velocity for the rotational gap is defined as $\gamma_{J_{i\phi}} := \dot{g}_{J_{i\phi}}$ for $i = 1, \dots, n - 1$. Thus,

$$\gamma_{J_{i\phi}} = \mathbf{w}_{J_{i\phi}}^T \begin{bmatrix} \mathbf{u}_i \\ \mathbf{u}_{i+1} \end{bmatrix}, \quad (4.64)$$

where

$$\mathbf{w}_{J_{i\phi}} = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ - \left(\mathbf{R}_{B_i B_i}^I \tilde{\mathbf{e}}_y^{B_i} \right)^T \left(\mathbf{R}_{B_{i+1} B_{i+1}}^I \mathbf{e}_x^{B_{i+1}} \right) \\ \mathbf{0}_{3 \times 1} \\ - \left(\mathbf{R}_{B_{i+1} B_{i+1}}^I \tilde{\mathbf{e}}_x^{B_{i+1}} \right)^T \left(\mathbf{R}_{B_i B_i}^I \mathbf{e}_y^{B_i} \right) \end{bmatrix}. \quad (4.65)$$

Let $\boldsymbol{\gamma}_{J_i} = \left[\gamma_{J_{ix}} \quad \gamma_{J_{iy}} \quad \gamma_{J_{iz}} \quad \gamma_{J_{i\phi}} \right]^T \in \mathbb{R}^4$, then we can gather all the relative velocities concerned with the bilateral constraints in one vector

$$\boldsymbol{\gamma}_J = \mathbf{W}_J^T \mathbf{u}, \quad (4.66)$$

where

$$\boldsymbol{\gamma}_J = \begin{bmatrix} \gamma_{J_1} \\ \vdots \\ \gamma_{J_{n-1}} \end{bmatrix} \in \mathbb{R}^{4(n-1)}, \quad (4.67)$$

$$\mathbf{W}_J = \begin{bmatrix} \mathbf{W}_{J_1}^T & \mathbf{0}_{4 \times 6} & \cdots & \mathbf{0}_{4 \times 6} \\ \mathbf{0}_{4 \times 6} & \mathbf{W}_{J_2}^T & & \vdots \\ \vdots & & \ddots & \mathbf{0}_{4 \times 6} \\ \mathbf{0}_{4 \times 6} & \cdots & \mathbf{0}_{4 \times 6} & \mathbf{W}_{J_{n-1}}^T \end{bmatrix}^T \in \mathbb{R}^{6n \times 4(n-1)}, \quad (4.68)$$

and $\mathbf{W}_{J_i} = [\mathbf{w}_{J_{ix}} \quad \mathbf{w}_{J_{iy}} \quad \mathbf{w}_{J_{iz}} \quad \mathbf{w}_{J_{i\phi}}] \in \mathbb{R}^{12 \times 4}$ for $i = 1, \dots, n-1$.

4.4 Non-smooth Dynamics

In this section we first present the various components of the *equality of measures* for the 3D model. The equality of measures describes the dynamics of the snake robot as seen in Section 3.5.

4.4.1 The Equality of Measures

Consider the generalized velocity to be a function $t \mapsto \mathbf{u}(t)$ of locally bounded variation on a time-interval $I_t = [t_A, t_E]$ (Moreau, 1988). The velocity function $\mathbf{u}(t)$ and its differential measure $d\mathbf{u}$ for the 3D model have the same properties as the same function elaborated on for the 2D model in Section 3.5.1. The only difference is that the velocity function for the 2D model describes, of course, the velocity of the links in the 2D model, while the velocity function in this chapter describes the velocities in the 3D model.

The equality of measures (3.43) on page 54 is written for the 3D model of the snake robot as

$$\mathbf{M}d\mathbf{u} - \mathbf{h}(\mathbf{u})dt - d\mathbf{R} = \boldsymbol{\tau}dt, \quad (4.69)$$

where the mass matrix \mathbf{M} , the vector of smooth forces $\mathbf{h}(\mathbf{u})$, the force measure of possibly atomic impact impulsions $d\mathbf{R}$ and the vector of applied torques $\boldsymbol{\tau}$ will be described in the following. Note that the differential measure $d\mathbf{u}$ describes both the continuous change and possible discontinuities of the velocity \mathbf{u} (see Section 3.5.1).

For our choice of coordinates, the mass matrix is *diagonal* and *constant*

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & \mathbf{0}_{6 \times 6} & \cdots & \mathbf{0}_{6 \times 6} \\ \mathbf{0}_{6 \times 6} & \ddots & & \vdots \\ \vdots & & \ddots & \mathbf{0}_{6 \times 6} \\ \mathbf{0}_{6 \times 6} & \cdots & \mathbf{0}_{6 \times 6} & \mathbf{M}_n \end{bmatrix} \in \mathbb{R}^{6n \times 6n}, \quad (4.70)$$

where

$$\mathbf{M}_i = \begin{bmatrix} m_i \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & {}_{B_i} \Theta_{G_i} \end{bmatrix},$$

with ${}_{B_i} \Theta_{G_i} = \text{diag}([\Theta_{\psi_i} \ \Theta_{\psi_i} \ \Theta_{\rho_i}])$, m_i is the mass of link i , and Θ_{ψ_i} and Θ_{ρ_i} are its moments of inertia. The smooth forces, here consisting of gravity and gyroscopic accelerations, are described by

$$\mathbf{h}(\mathbf{u}) = \begin{bmatrix} \mathbf{h}_1(\mathbf{u}_1) \\ \vdots \\ \mathbf{h}_n(\mathbf{u}_n) \end{bmatrix} \in \mathbb{R}^{6n}, \quad (4.71)$$

where

$$\mathbf{h}_i(\mathbf{u}_i) = \begin{bmatrix} 0 \\ 0 \\ -m_i g \\ -({}_{B_i} \tilde{\boldsymbol{\omega}}_{IB_i} {}_{B_i} \Theta_{G_i} {}_{B_i} \boldsymbol{\omega}_{IB_i}) \end{bmatrix} \in \mathbb{R}^6. \quad (4.72)$$

The force measure $d\mathbf{R}$ accounts for all the contact forces and impulses. The contact efforts that constitute $d\mathbf{R}$ are found from the force-laws given in Section 4.4.2. Let \mathcal{I} be the set of all active contacts with the ground and \mathcal{H} be the set off all active contacts with the obstacles:

$$\mathcal{I}(t) = \{a | g_{N_a}(\mathbf{q}(t)) = 0\} \subseteq \{1, 2, \dots, 2n\}, \quad (4.73)$$

$$\mathcal{H}(t) = \{b | g_{H_b}(\mathbf{q}(t)) = 0\} \subseteq \{1, 2, \dots, n\nu\}, \quad (4.74)$$

where g_{N_a} is the a -th element of the vector \mathbf{g}_N in (4.9) and g_{H_b} is the b -th element of the vector \mathbf{g}_H in (4.16).

Now, the force measure is written as

$$\begin{aligned} d\mathbf{R} &= \mathbf{W}_J d\mathbf{P}_J + \sum_{a \in \mathcal{I}} (\mathbf{W}_N)_a dP_{N_a} \\ &+ \sum_{a \in \mathcal{I}} ((\mathbf{W}_T)_{2a-1} dP_{T_{ax}} + (\mathbf{W}_T)_{2a} dP_{T_{ay}}) \\ &+ \sum_{a \in \mathcal{I}} ((\mathbf{W}_V)_{2a-1} dP_{V_{ax}} + (\mathbf{W}_V)_{2a} dP_{V_{ay}}) \\ &+ \sum_{b \in \mathcal{H}} (\mathbf{W}_H)_b dP_{H_b}, \end{aligned} \quad (4.75)$$

where dP_{N_a} is the normal contact impulse measure between the ground and a link, $dP_{T_{ax}}$ and $dP_{T_{ay}}$ are the tangential contact impulse measures (friction) between the floor and a link, longitudinal and transversal to the link (i.e. along the e_x^{II} - and e_y^{II} -axis), respectively, $dP_{V_{ax}}$ and $dP_{V_{ay}}$ are the rotational contact impulse measures (friction) between the floor and a link, along the e_x^{I} -axis and e_y^{I} -axis, respectively, $d\mathbf{P}_J$ is the contact impulse measure due to the bilateral constraints in the joints (these constraints are always active), dP_{H_b} is the normal contact impulse measure between a link and an obstacle and the lower-case subscripts on the \mathbf{W} -matrices indicate which column of the matrix is used. The position of the elements of the vectors $d\mathbf{P}_N$, $d\mathbf{P}_J$, $d\mathbf{P}_T$, $d\mathbf{P}_V$ and $d\mathbf{P}_H$ corresponds to the position of the elements in their respective vector of relative velocity γ_N , γ_J , γ_T , γ_H and γ_V . Hence, by looking at for example the expression (4.41) for γ_T , we see that e.g.

$$d\mathbf{P}_{T_{n+1}} = \begin{bmatrix} dP_{T_{(n+1)x}} \\ dP_{T_{(n+1)y}} \end{bmatrix} \quad (4.76)$$

corresponds to $\gamma_{T_{R1}}$ and we know from this that $d\mathbf{P}_{T_{n+1}}$ is the tangential contact impulse measure between the ground and the rear part of link 1.

The contact impulse measures are, similarly to the velocity measure, decomposed into a Lebesgue-measurable force and a purely atomic impact impulse. This is written for dP_{N_a} as

$$dP_{N_a} = \lambda_{N_a} dt + \Lambda_{N_a} d\eta, \quad (4.77)$$

where λ_{N_a} is the Lebesgue-measurable force due to a continuous contact between a link and the ground surface and Λ_{N_a} is the purely atomic impact impulse caused by a collision between the two objects. The same decomposition can be performed for the other contact impulse measures.

The vector $\boldsymbol{\tau}$ of controlled torques is described in Section (4.4.3) below.

4.4.2 Constitutive Laws for Contact Forces

In this section we will give the set-valued force laws for normal contact with the ground surface and the obstacles, together with tangential Coulomb friction and rolling friction. These laws will all be formulated on velocity level using the relative contact velocities given by γ_N in (4.26), γ_T in (4.40), γ_V in (4.46) and γ_H in (4.58). Subsequently, the set-valued force laws are formulated as equalities in Section 4.4.2 using the so-called ‘proximal point function’ in order to include the force laws in the numerical simulation (Leine and Nijmeijer, 2004). Subscripts i , j , a and b will be omitted for brevity.

The set-valued force laws for the normal contact forces which arise from the contact with the ground surface and the obstacles are described in exactly the same manner as for the 2D model given in Section 3.5.2.

Normal Contact Force

The general form of the set-valued force law for the normal contact forces that arise from contact with the ground surface and the obstacles are exactly the same. In addition, this formulation is the same for both the 2D and the 3D model. To this end, we only present the force law for normal contact briefly here. Please see Section 3.5.2 for thorough explanation of the force law.

The Lebesgue measurable normal contact force (not an impulse) between the front or rear part of a link and the ground surface may be written for a closed contact $g_N = 0$ as an inclusion on velocity level

$$-\gamma_N \in N_{C_N}(\lambda_N), \quad (4.78)$$

where $\gamma_N \in \mathbb{R}$ is the relative velocity in (4.24), $\lambda_N \in \mathbb{R}$ is the corresponding (possible) normal force between a link and the ground surface, the convex set $C_N = \{\lambda_N \mid \lambda_N \geq 0\} = \mathbb{R}^+$ is the set of admissible contact forces and N_{C_N} is the normal cone to C_N (see Example 3.2 on page 57 and Example 3.3 for a description of the normal cone).

The Lebesgue measurable normal contact force between a link and an obstacle may be written for a closed contact $g_H = 0$ as

$$-\gamma_H \in N_{C_H}(\lambda_H), \quad (4.79)$$

where $\gamma_H \in \mathbb{R}$ is the relative velocity in (4.52) or (4.56), $\lambda_H \in \mathbb{R}$ is the corresponding (possible) normal force between a link and an obstacle, the convex set $C_H = \{\lambda_H \mid \lambda_H \geq 0\} = \mathbb{R}^+$ is the set of admissible contact forces and N_{C_H} is the normal cone to C_H .

The impact laws for a completely inelastic impact at a closed contact with the ground surface and an obstacle can now be written as

$$-\gamma_N^+ \in N_{C_N}(\Lambda_N), \quad C_N = \{\Lambda_N \mid \Lambda_N \geq 0\} = \mathbb{R}^+, \quad (4.80)$$

$$-\gamma_H^+ \in N_{C_H}(\Lambda_H), \quad C_H = \{\Lambda_H \mid \Lambda_H \geq 0\} = \mathbb{R}^+, \quad (4.81)$$

respectively, where γ_N^+ is the right limit of γ_N and γ_H^+ is the right limit of γ_H .

Coulomb Friction Force

In this section we describe the friction force between the snake robot and the ground, when the snake robot slides along the ground surface, as set-valued Coulomb friction. Let $\gamma_T \in \mathbb{R}^2$ be a tangential relative sliding velocity found in (4.39). Then the friction force $\lambda_T = [\lambda_{T_x} \ \lambda_{T_y}]^T \in \mathbb{R}^2$, in the two-dimensional tangent plane to the contact point, is modelled with a Coulomb friction law

$$-\gamma_T \in N_{C_T}(\lambda_T), \quad (4.82)$$

where N_{C_T} is the normal cone to the convex set C_T . See Example 3.4 on page 58 for an interpretation of the force law (4.82). The friction force is modelled both as isotropic and orthotropic Coulomb friction by changing the set C_T to either C_{T_1} or C_{T_2} as described below and the friction force law (4.82) is on the same form as the friction force law (3.60) for the two models. In addition, the convex sets C_{T_1} in (3.61) and C_{T_2} in (3.63) from the 2D model are again employed to describe isotropic and orthotropic friction, respectively. We reproduce the sets here for convenience:

$$C_{T_1} = \{\lambda_T \mid \|\lambda_T\| \leq \mu_T \lambda_N\}, \quad (4.83)$$

$$C_{T_2} = \left\{ \lambda_T \mid \frac{\lambda_{T_x}^2}{(\mu_{T_x} \lambda_N)^2} + \frac{\lambda_{T_y}^2}{(\mu_{T_y} \lambda_N)^2} \leq 1 \right\}, \quad (4.84)$$

where λ_N is the (varying) normal force, $\mu_T > 0$ is the friction coefficient for isotropic friction, and $\mu_{T_x} > 0$ and $\mu_{T_y} > 0$ are the directional friction coefficients along the e_x^Π - and e_y^Π -axis in (4.36), respectively. See Section 3.5.2 for a more thorough explanation of the model of the set-valued Coulomb friction and the convex sets C_{T_1} and C_{T_2} .

The force law for the impact impulse $\Lambda_T \in \mathbb{R}^2$ is found from

$$-\gamma_T^+ \in N_{C_T}(\Lambda_T), \quad (4.85)$$

where $\gamma_T^+ \in \mathbb{R}^2$ is the right limit of γ_T .

Rolling Friction

The 3D model of the snake robot has cylindrical links which means that it may roll sideways. The spatial Coulomb friction described in the previous section arises from translational motion. In addition, there is also a force that resists the rolling motion of the snake robot. We model this force as a

‘rolling friction’ $\lambda_V \in \mathbb{R}^2$ (additional subscripts omitted for simplicity) in this chapter by employing the same set-valued force law as for the tangential Coulomb friction force λ_T in Section 4.4.2. However, we consider the rolling friction to be isotropic and therefore we find the rolling friction as

$$-\gamma_V \in N_{C_V}(\lambda_V), \quad (4.86)$$

where

$$C_V = \{\lambda_V \mid \|\lambda_V\| \leq \mu_V \lambda_N\}, \quad (4.87)$$

and $\mu_V > 0$ is the friction coefficient.

The general form of the force law (4.86) is also valid for impact impulses Λ_V in the same way as for the tangential Coulomb friction described in Section 4.4.2. Subsequently, the impact impulse is found by exchanging γ_V with γ_V^+ and λ_V with Λ_V in (4.86)-(4.87).

Constitutive Laws as Projections

The proximal point function is used to transform the force laws into implicit equalities (see Section 3.5.2)

$$-\gamma_\kappa \in N_{C_\kappa}(\lambda_\kappa) \iff \lambda_\kappa = \text{prox}_{C_\kappa}(\lambda_\kappa - r_\kappa \gamma_\kappa), \quad (4.88)$$

where $r_\kappa > 0$ for $\kappa = N, H, T, V$.

4.4.3 Joint Actuators

Each cardan joint has 2 DOF that are controlled by two joint actuators. The actuators are modelled as controlled torques applied around the axes of rotation for the joint. Figure 4.6 illustrates how the direction of positive rotation is defined. Define for link i a positive control torque $\tau_{C_{vi}}$ to give a positive rotational velocity around $e_x^{B_{i+1}}$ and a positive control torque $\tau_{C_{hi}}$ to give a positive rotational velocity around $e_y^{B_i}$.

The total torque $\tau_{C_i} \in \mathbb{R}^3$ applied to link i is

$$\tau_{C_i} = \begin{bmatrix} 0 \\ \tau_{C_{hi}} \\ 0 \end{bmatrix} - \mathbf{R}_{B_{i-1}}^{B_i} \begin{bmatrix} 0 \\ \tau_{C_{h(i-1)}} \\ 0 \end{bmatrix} + \mathbf{R}_{B_{i+1}}^{B_i} \begin{bmatrix} \tau_{C_{vi}} \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} \tau_{C_{v(i-1)}} \\ 0 \\ 0 \end{bmatrix}, \quad (4.89)$$

for $i = 1, \dots, n$, where the relative rotation matrix is

$$\mathbf{R}_{B_{i+1}}^{B_i} = (\mathbf{R}_{B_i}^I)^T \mathbf{R}_{B_{i+1}}^I, \quad (4.90)$$

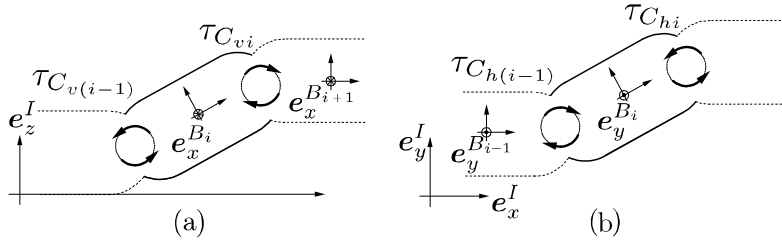


Figure 4.6: Control torques for (a) side-view and (b) top-view.

and $\tau_{C_{h0}} = \tau_{C_{v0}} = \tau_{C_{hn}} = \tau_{C_{vn}} = 0$. The total vector of applied torques $\boldsymbol{\tau} \in \mathbb{R}^{6n}$ employed in the equality of measures (4.69) is

$$\boldsymbol{\tau} = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \boldsymbol{\tau}_{C_1} \\ \mathbf{0}_{3 \times 1} \\ \boldsymbol{\tau}_{C_2} \\ \vdots \\ \mathbf{0}_{3 \times 1} \\ \boldsymbol{\tau}_{C_n} \end{bmatrix}. \quad (4.91)$$

4.5 Accessing and Control of Joint Angles

In this section we will define the joint angles and show how to control them.

The joint angles are not directly accessible from the non-minimal coordinates, but they can be calculated from the relative rotation matrices $\mathbf{R}_{B_{i+1}}^{B_i}$ in (4.90). Assume that $\mathbf{R}_{B_{i+1}}^{B_i}$ is constructed from the successive rotations (i.e. Euler angles with the zyx -convention) ϕ_{z_i} , ϕ_{h_i} and ϕ_{v_i} :

$$\mathbf{R}_{B_{i+1}}^{B_i} = \mathbf{R}_{\phi_{z_i}} \mathbf{R}_{\phi_{h_i}} \mathbf{R}_{\phi_{v_i}}. \quad (4.92)$$

Since we have cardan joints, let

$$\mathbf{R}_{\phi_{z_i}} = \begin{bmatrix} \cos \phi_{z_i} & -\sin \phi_{z_i} & 0 \\ \sin \phi_{z_i} & \cos \phi_{z_i} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.93)$$

be the rotation around the $e_z^{B_i}$ -axis and let

$$\mathbf{R}_{\phi_{h_i}} = \begin{bmatrix} \cos \phi_{h_i} & 0 & \sin \phi_{h_i} \\ 0 & 1 & 0 \\ -\sin \phi_{h_i} & 0 & \cos \phi_{h_i} \end{bmatrix}, \quad \mathbf{R}_{\phi_{v_i}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi_{v_i} & -\sin \phi_{v_i} \\ 0 & \sin \phi_{v_i} & \cos \phi_{v_i} \end{bmatrix}, \quad (4.94)$$

describe the rotation around $e_y^{B_i}$ and $e_x^{B_{i+1}}$, respectively. Hence, we should have that $\phi_{z_i} \equiv 0$ and if the snake robot is lying such that the $e_y^{B_i}$ -axis is pointing upwards, then ϕ_{h_i} describes the DOF of the cardan joint between link i and link $i + 1$ that moves link i and $i + 1$ from side to side (i.e. the yaw angle) and ϕ_{v_i} (the pitch angle) describes the lifting and lowering of the links ($\phi_{h_i} = \phi_{v_i} = 0$ when the snake robot is lying flat and straight on the ground). The relative rotation matrix (4.92) can now be written as

$$\mathbf{R}_{B_{i+1}}^{B_i} = \begin{bmatrix} * & * & * \\ * & * & * \\ -\sin \phi_{h_i} & \cos \phi_{h_i} \sin \phi_{v_i} & \cos \phi_{h_i} \cos \phi_{v_i} \end{bmatrix}. \quad (4.95)$$

where the elements of $\mathbf{R}_{B_{i+1}}^{B_i}$ which are not interesting for the imminent computations are marked by $*$.

We see from (4.95) that the joint rotation angles can be found as

$$\phi_{v_i} = \arctan \frac{\left(\mathbf{R}_{B_{i+1}}^{B_i}\right)_{32}}{\left(\mathbf{R}_{B_{i+1}}^{B_i}\right)_{33}} \quad (4.96)$$

$$\phi_{h_i} = -\arcsin \left(\mathbf{R}_{B_{i+1}}^{B_i}\right)_{31}, \quad (4.97)$$

for $i = 1, \dots, n - 1$ where $\left(\mathbf{R}_{B_{i+1}}^{B_i}\right)_{32}$ is the element of the matrix $\mathbf{R}_{B_{i+1}}^{B_i}$ in row 3 column 2 and the same notation applies for $\left(\mathbf{R}_{B_{i+1}}^{B_i}\right)_{31}$ and $\left(\mathbf{R}_{B_{i+1}}^{B_i}\right)_{33}$. By inspecting (4.95), we note that the expressions (4.96)-(4.97) are only valid for $-\frac{\pi}{2} < \phi_{v_i} < \frac{\pi}{2}$ and $-\frac{\pi}{2} < \phi_{h_i} < \frac{\pi}{2}$. However, most snake robot motion patterns are possible without any larger feasible movement of the joints and only these motion patterns are considered for simulation and experiments in this thesis.

The rotational velocities of the joints are found directly from the rotational velocities ${}_{B_i}\boldsymbol{\omega}_{IB_i}$ of the links. We define the rotational velocity for sideways and lifting motion as

$$\omega_{h_i} = \mathbf{d}_2^T \boldsymbol{\omega}_{J_i}, \quad (4.98)$$

$$\omega_{v_i} = \mathbf{d}_1^T \boldsymbol{\omega}_{J_i}, \quad (4.99)$$

respectively, for the joint between link i and $i + 1$, where $\mathbf{d}_1^T = [1 \ 0 \ 0]$, $\mathbf{d}_2^T = [0 \ 1 \ 0]$ and

$$\boldsymbol{\omega}_{J_i} = {}_{B_{i+1}}\boldsymbol{\omega}_{IB_{i+1}} - \left(\mathbf{R}_{B_{i+1}}^{B_i}\right)^T {}_{B_i}\boldsymbol{\omega}_{IB_i}. \quad (4.100)$$

Let the desired values of ϕ_{h_i} and ϕ_{v_i} be $\phi_{h_i,d}$ and $\phi_{v_i,d}$, respectively. In addition, the reference values for the joint velocities ω_{h_i} , ω_{v_i} are given by $\dot{\phi}_{h_i,d}$ and $\dot{\phi}_{v_i,d}$, respectively. Then the torques applied to the joints are found by PD-controllers as

$$\tau_{C_{h_i}} = -k_P (\phi_{h_i,d} - \phi_{h_i}) - k_D (\dot{\phi}_{h_i,d} - \omega_{h_i}), \quad (4.101)$$

$$\tau_{C_{v_i}} = -k_P (\phi_{v_i,d} - \phi_{v_i}) - k_D (\dot{\phi}_{v_i,d} - \omega_{v_i}), \quad (4.102)$$

for $i = 1, \dots, n - 1$ where $k_P \in \mathbb{R}^+$, $k_D \in \mathbb{R}^+$ are positive constants equal for all i .

4.6 Numerical Algorithm: Time-stepping

In this section we will present the time-stepping method as the numerical solution of the equality of measures for the 3D model of the snake robot. The general form of the algorithm is similar to that presented for the 2D model in Chapter 3. However, the non-minimal coordinates are found with a different function and more contact impulse measures are added, both since we are now dealing with a 3D model instead of a planar one.

4.6.1 Time Discretization

The time discretization of the equality of measures for the 2D model is shown in Section 3.6 and the general approach is the same for the 3D model. Therefore, for the 3D model, we will only state a concise algorithm that constitutes the time-stepping method for the 3D model. An elaboration on this procedure is found in Section 3.6. In the following, we will describe the time-stepping algorithm for the 3D model.

The mid-point \mathbf{q}_M in the time-stepping method for the 3D model is found from

$$\mathbf{q}_M = \mathbf{q}_A + \frac{\Delta t}{2} \mathbf{F}(\mathbf{q}_A) \mathbf{u}_A \quad (4.103)$$

where we have used the equality $\dot{\mathbf{p}} = \frac{1}{2} \bar{\mathbf{H}}^T \mathbf{B}_i \boldsymbol{\omega}_{IB_i}$ (Egeland and Gravdahl, 2002), and \mathbf{q}_A and \mathbf{u}_A are the coordinates and velocities of all links at the

beginning of a time-step. Moreover,

$$\mathbf{F}(\mathbf{q}_A) = \begin{bmatrix} \mathbf{F}_{H_1} & \mathbf{0}_{7 \times 6} & \cdots & \mathbf{0}_{7 \times 6} \\ \mathbf{0}_{7 \times 6} & \mathbf{F}_{H_2} & & \vdots \\ \vdots & & \ddots & \mathbf{0}_{7 \times 6} \\ \mathbf{0}_{7 \times 6} & \cdots & \mathbf{0}_{7 \times 6} & \mathbf{F}_{H_n} \end{bmatrix} \quad (4.104)$$

$$\mathbf{F}_{H_i} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{4 \times 3} & \frac{1}{2} \bar{\mathbf{H}}_i^T \end{bmatrix} \in \mathbb{R}^{7 \times 6}, \quad (4.105)$$

where $\bar{\mathbf{H}}_i$ is found from (4.4) by inserting the orientation of link i at t_A found in \mathbf{q}_A .

The time-stepping method needs to be initialized before it can run. This is done as follows: Set $l = 1$. Define \mathbf{q}_l as the initial position and orientation of all links admissible with respect to the bilateral constraints, the unilateral constraints and the unit-norm constraint $\|\mathbf{p}_i\| = 1$, $i = 1, \dots, n$, where n is the number of links. Define \mathbf{u}_l as the initial translational and rotational velocity of all links. Define r_N , r_T , r_V , $r_H > 0$. Find the mass matrix \mathbf{M} from (4.70). Initialize all contact impulsions $\mathbf{P}_{N,l} = \mathbf{0}_{2n}$, $\mathbf{P}_{T,l} = \mathbf{0}_{4n}$, $\mathbf{P}_{V,l} = \mathbf{0}_{4n}$ and $\mathbf{P}_{H,l} = \mathbf{0}_{n\nu}$ where ν is the number of obstacles. Define the start time t_l and a small step size Δt . Now the time-stepping method for the 3D model given by the following algorithm, can be iterated on.

Algorithm 4.1 (Time-stepping Method for the 3D Model)

1. *Initialize time-step:* $t_A = t_l$, $\mathbf{q}_A = \mathbf{q}_l$, $\mathbf{u}_A = \mathbf{u}_l$.
2. *Find mid-point:* $t_M = t_A + \frac{1}{2}\Delta t$, $\mathbf{q}_M = \mathbf{q}_A + \frac{\Delta t}{2}\mathbf{F}(\mathbf{q}_A)\mathbf{u}_A$.
3. *System vectors:* $\mathbf{h}_M = \mathbf{h}(\mathbf{u}_A)$, $\boldsymbol{\tau}_M = \boldsymbol{\tau}(\mathbf{q}_M, \mathbf{q}_A)$ (see Remark 4.1).
4. *Find active contact points:*

$$\mathcal{I}_{M,l} = \{a | g_{N_a}(\mathbf{q}_M) \leq 0\}, \quad \mathcal{H}_{M,l} = \{b | g_{H_b}(\mathbf{q}_M) \leq 0\}. \quad (4.106)$$

5. *Calculate generalized force directions:*

$$\begin{aligned} \mathbf{W}_{NM} &= \mathbf{W}_N(\mathbf{q}_M), \\ \mathbf{W}_{TM} &= \mathbf{W}_T(\mathbf{q}_M), \\ \mathbf{W}_{VM} &= \mathbf{W}_V(\mathbf{q}_M), \\ \mathbf{W}_{HM} &= \mathbf{W}_H(\mathbf{q}_M). \end{aligned} \quad (4.107)$$

6. Initialize and run Algorithm 4.2 with $k = 1$, $\mathbf{P}_N^k = \mathbf{P}_{N,l}$, $\mathbf{P}_T^k = \mathbf{P}_{T,l}$, $\mathbf{P}_V^k = \mathbf{P}_{V,l}$ and $\mathbf{P}_H^k = \mathbf{P}_{H,l}$. The algorithm returns \mathbf{u}_E together with all the contact forces \mathbf{P}_{NE} , \mathbf{P}_{TE} , \mathbf{P}_{VE} and \mathbf{P}_{HE} .

7. Find positions and orientations from \mathbf{u}_E and \mathbf{q}_M as

$$\mathbf{q}_{E,MNA} = \mathbf{q}_M + \frac{\Delta t}{2} \mathbf{F}(\mathbf{q}_M) \mathbf{u}_E. \quad (4.108)$$

8. Initialize Algorithm 4.3 with $\mathbf{q}_{E,MNA}$ and use the algorithm to find \mathbf{q}_E which is admissible with respect to the bilateral constraints and the unit-norm $\|\mathbf{p}_i\| = 1$ for $i = 1, \dots, n$.

9. Prepare for next iteration: $t_E = t_M + \frac{1}{2} \Delta t$ and

$$\begin{aligned} t_{l+1} &= t_E, & \mathbf{q}_{l+1} &= \mathbf{q}_E, & \mathbf{u}_{l+1} &= \mathbf{u}_E, \\ \mathbf{P}_{N,l+1} &= \mathbf{P}_{NE}, & \mathbf{P}_{T,l+1} &= \mathbf{P}_{TE}, \\ \mathbf{P}_{V,l+1} &= \mathbf{P}_{VE}, & \mathbf{P}_{H,l+1} &= \mathbf{P}_{HE}. \end{aligned} \quad (4.109)$$

Remark 4.1 (Algorithm 4.1 step 3) The torques $\boldsymbol{\tau}_M = \boldsymbol{\tau}(\mathbf{q}_M, \mathbf{q}_A)$ applied to the links are found from both \mathbf{q}_M and \mathbf{q}_A . This is because \mathbf{q}_M is employed to determine how the control torques applied to the joints affect each link (similarly to that \mathbf{W}_{NM} defines how the normal contact forces with the ground affect the links). However, \mathbf{q}_M is only an internal state in the numerical algorithm and is therefore not available to the controllers (4.101)-(4.102) of an actual snake robot. Hence, to determine what torques to apply in order to let each joint follow its desired trajectory, we need to employ \mathbf{q}_A to find the relative joint angles and the relative rotational velocities employed in the controllers.

4.6.2 Solving for Contact Impulsions

The Modified Newton Algorithm employed in Step 6 in Algorithm 4.1 will be presented for the 3D model in this section. The Modified Newton Algorithm is utilized to find the velocities \mathbf{u}_E at the end of the time-step l . In addition, the algorithm is employed to find the contact impulses $P_{N_a} \in \mathbb{R}$, $\mathbf{P}_{T_a} \in \mathbb{R}^2$, $\mathbf{P}_{V_a} \in \mathbb{R}^2$ and $\mathbf{P}_J \in \mathbb{R}^{4(n-1)}$ for $a \in \mathcal{I}_{M,l}$ and $P_{H_b} \in \mathbb{R}$ for $b \in \mathcal{H}_{M,l}$ (the subscripts on the contact impulses refer to the element(s) in their respective vector \mathbf{P}_N , \mathbf{P}_T , \mathbf{P}_V , or \mathbf{P}_H the same way as described for (4.76) on page 85). The contact impulses consist of forces λ acting *during* I_t and possible impulses Λ acting *in* the time-interval I_t .

The Modified Newton Algorithm is initialized as presented in Step 6 in Algorithm 4.1.

Algorithm 4.2 (Modified Newton Algorithm for 3D Model)

1. *Solve*

$$\mathbf{P}_J^k = (\mathbf{W}_{JM}^T \mathbf{M}^{-1} \mathbf{W}_{JM})^{-1} \cdot \left[\mathbf{W}_{JM}^T \mathbf{u}_A - \mathbf{W}_{JM}^T \mathbf{M}^{-1} (\mathbf{h}_M \Delta t + \mathbf{S}^k + \tau \Delta t) \right], \quad (4.110)$$

where

$$\begin{aligned} \mathbf{S}^k &= \sum_{a \in \mathcal{I}_{M,l}} (\mathbf{W}_{NM})_a P_{N_a}^k \\ &+ \sum_{a \in \mathcal{I}_{M,l}} \begin{bmatrix} (\mathbf{W}_{TM})_{2a-1} & (\mathbf{W}_{TM})_{2a} \end{bmatrix} \mathbf{P}_{T_a}^k \\ &+ \sum_{a \in \mathcal{I}_{M,l}} \begin{bmatrix} (\mathbf{W}_{VM})_{2a-1} & (\mathbf{W}_{VM})_{2a} \end{bmatrix} \mathbf{P}_{V_a}^k \\ &+ \sum_{b \in \mathcal{H}_{M,l}} (\mathbf{W}_{HM})_b P_{H_b}^k. \end{aligned} \quad (4.111)$$

2. *Solve* \mathbf{u}_E^{k+1} *from*

$$\mathbf{M} (\mathbf{u}_E^{k+1} - \mathbf{u}_A) - \mathbf{h}_M \Delta t - \mathbf{S}^k - \mathbf{W}_{JM} \mathbf{P}_J^k = \tau \Delta t. \quad (4.112)$$

3. *Solve for* $a \in \mathcal{I}_{M,l}$ *and* $b \in \mathcal{H}_{M,l}$

$$P_{N_a}^{k+1} = \text{prox}_{C_N} \left(P_{N_a}^k - r_N \gamma_{NE_a}^{k+1} \right), \quad (4.113)$$

$$\mathbf{P}_{T_a}^{k+1} = \text{prox}_{C_T} \left(\mathbf{P}_{T_a}^k - r_T \gamma_{TE_a}^{k+1} \right), \quad (4.114)$$

$$\mathbf{P}_{V_a}^{k+1} = \text{prox}_{C_V} \left(\mathbf{P}_{V_a}^k - r_V \gamma_{VE_a}^{k+1} \right), \quad (4.115)$$

$$P_{H_a}^{k+1} = \text{prox}_{C_H} \left(P_{H_a}^k - r_N \gamma_{HE_a}^{k+1} \right), \quad (4.116)$$

where

$$\gamma_{NE}^{k+1} = \mathbf{W}_{NM}^T \mathbf{u}_E^{k+1}, \quad (4.117)$$

$$\gamma_{TE}^{k+1} = \mathbf{W}_{TM}^T \mathbf{u}_E^{k+1}, \quad (4.118)$$

$$\gamma_{VE}^{k+1} = \mathbf{W}_{VM}^T \mathbf{u}_E^{k+1}, \quad (4.119)$$

$$\gamma_{HE}^{k+1} = \mathbf{W}_{HM}^T \mathbf{u}_E^{k+1}. \quad (4.120)$$

4. If

$$\begin{aligned} & \|\mathbf{P}_N^{k+1} - \mathbf{P}_N^k\| + \|\mathbf{P}_T^{k+1} - \mathbf{P}_T^k\| \\ & + \|\mathbf{P}_V^{k+1} - \mathbf{P}_V^k\| + \|\mathbf{P}_H^{k+1} - \mathbf{P}_H^k\| < \epsilon \end{aligned} \quad (4.121)$$

where $\epsilon > 0$ is a user-defined tolerance, then stop the algorithm and return

$$\begin{aligned} \mathbf{P}_{NE} &= \mathbf{P}_N^{k+1}, & \mathbf{P}_{TE} &= \mathbf{P}_T^{k+1}, \\ \mathbf{P}_{VE} &= \mathbf{P}_V^{k+1}, & \mathbf{P}_{HE} &= \mathbf{P}_H^{k+1}. \end{aligned} \quad (4.122)$$

If the condition (4.121) is not satisfied, then start over from Step 1.

Remark 4.2 (Implementation of the time-stepping method) *It is not necessary to calculate the complete \mathbf{W} -matrices in Algorithm 4.1, step 5 for implementation of the time-stepping method in, for example, Matlab. Only the columns of the matrices that correspond to the active contact points need to be calculated for each time-step. The set of active contact points also affects the number of contact impulses that need to be considered. Hence, a more effective way of writing Algorithm 4.1 and Algorithm 4.2 is to only calculate the contact impulses, \mathbf{W} -matrices and relative velocities associated with the active contact points defined by the sets $\mathcal{I}_{M,l}$ and $\mathcal{H}_{M,l}$. Moreover, Matlab offers a convenient programming environment for implementing such an approach. However, we have chosen to present the algorithms as they are to ease the understanding of the algorithms and to avoid introducing additional notation and subscripts.*

Most often, a higher value of the parameters r_N , r_T , r_V , r_H yields a faster convergence rate at the risk of divergence. However, a general convergence result for the Modified Newton Algorithm does not exist. See Remark 3.1 on page 64 for a comment on the implementation of the Modified Newton Algorithm.

The constitutive laws (4.113)-(4.120) used to describe the contact impulses are given on *velocity level*. This means that both the bilateral constraints on position level and the unit norm constraint on the Euler parameters are in general not satisfied. A solution to these problems is suggested in the following.

4.6.3 Constraint Violation

We initialize the link projection algorithm given by Algorithm 4.3 below as described in Step 8 in Algorithm 4.1 with $\mathbf{q}_{E,MNA}$ found in (4.108). The link projection algorithm returns a vector \mathbf{q}_E of positions and orientations

of the links which are admissible both to the bilateral constraints introduced by the joints and the unit-norm constraint on the quaternions.

Algorithm 4.3 (Projection of the Links)

1. Satisfy the unit-norm constraint $\|\mathbf{p}_i\| = 1$ by setting

$$\mathbf{p}_{E_i} = \frac{\mathbf{p}_{E,MNA_i}}{\|\mathbf{p}_{E,MNA_i}\|}, \quad i = 1, \dots, n, \quad (4.123)$$

where \mathbf{p}_{E,MNA_i} is found from the vector $\mathbf{q}_{E,MNA}$ and is the (non-unit-norm) quaternion. Let $\mathbf{q}_{E,quat}$ be the new vector with positions and the update orientations found with (4.123).

2. Employ $\mathbf{q}_{E,quat}$ and let the orientation of link 1 be fixed. In addition, let all the $\mathbf{e}_z^{B_i}$ -axes be kept fixed. Now, change the $\mathbf{e}_x^{B_i}$ - and $\mathbf{e}_y^{B_i}$ -axis such that $g_{J_{i\phi}} = 0$ is satisfied for $i = 1, \dots, n-1$, where the gap function $g_{J_{i\phi}}$ is found in (4.20). This is achieved as follows for $i = 1, \dots, n-1$: The current body axes can be found from the corresponding rotation matrix as

$$\mathbf{R}_{B_i}^I(\mathbf{q}_{E,quat}) = [I\mathbf{e}_x^{B_i} \quad I\mathbf{e}_y^{B_i} \quad I\mathbf{e}_z^{B_i}]. \quad (4.124)$$

Denote the new $\mathbf{e}_x^{B_{i+1}}$ - and $\mathbf{e}_y^{B_{i+1}}$ -axis $\hat{\mathbf{e}}_x^{B_{i+1}}$ and $\hat{\mathbf{e}}_y^{B_{i+1}}$, respectively. Then we have that

$$\hat{\mathbf{e}}_x^{B_{i+1}} = \frac{\hat{\mathbf{e}}_y^{B_i} \times \mathbf{e}_z^{B_{i+1}}}{\|\hat{\mathbf{e}}_y^{B_i} \times \mathbf{e}_z^{B_{i+1}}\|}, \quad (4.125)$$

$$\hat{\mathbf{e}}_y^{B_{i+1}} = \frac{\mathbf{e}_z^{B_{i+1}} \times \hat{\mathbf{e}}_x^{B_{i+1}}}{\|\mathbf{e}_z^{B_{i+1}} \times \hat{\mathbf{e}}_x^{B_{i+1}}\|}. \quad (4.126)$$

Now, the new rotation matrix for link $i+1$ can be constructed as

$$\mathbf{R}_{B_{i+1}}^I = [I\hat{\mathbf{e}}_x^{B_{i+1}} \quad I\hat{\mathbf{e}}_y^{B_{i+1}} \quad I\mathbf{e}_z^{B_{i+1}}], \quad (4.127)$$

and the final quaternions describing the orientation of the link are found from the new rotation matrix. Let $\mathbf{q}_{E,rot}$ be the new coordinate vector with the updated quaternions.

3. Employ $\mathbf{q}_{E,rot}$ and let the position of link 1 be fixed. Change the positions of links $i = 2, \dots, n$ such that $g_{J_{i\chi}} = 0$ is satisfied for $i = 1, \dots, n-1$, $\chi = x, y, z$ where the gap function $g_{J_{i\chi}}$ is given by (4.18). Let \mathbf{q}_E be the new coordinate vector with the updated positions. Now, return \mathbf{q}_E which satisfies the bilateral constraints and the unit-norm constraint.

4.7 Summary

We have extended the non-smooth (hybrid) 2D model presented in Chapter 3 to 3D motion. The model enables synthesis and testing of 3D snake robot motion patterns. It is based on the framework of non-smooth dynamics and convex analysis which allows us to easily and systematically incorporate unilateral contact forces (i.e. between the snake robot and the ground surface and the obstacles) and friction forces based on Coulomb's law of dry friction. In addition, the particular choice of non-minimal coordinates results in an well arranged structure of the elements in the equality of measures which describes the motion of the snake robot. Conventional numerical solvers can not be employed directly for numerical treatment of the non-smooth model due to the set-valued force laws and possible instantaneous velocity changes. Therefore, we show how to implement the model for numerical treatment with the numerical integrator called the time-stepping method. This method helps to avoid explicit changes between equations during simulation even though the system is hybrid.

Chapter 5

Obstacle Aided Locomotion

5.1 Introduction

This chapter treats various aspects of obstacle aided locomotion with snake robots in order to provide a better understanding of the simulation and experimental results presented later in this thesis. We define obstacle aided locomotion as snake robot locomotion where the snake robot utilizes walls or other external objects, apart from the flat ground, for propulsion.

This chapter is based on Transeth *et al.* (2007b) and Transeth *et al.* (2008b) and it is organized as follows. We discuss the importance of obstacle aided locomotion in Section 5.2. In Section 5.3 a study by Gray (1946) is employed to understand how snakes move forward by pushing against irregularities in the ground. Requirements for obstacle aided locomotion for a snake robot are suggested in Section 5.4, while an approach to experimental validation of obstacle aided locomotion is presented in Section 5.5.

5.2 Motivation

Biological snakes exploit objects and irregularities in their environment to achieve more efficient propulsion. A comparison with legged creatures provides an indication of this statement. While the gait displayed by a legged creature is highly dependent on the current speed of this creature, the gait displayed by a snake will mainly be dependent on the properties of the environment surrounding the snake. The flexibility and robust mobility of snakes in cluttered environments is a significant motivation for research on snake robots. However, in order to fully benefit from the advantages of snake locomotion in such environments, as illustrated in Figure 5.1, snake

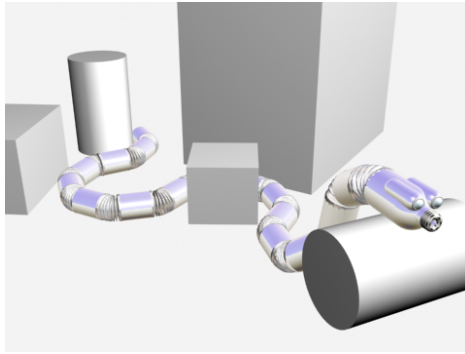


Figure 5.1: A snake robot in a cluttered environment (motivating illustration).

robots must learn to sense and understand the environment surrounding them at any given time.

In this thesis the word obstacle is used 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 to emphasize one of the fundamental differences between snake locomotion and other traditional means of mobility, such as wheeled, tracked and legged robots. 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 therefore not *obstacle avoidance*, but rather *obstacle utilization*.

5.3 Understanding Obstacle Aided Locomotion

The gait called lateral undulation represents a common form of snake locomotion (Mattison, 2002). During this motion a series of sinus-like curves are propagated backwards through the snake body. These waves interact with irregularities in the environment of the snake in such a way that the resulting forces propel the snake forward. Progression by lateral undulation is highly dependent on interaction forces from the environment. The study of this motion pattern therefore provides a good understanding of how a

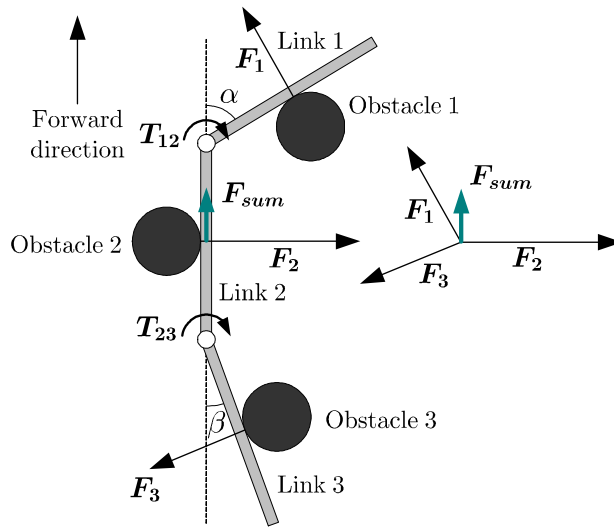


Figure 5.2: A snake robot with three links interconnected by two joints. The resultant of the contact forces from the obstacles propels the snake. The propelling force points in the forward direction if $\alpha > \beta$, in the backward direction if $\alpha < \beta$, and is zero if $\alpha = \beta$.

snake robot may use its environment in order to move forward.

A study of lateral undulation and its dependency on external push-points was given by Gray (1946). Gray depicted a scenario similar to the one illustrated in Figure 5.2. It shows an articulated mechanism with three links interconnected by two joints. Torques are applied to the two joints so that each link is pushed towards a perfectly smooth obstacle, giving rise to contact forces acting in the lateral direction of each link. Gray showed that this mechanism will glide in the forward direction if the angle α (between link 1 and 2) is greater than the angle β (between link 2 and 3) and in the backward direction if α is smaller than β . This may be understood by studying the sum of the contact forces to the right in the figure. It is assumed that the resultant of the contact forces \mathbf{F}_{sum} is large enough to overcome the friction forces from the ground.

A snake robot is constructed by serially connecting multiple mechanisms similar to the one in Figure 5.2. The structure of such a snake robot is depicted in Figure 5.3 (a), where the snake robot posture and obstacles in Figure 5.2 have been duplicated. Three push-points are needed for locomotion (Mattison, 2002) and we see from the Figure 5.2. that Obstacle

1 is associated with the normal reaction force that has an advantageous contribution in the (upwards) direction of propulsion. This suggests that Obstacle 1 should also be utilized for locomotion of the interconnected structure in Figure 5.3 (a). Hence, a natural choice of push-points for obstacle aided locomotion by lateral undulation are the three occurrences of Obstacle 1 depicted in Figure 5.3 (b). Moreover, we see from the figure that the push-points are close to the centre line of the snake robot posture. This corresponds with the utilization of walls by snakes moving by lateral undulation through a winding track depicted in Gray (1946). The sequence of pictures of such a scenario in Gray (1946) clearly suggest that the snake pushes against the walls of the track approximately along the centre line of its undulatory motion. Hence, the angle between the tangent to the snake body at the contact point and the direction of locomotion is large (see the definition of α in Figure 5.2 for an example of such an angle). This results in an efficient utilization of the push-points which is explained in the following. The reaction force from an obstacle can be decomposed into two parts: a longitudinal and a lateral part with respect to the direction of locomotion. If the push-points are far from the centre line of the snake robot posture (for example if the two bottom obstacles in Figure 5.3 (b) were located next to link 3 and link 6, respectively, instead), then the angles between the links pushing against the obstacles and the centre line are decreased, and a substantial portion of the normal contact reaction force points in the lateral direction of locomotion and does not contribute to moving the snake robot forward. In addition, larger joint torques are needed to obtain the same longitudinal reaction forces compared to the situation where the obstacles are located closer to the centre line. Hence, moving the contact point to the centre line of the snake robot posture optimizes the longitudinal contribution of the normal contact reaction forces and reduces the necessary joint torques.

5.4 Requirements for Intelligent Obstacle Aided Locomotion

We suggest that three main physical properties in a snake robot are required for efficient obstacle aided locomotion in a cluttered and irregular environment. These are a *smooth exterior body surface*, a *contact force sensing system* and an *obstacle locating system*. The first property will give the robot the ability to glide forward along any irregularities in its environment. Hence, the friction forces between the snake robot and any projection from

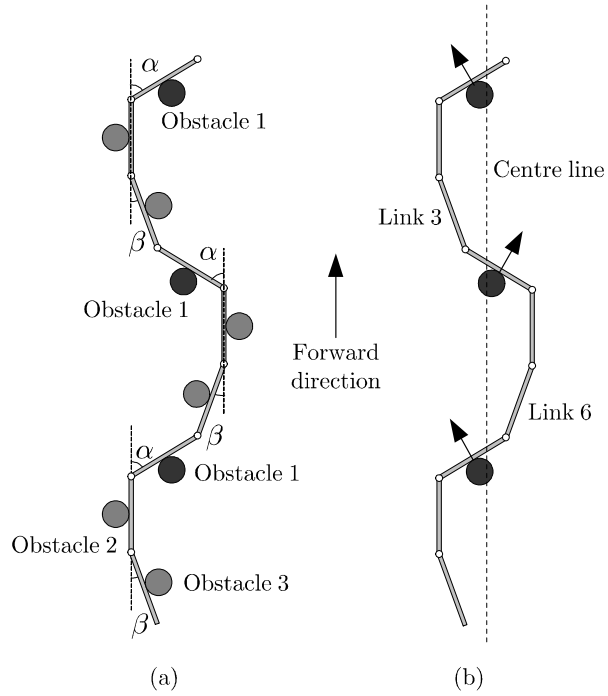


Figure 5.3: Snake robot (a) with all obstacles from Figure 5.2, and (b) with only the necessary and most suited obstacles for locomotion.

the ground used as a push-point should be as low as possible. Moreover, it will mimic the skin of a biological snake more closely. Any irregularities on the body surface of the snake robot will potentially obstruct the gliding motion of the robot. The second property is crucial for intelligent obstacle utilization. Contact force sensing allows the snake robot to detect when the body is in contact with a push-point and also control the force exerted on a push-point. Since the sum of contact forces along the snake body is what propels the robot forward, the ability to measure these forces is important in order to be able to control the propulsion. The third property enables the snake robot to search for new push-points which it is not currently able to touch (i.e. sense with the force sensors).

A control strategy for obstacle aided locomotion should in general consist of two continuous tasks running in parallel. The function of the first task should be high-level motion planning. This task should establish some sort of map of the environment surrounding the snake robot and use it to plan a path through this environment. In contrast to conventional path planning strategies for mobile robots, the goal should not be to avoid obstacles, but rather to move relative to these irregularities in a way that enables the snake robot to utilize push-points for more efficient propulsion.

The actual obstacle utilization should be continuously carried out in the second task as a function of the current force measurements along the body and the desired direction of motion. This task should basically control the joints of the snake robot so that the sum of the contact forces along the snake body is pointing toward the desired direction of motion.

One of the purposes of this thesis is to present a mathematical framework for the study of obstacle aided locomotion. This thesis will therefore not treat the specific implementation of the motion controller discussed above. However, we note that all the three above main physical properties are in effect available in the mathematical model presented in this thesis: The exterior of the snake robot model is smooth, we are able to calculate the forces between the snake robot and the obstacles and we have an obstacle locating system since the position of all obstacles are known in advance. Hence, the model can be employed to test intelligent obstacle aided locomotion when such future control schemes are developed.

5.5 Experimental Observation of Obstacle Aided Locomotion

As described earlier in this chapter, the study of lateral undulation provides a good understanding of how a snake robot may utilize interaction forces from its environment in order to progress forward. The ‘serpenoid curve’ is a sinus-like curve that describes the shape of a snake during undulatory motion (Hirose, 1993). Inspired by the serpenoid curve, we can imitate the undulatory motion of a snake by setting the snake robot joint angles according to

$$\phi_{h_i,d} = A_h \sin(\omega_h t + (i - 1) \delta_h), \quad (5.1)$$

where $\phi_{h_i,d}$ is the desired relative angle between link i and $i + 1$, A_h is the amplitude of joint oscillation, ω_h determines the angular frequency of oscillation of the joints, t is the time and δ_h is the phase-shift between adjacent joints (Saito *et al.*, 2002).

Lateral interaction forces from the environment are needed for this gait to progress the snake forward. The sum of these forces must act forward for the snake robot to move in this direction. A simple way to study obstacle aided locomotion is therefore to curve the snake robot to the initial shape of this gait (according to (5.1) for $t = t_0$) and place obstacles along the snake robot similar to the position of the obstacles in Figure 5.3 (b). When the snake robot starts the gait (5.1) from its initial (curved) shape, it will progress forward by lateral undulation due to the reaction forces from the obstacles. Note from (5.1) and Figure 5.3 (b) that the chosen amplitude A_h and phase-shift δ_h depends on the design of the snake robot since it needs three push-points to move forward. Hence, we can choose from a bigger variety of A_h and δ_h for a snake robot with many links since it might be able to utilize the obstacles for many different sinus-like shapes. However, for a snake robot with fewer links, the choices of A_h and δ_h are more predetermined. This is because a small δ_h will not result in a sinus-like shape, but rather a C-like shape. The snake robot will not be able to utilize the three obstacles in a C-like shape based on the elaboration above.

The approximate distance between two adjacent obstacles are determined such that the snake robot is always in contact with three obstacles simultaneously. In addition, it is assumed that the contact point with the obstacles should be close to the centre-line of the serpenoid curve based on the illustrations in Gray (1946). Note that the obstacles need to be placed according to the shape the snake robot obtains by controlling its joints according to (5.1). We will see from the simulation results in Section 7.3.4

that a slight variation of the position of one obstacle does not seriously affect ability of the snake robot to move forward. However, if an obstacle is moved far away from its original position, then the snake robot is not able to complete its track while following the preprogrammed motion pattern given by (5.1). The snake robot needs to be equipped with sensors to detect contact with the obstacles for a scenario with randomly placed obstacles and this is beyond the scope of this thesis.

A real rescue operation in a shattered building will require a snake robot with a smooth and sealed exterior. This avoids that small pieces of rubble get stuck in the joints of the snake robot and hinders it to move. However, the principle motion pattern for how to move in such an environment can be studied in a laboratory setting without having developed a robust snake robot design first. To this end, we simply choose the obstacles large enough so that they do not get stuck in the snake robot joints. This is advantageous since it reduces the development time and cost of the snake robot used in the experiments. Moreover, the basics of obstacle aided locomotion are the same for small and large obstacles since the fundamental goal is still to allocate enough push-points such that the snake robot is able to move forward.

Chapter 6

Modelling and Control of a Planar Snake Robot

6.1 Introduction

Many authors base the choice and implementation of the most common serpentine motion pattern called lateral undulation on the serpenoid curve found in Hirose (1993). This is a curve that describes the motion of a biological snake while moving by lateral undulation, and a snake robot can follow an approximation to this curve by setting its joint angles according to a sine-curve that is phase-shifted between adjacent joints. This approach to snake robot locomotion has been widely implemented for snake robots that have either wheels (Ostrowski and Burdick, 1996; Prautsch and Mita, 1999) or a friction property such that each link of the snake robot glides easier forward and backward compared to transversal motion (Ma, 2001; Saito *et al.*, 2002). A no-slip constraint (i.e. a non-holonomic velocity constraint) on each wheel is sometimes introduced in the mathematical model, thus avoiding that a link slips sideways. Such an approach is presented in Prautsch and Mita (1999), where the no-slip constraint allows one to significantly reduce the model. A Lyapunov-based proof for this reduced model together with a proposed controller shows that the snake robot is able to move to a position reference. A controller for the velocity and heading for a wheel-less snake robot with the friction property described above, is presented in Saito *et al.* (2002). The simulation and experimental results presented in that paper indicate that the snake robot is able to stay within a reasonable offset of the desired speed. However, no formal stability analysis of the closed-loop system is given. For wheel-less snake robots it is not al-

ways sufficient to consider a purely kinematic model for the motion pattern lateral undulation, as the friction between the snake robot and the ground surface may be essential for locomotion. Therefore, the friction needs to be considered for wheel-less snake robots and this motivates including the dynamics into controller synthesis and stability analysis of the snake robot.

The above published results all consider control of the position or orientation of a selection of snake robot links or some mean value of the velocity and orientation of all the links of a snake robot. Hence, the main concern is where some point on the snake robot is in space or where it is heading with what velocity. However, for delicate operations such as inspection and maintenance in industrial plants, every point on the snake robot body need to be accurately controlled. As a first step toward designing such a control scheme, we present in this chapter an input-output linearizing controller for tracking control of snake robot joints. Such model-based control of the *internal configuration* of the snake robot has, to the best of our knowledge, not been investigated before, and it results in a challenging stability analysis due to the large number of degrees of freedom that needs to be considered. It is hoped that the results presented in this chapter can be combined with the controllers presented for control of the position and/or orientation of the snake robot to be able to accurately control both the global position of the snake robot in the plane together with precise control of the internal configuration of the snake robot. This way, every point on the snake robot body relative to some fixed point in space can be accurately controlled.

In the previous chapters on modelling of a snake robot, we focused on developing mathematical models for realistic and efficient simulations of planar and 3D snake robot motion. In this chapter we shift the main focus to control. To this end, we develop a new 2D model of the snake robot based on the standard robot manipulator model commonly employed to model robot manipulators (usually with less degrees of freedom than a snake robot) and marine vessels (Fossen, 2002; Spong and Vidyasagar, 1989). This model is denoted as the process plant model. By employing this particular form of the model, a large area of techniques regarding control and stability analysis of such systems becomes available. However, this representation of a model of a snake robot (with such a large number of degrees of freedom) is not advantageous for simulation as will be shown in Section 7.3.5 so the 2D model developed earlier in this chapter is still important. The process plant model contains the global orientation of the snake robot. However, to make the stability analysis manageable we derive a control plant model from the process plant model where this orientation is

removed. In this chapter, we employ one group of the techniques available for the robot manipulator model, called input-output linearization (Khalil, 2002; Nijmeijer and Schaft, 1990) to develop a controller for tracking control of the snake robot joints, and we show, using the control plant model, that the joint angles converge exponentially to their desired trajectory. The non-linear control techniques employed in this chapter are generally only fit to cope with smooth non-linear systems. Moreover, few control techniques exists for non-smooth systems such as the 2D model presented in Chapter 3. Therefore, in order to make the control problem tractable, we approximate the Coulomb friction with a smooth friction model and design a non-linear control law for the resulting smooth system. Note that we do not give a reference for the global position and orientation of the snake robot which is beyond the scope of this thesis. Instead, we focus on proving that each joint is able to track a desired reference angle.

This chapter is based on Transeth *et al.* (2007a) and it is organized as follows. Section 6.2 presents mathematical models, both based on the standard dynamics of a robot manipulator, of the snake robot and used for simulation and stability analysis, respectively. The controller and the stability analysis are given in Section 6.3 and a summary of the chapter is found in Section 6.4.

6.2 Mathematical Modelling

In this section we first develop the mathematical model of a planar snake robot without wheels based on the standard model for a robot manipulator with minimal coordinates found in for example Spong and Vidyasagar (1989). We denote this model the *process plant model* and implement it for two different constitutive laws for Coulomb friction. The process plant model is necessary for us to be able to proceed to the next step which is to reformulate the process plant model to obtain a *control plant model* used both for designing the joint controllers and for deriving stability results for the closed-loop system. The process plant model differs mainly from the 2D model presented in Chapter 3 in that the model is described by an ordinary differential equation (ODE) instead of an equality of measures, a smooth approximation to the Coulomb friction is employed, and minimal coordinates are used to describe the kinematics.

The snake robot differs from a regular robot manipulator in that external forces act on each link instead of just the end-effector. This calls for a more extensive use of the various manipulator Jacobians employed

to find the velocities of each link from the time-derivative of the minimal coordinates and also to include the friction forces in the system dynamics.

Before we proceed with the modelling, let us first sum up the various planar models we have discussed so far. The ‘2D model’ is presented in Chapter 3 and is developed for efficient simulations. In this section, we also develop a 2D model denoted the ‘process plant model’, and this model can also be employed for simulation. However, it is not advantageous for simulation partly due to the large number of degrees of freedom of a snake robot which results in large system matrices. Nonetheless, it may be convenient to employ the process plant model for simulation just in order to test the controller since many of the system matrices utilized in the controller are already available in the numerical treatment of the model. From the process plant model, we will develop a ‘control plant model’ for designing the model-based controller and for stability analysis. We will sometimes refer to the process plant model as the ‘robot model’ throughout this thesis in order to distinguish it from the ‘2D model’ from Chapter 3.

6.2.1 Process Plant Model

In this section we derive a mathematical model of a wheel-less snake robot based on the Euler-Lagrange equations as found in, for example, Spong and Vidyasagar (1989). This model is denoted the process plant model. We present two models for the Coulomb friction between the snake robot and the ground. The first model is a smooth approximation to the Coulomb friction. This approximation is performed to be able to employ standard techniques for controller design. The second friction model is a model of *set-valued* Coulomb friction using a set-valued force law (similar to Chapter 3). We implement the latter model to check how the controller works for a more realistic model of a snake robot.

In the following we will first describe the kinematics and then we present the dynamics of the snake robot.

Kinematics and Velocities

The 2D snake robot model presented in this chapter consists of $n - 2$ equal links connected by $n - 3$ revolute joints, each with one degree of freedom (DOF). See Figure 6.1 for an illustration of a snake robot with four links. The inertial position and orientation of the snake robot adds three DOF to the model. As opposed to n being the number of links in the previous chapters, notice that n now describes the number of DOF of the system. To

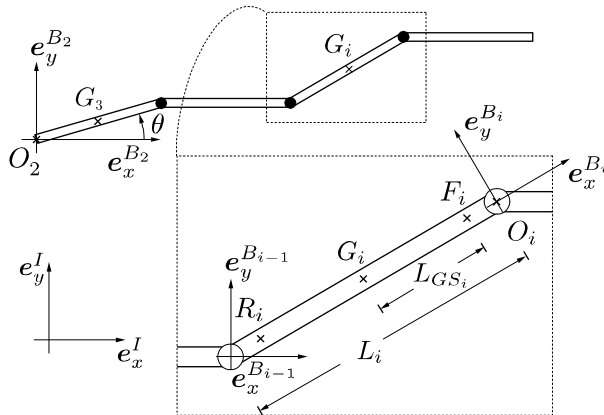


Figure 6.1: Snake robot with 4 links. Notice that G_i is the centre of gravity of link $i = 5$ in the picture.

be able to give a minimal coordinate representation of the snake robot in an inertial frame, we need to include *virtual* joints in the model as introduced for snake robots in Liljebäck *et al.* (2005). These joints consist of two prismatic joints and one revolute joint to give the position and orientation, respectively, of the first link of the snake robot with respect to an inertial frame. Hence, the virtual joints are not physical joints located on the snake robot, but rather just way to denote the position and orientation of the first link of the snake robot in the plane. Due to these virtual joints, we also obtain virtual links (link 1 and 2) in the model which do not have mass. Each (real) link $i = 3, \dots, n$ has length L_i and the position of the centre of gravity (CG) is given by the point G_i at the middle of link i as depicted in Figure 6.1. We lump the multi-point contact between link i and the ground into the two points, F_i and R_i , each of distance L_{GS_i} from G_i . We employ these two contact points to keep the model as simple as possible while still accounting for the friction forces that arise from a purely rotational motion of the link (this would not be the case if G_i were chosen as the contact point). We only included one contact point in the planar model in Chapter 3 to keep the model as simple as possible. However, in this chapter the friction forces from a purely rotational motion need to be included in the model since they become important for the subsequent stability analysis.

We denote the origin of an earth-fixed frame $I = \{O_I, e_x^I, e_y^I\}$ with origin O_I as an approximation to an inertial reference frame. A general

notation used throughout this chapter is that a position vector from the origin of frame I to a point A is given by $\mathbf{r}_A \in \mathbb{R}^2$ and a vector from point A to point B is written as \mathbf{r}_{AB} . Vector \mathbf{r}_A described in frame I is written as ${}_I\mathbf{r}_A$. The body fixed frame $B_i = \{O_i, \mathbf{e}_x^{B_i}, \mathbf{e}_y^{B_i}\}$, for $i = 3, \dots, n$, is attached to the front of link i and $\mathbf{e}_x^{B_i}$ and $\mathbf{e}_y^{B_i}$ are pointing as shown in Figure 6.1. The remaining frames B_i , $i = 0, \dots, 2$, are utilized to give the inertial position and orientation of the first (real) link of the snake robot (later referred to as the ‘tail’ of the snake robot) and will be described shortly. The transformation of a vector \mathbf{r} between frame I and B_i is given by

$${}_I\mathbf{r} = \mathbf{R}_{B_i}^I {}_{B_i}\mathbf{r}, \quad (6.1)$$

where $\mathbf{R}_{B_i}^I$ is the corresponding rotation matrix.

We define for convenience $B_0 = B_1 = I$ and let ${}_I\mathbf{r}_{O_2} = [x_{O_2} \ y_{O_2}]^T$ be the position of the rear part of the tail of the snake robot (link 3) where $\mathbf{e}_x^{B_2} = \mathbf{e}_x^I$ and $\mathbf{e}_y^{B_2} = \mathbf{e}_y^I$. The orientation of link 3 is given by $\theta \in \mathcal{S}^1$. Let $\phi_j \in \mathcal{S}^1$, $j = 1, \dots, n-3$ be the relative joint angles between link $j+2$ and $j+3$ and define ${}_I\mathbf{r}_{O_0} = {}_I\mathbf{r}_{O_1} = {}_I\mathbf{r}_{G_1} = {}_I\mathbf{r}_{G_2} = [0 \ 0]^T$. Subsequently, the positions of the origin of each frame and the centre of gravity of each link are found as

$$\mathbf{r}_{O_i} = \mathbf{r}_{O_{i-1}} + L_i \mathbf{e}_x^{B_i}, \quad \mathbf{r}_{G_i} = \mathbf{r}_{O_i} - \frac{L_i}{2} \mathbf{e}_x^{B_i}, \quad (6.2)$$

respectively, for $i = 3, \dots, n$.

The generalized coordinates for the snake robot are

$$\mathbf{q} = [x_{O_2} \ y_{O_2} \ \theta \ \phi_1 \ \dots \ \phi_{n-3}]^T \in \mathbb{R}^n, \quad (6.3)$$

where ϕ_j is the joint angle between link $j+2$ and $j+3$. Hence, x_{O_2} , y_{O_2} and θ represent the virtual joints, while $\phi_1, \phi_2, \dots, \phi_{n-3}$ represent the actual joints of the snake robot.

A common way to define the mapping from the time-derivative $\dot{\mathbf{q}}$ of the generalized coordinates to the linear \mathbf{v} and angular ω velocity of link i is to employ something called a manipulator Jacobian (Spong and Vidyasagar, 1989). In the following, we will derive these Jacobians for the points G_i in order obtain the mass matrix, and the Jacobians for F_i and R_i to be able to include friction in the system dynamics.

The translational velocity ${}_I\mathbf{v}_{G_i}$ of the centre of gravity of link $i = 3, \dots, n$ is found from

$${}_I\mathbf{v}_{G_i} = \mathbf{J}_{v_{G_i}}^P(\mathbf{q})\dot{\mathbf{q}}, \quad (6.4)$$

where we used the superscript P to denote the process plant model and the Jacobian $\mathbf{J}_{v_{G_i}}^P(\mathbf{q})$ is found from

$$\mathbf{J}_{v_{G_i}}^P(\mathbf{q}) = [(\mathbf{J}_{v_{G_i}}^P)_1 \quad (\mathbf{J}_{v_{G_i}}^P)_2 \quad \cdots \quad (\mathbf{J}_{v_{G_i}}^P)_n] \in \mathbb{R}^{2 \times n}, \quad (6.5)$$

with

$$(\mathbf{J}_{v_{G_i}}^P)_k = \begin{cases} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} (I\mathbf{r}_{G_i} - I\mathbf{r}_{O_{k-1}}) & , \text{revolute joint } k \leq i \\ I\mathbf{e}_\zeta^{B_{k-1}} & , \text{prismatic joint } k \leq i \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} & , k > i, \end{cases} \quad (6.6)$$

and ζ is the axis of elongation for the prismatic joint k . The rotational velocity ω_i of link $i = 3, \dots, n$ is

$$\omega_i = \mathbf{J}_{\omega_i}^P(\mathbf{q})\dot{\mathbf{q}}, \quad (6.7)$$

where $\mathbf{J}_{\omega_i}^P(\mathbf{q}) \in \mathbb{R}^{1 \times n}$ is

$$\mathbf{J}_{\omega_i}^P(\mathbf{q}) = [(J_{\omega_i}^P)_1 \quad (J_{\omega_i}^P)_2 \quad \cdots \quad (J_{\omega_i}^P)_n] \quad (6.8)$$

where

$$(J_{\omega_i}^P)_k = \begin{cases} 1 & , \text{revolute joint } k \leq i \\ 0 & , \text{prismatic joint } k \leq i \\ 0 & , k > i. \end{cases} \quad (6.9)$$

The above Jacobians are employed in the system dynamics to find the mass matrix as will be shown in the subsequent section. Since we have defined the friction to act on the links at positions different from the centre of gravity, we also need to define an additional set of Jacobians. The translational velocities of F_i and R_i are found from

$$I\mathbf{v}_{\xi_i} = \mathbf{J}_{v_{\xi_i}}^P(\mathbf{q})\dot{\mathbf{q}}, \quad (6.10)$$

for $\xi_i = F_i, R_i$ and $i = 3, \dots, n$, where the Jacobian $\mathbf{J}_{v_{\xi_i}}^P(\mathbf{q}) \in \mathbb{R}^{2 \times n}$ is found from

$$\mathbf{J}_{v_{\xi_i}}^P(\mathbf{q}) = [(\mathbf{J}_{v_{\xi_i}}^P)_1 \quad (\mathbf{J}_{v_{\xi_i}}^P)_2 \quad \cdots \quad (\mathbf{J}_{v_{\xi_i}}^P)_n], \quad (6.11)$$

where

$$(\mathbf{J}_{v_{\xi_i}}^P)_k = \begin{cases} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} (I\mathbf{r}_{\xi_i} - I\mathbf{r}_{O_{k-1}}) & , \text{revolute joint } k \leq i \\ I\mathbf{e}_\zeta^{B_{k-1}} & , \text{prismatic joint } k \leq i \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} & , k > i, \end{cases} \quad (6.12)$$

and

$$\mathbf{r}_{F_i} = \mathbf{r}_{G_i} + L_{GS_i} \mathbf{e}_x^{B_i}, \quad \mathbf{r}_{R_i} = \mathbf{r}_{G_i} - L_{GS_i} \mathbf{e}_x^{B_i}. \quad (6.13)$$

Dynamics

The equations of motion for the snake robot are based on the Euler-Lagrange equations and are formulated in a standard matrix form for planar robot manipulators (see for example Spong and Vidyasagar (1989)) as follows:

$$\begin{aligned} \mathbf{M}_P(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}_P(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} &= \mathbf{P}_\phi \boldsymbol{\tau} \\ &+ \sum_{i=3}^n \left((\mathbf{J}_{v_{F_i}}^P)^T \mathbf{I} \mathbf{F}_{F_i}^P + (\mathbf{J}_{v_{R_i}}^P)^T \mathbf{I} \mathbf{F}_{R_i}^P \right), \end{aligned} \quad (6.14)$$

where the mass matrix $\mathbf{M}_P(\mathbf{q}) \in \mathbb{R}^{n \times n}$, the Coriolis and centripetal matrix $\mathbf{C}_P(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$, the applied joint torques $\mathbf{P}_\phi \boldsymbol{\tau}$ and the friction forces $\mathbf{I} \mathbf{F}_{F_i}^P$ and $\mathbf{I} \mathbf{F}_{R_i}^P$ will be described in the following paragraphs. We have included the friction forces $\mathbf{I} \mathbf{F}_{F_i}^P$ and $\mathbf{I} \mathbf{F}_{R_i}^P$ acting on link i by using the corresponding Jacobians. This is similar to how external forces acting on the end-effector of a robot manipulator is included in its system dynamics.

The mass matrix is written in a standard way (Spong and Vidyasagar, 1989) as

$$\mathbf{M}_P(\mathbf{q}) = \sum_{i=3}^n \left(m_i (\mathbf{J}_{v_{G_i}}^P)^T \mathbf{J}_{v_{G_i}}^P + \Theta_{\psi_i} (\mathbf{J}_{\omega_i}^P)^T \mathbf{J}_{\omega_i}^P \right), \quad (6.15)$$

where m_i and Θ_{ψ_i} are the mass and moment of inertia of link i with respect to its centre of mass, respectively ($m_1 = m_2 = \Theta_{\psi_1} = \Theta_{\psi_2} = 0$ due to the virtual joints and are therefore not included in $\mathbf{M}_P(\mathbf{q})$). By defining $\mathbf{M}_P(\mathbf{q}) = \{M_{P_{ij}}\}$ and $\mathbf{C}_P(\mathbf{q}, \dot{\mathbf{q}}) = \{C_{P_{ij}}\}$ we can find the elements of the Coriolis and centripetal matrix as

$$C_{P_{ij}} = \sum_{k=1}^n c_{ijk} \dot{q}_k, \quad (6.16)$$

where

$$c_{ijk} = \frac{1}{2} \left(\frac{\partial M_{P_{ij}}}{\partial q_k} + \frac{\partial M_{P_{ik}}}{\partial q_j} - \frac{\partial M_{P_{kj}}}{\partial q_i} \right). \quad (6.17)$$

The torques $\boldsymbol{\tau} \in \mathbb{R}^{n-3}$ applied by the motors in the joints are mapped into the system dynamics by the constant matrix

$$\mathbf{P}_\phi = \begin{bmatrix} \mathbf{0}_{3 \times n-3} \\ \mathbf{I}_{n-3 \times n-3} \end{bmatrix}, \quad (6.18)$$

where $\mathbf{I}_{n-3 \times n-3} \in \mathbb{R}^{n-3 \times n-3}$ is the identity matrix. The friction between the ground surface and the snake robot links is modelled as a combination of Coulomb friction $\mathbf{F}_{F_i}^{PC}$, $\mathbf{F}_{R_i}^{PC}$ and viscous friction $\mathbf{F}_{F_i}^{PV}$, $\mathbf{F}_{R_i}^{PV}$. Hence, the total friction force on the front and rear part of link i is

$$\mathbf{F}_{\xi_i}^P = \mathbf{F}_{\xi_i}^{PC} + \mathbf{F}_{\xi_i}^{PV}, \quad (6.19)$$

for $\xi_i = F_i, R_i$. We formulate the Coulomb friction force law in two ways. First, we present a smooth approximation to Coulomb friction $\mathbf{F}_{\xi_i}^{PCc}$ and then we show how to find the set-valued Coulomb friction $\mathbf{F}_{\xi_i}^{PCt}$.

Coulomb friction is often modelled with a sign-function (Egeland and Gravdahl, 2002; Saito *et al.*, 2002) and the arctan-function can be employed as an approximation to the sign-function. Define arctan of a vector argument $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_n]$ as

$$\arctan(\mathbf{x}) = \begin{bmatrix} \arctan(x_1) \\ \arctan(x_2) \\ \vdots \\ \arctan(x_n) \end{bmatrix}. \quad (6.20)$$

Then we can find a smooth approximation of the Coulomb friction $\mathbf{F}_{\xi_i}^{PCc}$ from

$${}_{B_i} \mathbf{F}_{\xi_i}^{PCc} = -F_{N_i} \mathbf{D}_{\mu_C} \frac{1}{\pi} \arctan\left(\frac{{}_{B_i} \mathbf{v}_{\xi_i}}{\epsilon_C}\right), \quad (6.21)$$

where the total normal force acting on link i is $F_{N_i} = m_i g$, g is the acceleration of gravity, $\epsilon_C > 0$ is a small constant that determines how close the arctan-function approximates the sign-function, the diagonal matrix

$$\mathbf{D}_{\mu_C} = \begin{bmatrix} \mu_{C_x} & 0 \\ 0 & \mu_{C_y} \end{bmatrix}, \quad (6.22)$$

and $\mu_{C_x} \in \mathbb{R}^+$, $\mu_{C_y} \in \mathbb{R}^+$ are the Coulomb friction coefficients longitudinal and lateral to each link, respectively. Notice that the total normal force F_{N_i} is divided equally between the two contact points on each link in (6.21). This can be seen from (6.21) since

$$\lim_{x \rightarrow \infty} \frac{2}{\pi} \arctan(x) = 1. \quad (6.23)$$

In addition, we employ the common assumption for a planar model that the normal forces are equally distributed along the snake robot (Ma, 2001; Saito *et al.*, 2002).

Set-valued Coulomb friction $\mathbf{F}_{\xi_i}^{P_{C_t}}$ is modelled using a set-valued force law based on the framework of non-smooth dynamics and convex analysis, and an expression for the friction is given with the algebraic inclusion

$$-B_i \mathbf{v}_{\xi_i} \in N_{C_T} \left(B_i \mathbf{F}_{\xi_i}^{P_{C_t}} \right), \quad (6.24)$$

where $N_{C_T}(\cdot)$ is the normal cone to the set C_T given by the ellipse:

$$C_T = \left\{ B_i \mathbf{F}_{\xi_i}^{P_{C_t}} \mid \frac{\left(B_i F_{\xi_{ix}}^{P_{C_t}} \right)^2}{\left(\mu_{C_x} \frac{F_{N_i}}{2} \right)^2} + \frac{\left(B_i F_{\xi_{iy}}^{P_{C_t}} \right)^2}{\left(\mu_{C_y} \frac{F_{N_i}}{2} \right)^2} \leq 1 \right\}, \quad (6.25)$$

where we have defined $B_i \mathbf{F}_{\xi_i}^{P_{C_t}} = \begin{bmatrix} B_i F_{\xi_{ix}}^{P_{C_t}} & B_i F_{\xi_{iy}}^{P_{C_t}} \end{bmatrix}^T$. A more thorough explanation of set-valued force laws is found in Section 3.5.2. However, the notation is different. This is because in this chapter we try to conform to the standard notation employed for modelling of robot manipulators. Note that when the set-valued force law (6.24) is employed to describe Coulomb friction, then the time-stepping method with the Modified Newton Algorithm can be employed for simulation. Hence, the equation of motion (6.14) has to be written as an equality of measures, e.g., on the form (3.43) on page 54 and the same discretization technique as described in Section 3.6.1 can be employed. This procedure will not be repeated here.

The viscous friction force can be expressed in frame B_i as

$$B_i \mathbf{F}_{\xi_i}^{P_V} = -\frac{F_{N_i}}{2} \mathbf{D}_{\mu_V} B_i \mathbf{v}_{\xi_i}, \quad \mathbf{D}_{\mu_V} = \begin{bmatrix} \mu_{V_x} & 0 \\ 0 & \mu_{V_y} \end{bmatrix}, \quad (6.26)$$

where $\mu_{V_x} \in \mathbb{R}^+$ and $\mu_{V_y} \in \mathbb{R}^+$ are the viscous friction coefficients longitudinal and lateral to each link, respectively.

6.2.2 Control Plant Model

In this section the control plant model is derived from the process plant model. We approximate the set-valued Coulomb friction law (6.24) by the smooth non-linear friction model (6.21). This enables us to apply input-output linearization control techniques developed for smooth systems. We reformulate the model such that the position ${}^I \mathbf{r}_{O_2} = [x_{O_2} \ y_{O_2}]^T$ and the orientation θ of the first link of the snake robot are removed from the state equations in (6.14). This allows for an infinite number of rotations of the snake robot without affecting the controller (which is based on the control plant model). We will see in the following that the translational and rotational velocity of the first link will still be included in the model.

Kinematics

In this section we present the new kinematics where the inertial position and orientation of the snake robot have been decoupled from the rest of the system dynamics. To achieve this, we define a new vector $\boldsymbol{\nu}$ of velocity coordinates that includes the velocity ${}_I\mathbf{v}_{O_2} = {}_I\dot{\mathbf{r}}_{O_2}$ in frame B_3 attached to the first link: ${}_{B_3}\mathbf{v}_{O_2} = (\mathbf{R}_{B_3}^I)^\top {}_I\mathbf{v}_{O_2}$. Hence,

$$\dot{\mathbf{q}} = \mathbf{T}_q \boldsymbol{\nu} \quad (6.27)$$

$$\begin{aligned} & \Updownarrow \\ \boldsymbol{\nu} &= \mathbf{T}_q^\top \dot{\mathbf{q}}, \end{aligned} \quad (6.28)$$

where

$$\mathbf{T}_q = \mathbf{T}_q(\theta) = \begin{bmatrix} \mathbf{R}_{B_3}^I & \mathbf{0}_{2 \times n-2} \\ \mathbf{0}_{n-2 \times 2} & \mathbf{I}_{n-2 \times n-2} \end{bmatrix}, \quad (6.29)$$

$$\boldsymbol{\nu} = \begin{bmatrix} {}_{B_3}\mathbf{v}_{O_2} \\ \dot{\theta} \\ \dot{\boldsymbol{\phi}} \end{bmatrix} \in \mathbb{R}^n, \quad \boldsymbol{\phi} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{n-3} \end{bmatrix} \in \mathcal{S}^{n-3}. \quad (6.30)$$

To be able to cancel θ from the model, we describe all translational velocities in frame B_3 . Let us investigate the new expression for the translational velocity \mathbf{v}_{F_i} of the front part of link i as an example: By employing the transformation ${}_I\mathbf{v}_{F_i} = \mathbf{R}_{B_3}^I {}_{B_3}\mathbf{v}_{F_i}$ and inserting (6.27) into (6.10), we find that

$${}_{B_3}\mathbf{v}_{F_i} = \mathbf{J}_{v_{F_i}}(\boldsymbol{\phi}) \boldsymbol{\nu}, \quad (6.31)$$

where

$$\mathbf{J}_{v_{F_i}}(\boldsymbol{\phi}) = (\mathbf{R}_{B_3}^I)^\top \mathbf{J}_{v_{F_i}}^P \mathbf{T}_q. \quad (6.32)$$

The expression for ${}_{B_3}\mathbf{v}_{R_i}$ is the same as (6.31) except that F_i needs to be replaced by R_i . The Jacobian $\mathbf{J}_{v_{R_i}}(\boldsymbol{\phi})$ is found similarly to $\mathbf{J}_{v_{F_i}}(\boldsymbol{\phi})$ as

$$\mathbf{J}_{v_{R_i}}(\boldsymbol{\phi}) = (\mathbf{R}_{B_3}^I)^\top \mathbf{J}_{v_{R_i}}^P \mathbf{T}_q. \quad (6.33)$$

The Jacobian $\mathbf{J}_{v_{F_i}}(\boldsymbol{\phi})$ (or $\mathbf{J}_{v_{R_i}}(\boldsymbol{\phi})$) which relates the new generalized velocities $\boldsymbol{\nu}$ and the translational velocity ${}_{B_3}\mathbf{v}_{F_i}$ (or ${}_{B_3}\mathbf{v}_{R_i}$) does not depend on the earth-fixed rotation angle θ of the snake robot tail (link 3). Namely, the translational velocities of the snake robot tail in $\boldsymbol{\nu}$ are given in frame B_3 and an earth-fixed rotation angle is therefore no longer necessary to find the translational link velocities in frame B_3 from $\boldsymbol{\nu}$.

Dynamics

In this section we employ the transformations above to reduce the model of the dynamics (6.14). From (6.27), we find that

$$\ddot{\mathbf{q}} = \dot{\mathbf{T}}_q \boldsymbol{\nu} + \mathbf{T}_q \dot{\boldsymbol{\nu}}, \quad (6.34)$$

and this equality together with (6.27) and the relation ${}_I \mathbf{F}_{\xi_i}^P = \mathbf{R}_{B_3}^I {}_{B_3} \mathbf{F}_{\xi_i}^P$ are inserted into the process plant model (6.14) to obtain

$$\begin{aligned} \mathbf{M}_P(\mathbf{q}) \mathbf{T}_q \dot{\boldsymbol{\nu}} + \left(\mathbf{M}_P(\mathbf{q}) \dot{\mathbf{T}}_q + \mathbf{C}_P(\mathbf{q}, \dot{\mathbf{q}}) \mathbf{T}_q \right) \boldsymbol{\nu} &= \mathbf{P}_\phi \boldsymbol{\tau} \\ + \sum_{i=3}^n \left((\mathbf{J}_{v_{F_i}}^P)^T \mathbf{R}_{B_3}^I {}_{B_3} \mathbf{F}_{F_i}^P + (\mathbf{J}_{v_{R_i}}^P)^T \mathbf{R}_{B_3}^I {}_{B_3} \mathbf{F}_{R_i}^P \right). \end{aligned} \quad (6.35)$$

By inspection \mathbf{T}_q in (6.29) and \mathbf{P}_ϕ in (6.18), we see that

$$\mathbf{T}_q^T \mathbf{P}_\phi = \mathbf{P}_\phi. \quad (6.36)$$

We pre-multiply (6.35) with \mathbf{T}_q^T , and using (6.32) and (6.36), we obtain

$$\begin{aligned} \mathbf{M}(\boldsymbol{\phi}) \dot{\boldsymbol{\nu}} + \mathbf{C}(\dot{\boldsymbol{\theta}}, \boldsymbol{\phi}, \dot{\boldsymbol{\phi}}) \boldsymbol{\nu} &= \mathbf{P}_\phi \boldsymbol{\tau} \\ + \sum_{i=3}^n \left((\mathbf{J}_{v_{F_i}}^T)^T {}_{B_3} \mathbf{F}_{F_i} + (\mathbf{J}_{v_{R_i}}^T)^T {}_{B_3} \mathbf{F}_{R_i} \right). \end{aligned} \quad (6.37)$$

where

$$\mathbf{M}(\boldsymbol{\phi}) = \mathbf{T}_q^T \mathbf{M}_P(\mathbf{q}) \mathbf{T}_q, \quad (6.38)$$

$$\mathbf{C}(\dot{\boldsymbol{\theta}}, \boldsymbol{\phi}, \dot{\boldsymbol{\phi}}) = \mathbf{T}_q^T \left(\mathbf{M}_P(\mathbf{q}) \dot{\mathbf{T}}_q + \mathbf{C}_P(\mathbf{q}, \dot{\mathbf{q}}) \mathbf{T}_q \right), \quad (6.39)$$

and we have used that

$$\mathbf{F}_{\xi_i} = \mathbf{F}_{\xi_i}^C + \mathbf{F}_{\xi_i}^V = \mathbf{F}_{\xi_i}^P, \quad (6.40)$$

for $\xi_i = F_i, R_i$ where the viscous friction term $\mathbf{F}_{\xi_i}^V = \mathbf{F}_{\xi_i}^{P_V}$ is the same as in (6.26), while the smooth approximation $\mathbf{F}_{\xi_i}^C = \mathbf{F}_{\xi_i}^{P_{C_c}}$ in (6.21) of the Coulomb friction in the process plant model is employed. We note from (6.38)-(6.39) that the new system matrices $\mathbf{M}(\boldsymbol{\phi})$ and $\mathbf{C}(\dot{\boldsymbol{\theta}}, \boldsymbol{\phi}, \dot{\boldsymbol{\phi}})$ are independent of the global orientation θ . This is because our choice of new coordinates renders the new equation of motion (6.37) independent of θ since all link velocities are described in frame B_3 instead of frame I .

6.3 Joints Control by Input-output Linearization

In this section we employ an input-output (IO) linearizing controller to be able to accurately track the desired joint angles and prove mathematically that this closed-loop system is stable. Inspired by Spong (1996), we divide the system (6.37) into two parts: one part for the directly actuated generalized coordinates called *active (or real) joints*, and one part for the unactuated degrees of freedom which are the *passive (or virtual) joints*. We will then show that the origin of the tracking error dynamics of the active joints are asymptotically stable and that the IO-linearizing controller can be employed.

6.3.1 Control Plant Model Reformulation

We begin by dividing the control plant model (6.37) into two parts (as performed for the standard robot manipulator model in Spong (1996)): one part for the directly actuated generalized coordinates ϕ called *active joints* and one part for the unactuated generalized velocities \mathbf{w} which are the *passive joints*. We define the unactuated generalized velocities as

$$\mathbf{w} = \begin{bmatrix} {}_{B_3}\mathbf{v}_{O_2} \\ \dot{\theta} \end{bmatrix} \in \mathbb{R}^3, \quad (6.41)$$

and we define a constant matrix employed often in the following as

$$\mathbf{P}_w = \begin{bmatrix} \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{n-3 \times 3} \end{bmatrix} \in \mathbb{R}^{n \times 3}. \quad (6.42)$$

Define

$$\mathbf{h} = \mathbf{h}(\phi, \boldsymbol{\nu}) = \mathbf{C}(\dot{\theta}, \phi, \dot{\phi}) \boldsymbol{\nu}, \quad (6.43)$$

$$\mathbf{g} = \mathbf{g}(\phi, \boldsymbol{\nu}) = \sum_{i=3}^n \left((\mathbf{J}_{v_{F_i}})^T {}_{B_3}\mathbf{F}_{F_i} + (\mathbf{J}_{v_{R_i}})^T {}_{B_3}\mathbf{F}_{R_i} \right), \quad (6.44)$$

where

$$\mathbf{h}(\phi, \boldsymbol{\nu}) = \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \end{bmatrix} \in \mathbb{R}^n, \quad \mathbf{g}(\phi, \boldsymbol{\nu}) = \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \end{bmatrix} \in \mathbb{R}^n, \quad (6.45)$$

and

$$\mathbf{h}_1 = \mathbf{P}_w^T \mathbf{C} \begin{pmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\phi} \end{pmatrix} \boldsymbol{\nu} \in \mathbb{R}^3, \quad (6.46)$$

$$\mathbf{h}_2 = \mathbf{P}_\phi^T \mathbf{C} \begin{pmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\phi} \end{pmatrix} \boldsymbol{\nu} \in \mathbb{R}^{n-3}, \quad (6.47)$$

$$\mathbf{g}_1 = \mathbf{P}_w^T \sum_{i=3}^n \left((\mathbf{J}_{v_{Fi}})^T {}_{B_3} \mathbf{F}_{F_i} + (\mathbf{J}_{v_{Ri}})^T {}_{B_3} \mathbf{F}_{R_i} \right) \in \mathbb{R}^3, \quad (6.48)$$

$$\mathbf{g}_2 = \mathbf{P}_\phi^T \sum_{i=3}^n \left((\mathbf{J}_{v_{Fi}})^T {}_{B_3} \mathbf{F}_{F_i} + (\mathbf{J}_{v_{Ri}})^T {}_{B_3} \mathbf{F}_{R_i} \right) \in \mathbb{R}^{n-3}. \quad (6.49)$$

From (6.30) and (6.41)-(6.49), the control plant model (6.37) may be written as follows:

$$\mathbf{M}_{11} \dot{\boldsymbol{\omega}} + \mathbf{M}_{12} \ddot{\boldsymbol{\phi}} + \mathbf{h}_1 = \mathbf{g}_1 \quad (6.50)$$

$$\mathbf{M}_{21} \dot{\boldsymbol{\omega}} + \mathbf{M}_{22} \ddot{\boldsymbol{\phi}} + \mathbf{h}_2 = \mathbf{g}_2 + \boldsymbol{\tau}, \quad (6.51)$$

where the mass matrix $\mathbf{M}(\boldsymbol{\phi})$ in (6.38) is divided as

$$\mathbf{M}(\boldsymbol{\phi}) = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix}, \quad \mathbf{M}_{11} \in \mathbb{R}^{3 \times 3}. \quad (6.52)$$

In the forthcoming section, we denote (6.50) as the *tracking dynamics* and (6.51) as the *input-output dynamics*.

6.3.2 Controller Design

In this section we derive an input-output linearizing controller such that the input-output dynamics (6.51) is able to asymptotically track a bounded reference $\boldsymbol{\phi}_d(t)$, with bounded derivatives $\dot{\boldsymbol{\phi}}_d(t)$, $\ddot{\boldsymbol{\phi}}_d(t)$.

Since $\mathbf{M}(\boldsymbol{\phi})$ is positive definite then the upper left submatrix \mathbf{M}_{11} in (6.52) of $\mathbf{M}(\boldsymbol{\phi})$ is positive definite and invertible and we can find an expression for $\dot{\boldsymbol{\omega}}$ from (6.50):

$$\dot{\boldsymbol{\omega}} = \mathbf{M}_{11}^{-1} \mathbf{g}_1 - \mathbf{M}_{11}^{-1} \mathbf{M}_{12} \ddot{\boldsymbol{\phi}} - \mathbf{M}_{11}^{-1} \mathbf{h}_1, \quad (6.53)$$

and insert the new expression for $\dot{\boldsymbol{\omega}}$ into (6.51) to obtain

$$\bar{\mathbf{M}}_{22} \ddot{\boldsymbol{\phi}} + \bar{\mathbf{h}}_2 = \bar{\mathbf{g}}_2 + \boldsymbol{\tau}, \quad (6.54)$$

where

$$\bar{\mathbf{M}}_{22} = \mathbf{M}_{22} - \mathbf{M}_{21} \mathbf{M}_{11}^{-1} \mathbf{M}_{12}, \quad (6.55)$$

$$\bar{\mathbf{h}}_2 = \mathbf{h}_2 - \mathbf{M}_{21} \mathbf{M}_{11}^{-1} \mathbf{h}_1, \quad (6.56)$$

$$\bar{\mathbf{g}}_2 = \mathbf{g}_2 - \mathbf{M}_{21} \mathbf{M}_{11}^{-1} \mathbf{g}_1, \quad (6.57)$$

Note that $\bar{\mathbf{M}}_{22}$ is symmetric and positive definite (Gu and Xu, 1993). The friction and system dynamics may now be cancelled from (6.54) by employing the feedback transformation

$$\boldsymbol{\tau} = \bar{\mathbf{M}}_{22}\mathbf{u} + \bar{\mathbf{h}}_2 - \bar{\mathbf{g}}_2, \quad (6.58)$$

where $\mathbf{u} \in \mathbb{R}^{n-3}$ is a new control input. The input-output dynamics (6.51) is now reduced to

$$\ddot{\boldsymbol{\phi}} = \mathbf{u}. \quad (6.59)$$

Hence, by using the stabilizing feedback law for \mathbf{u} :

$$\mathbf{u} = \ddot{\boldsymbol{\phi}}_d - \mathbf{K}_D (\dot{\boldsymbol{\phi}} - \dot{\boldsymbol{\phi}}_d) - \mathbf{K}_P (\boldsymbol{\phi} - \boldsymbol{\phi}_d), \quad (6.60)$$

we obtain a linear and asymptotically stable error dynamics for the input-output dynamics (6.51) for the $n - 3 \times n - 3$ positive definite, diagonal constant matrices \mathbf{K}_D and \mathbf{K}_P . We see from (6.44) that $\bar{\mathbf{g}}_2$ in the controller (6.58) contains the Coulomb and viscous friction terms. Now, the reason for the approximation (6.21) of the Coulomb friction becomes apparent. Namely, due to the set-valued Coulomb friction in the process plant model we are not able to compensate exactly for the Coulomb friction at zero velocity since the friction force can be anywhere in the set C_T in (6.25). To be able to include the case of zero velocity in the stability proof we therefore approximate the Coulomb friction with the smooth friction model as in (6.21). We see from (6.58) and (6.60) that the computed torque $\boldsymbol{\tau}$ only depends on $\boldsymbol{\phi}$, $\boldsymbol{\nu}$ and $\boldsymbol{\phi}_d$ and its time-derivatives. However, to implement the controller (6.58), (6.60) on an actual snake robot we would also have to find θ and ${}^I\mathbf{v}_{O_2}$ since ${}_{B_3}\mathbf{v}_{O_2}$ cannot be measured directly and has to be calculated from ${}_{B_3}\mathbf{v}_{O_2} = (\mathbf{R}_{B_3}^I)^T {}^I\mathbf{v}_{O_2}$.

6.3.3 Final Results on Input-output Linearization

We start this section by stating the following theorem.

Theorem 6.1 *Assume that the desired trajectory $\boldsymbol{\phi}_d(t) \in \mathbb{R}^{n-3}$ for the joints is bounded with bounded acceleration $\ddot{\boldsymbol{\phi}}_d(t)$ and bounded velocity $\|\dot{\boldsymbol{\phi}}_d(t)\| \leq c_{\dot{\boldsymbol{\phi}}}$ for all t for some sufficiently small $c_{\dot{\boldsymbol{\phi}}} > 0$. Along any trajectory of the closed-loop (6.50)-(6.51), (6.58), (6.60), the joint tracking error $\mathbf{e}_{\boldsymbol{\phi}}(t)$, defined by*

$$\mathbf{e}_{\boldsymbol{\phi}}(t) = \boldsymbol{\phi}(t) - \boldsymbol{\phi}_d(t), \quad (6.61)$$

converges to zero exponentially and the velocity vector $\mathbf{w}(t)$ remains bounded.

Proof. The desired trajectory ϕ_d together with its time-derivatives typically satisfies the boundedness assumption (for example by employing the standard expression $\phi_{d_i}(t) = A \sin(\omega t + (i-1)\delta)$ for constant A , ω and δ).

The exponential convergence to zero of the joint tracking error $e_\phi(t)$ is seen by inserting (6.60) into (6.59) and using (6.61). Thus, we end up with a linear system for the error dynamics:

$$\ddot{e}_\phi(t) + \mathbf{K}_D \dot{e}_\phi(t) + \mathbf{K}_P e_\phi(t) = 0. \quad (6.62)$$

Hence, the origin of (6.62) is exponentially stable and the error $e_\phi(t)$ converges to zero exponentially. This can easily be shown since (6.62) is a system of $n-3$ decoupled linear second order systems of the form

$$\ddot{e}_{\phi_j} + K_{D_j} \dot{e}_{\phi_j} + K_{P_j} e_{\phi_j} = 0, \quad j = 1, \dots, n-3, \quad (6.63)$$

where we have defined $e_\phi = [e_{\phi_1}, e_{\phi_2}, \dots, e_{\phi_{n-3}}]^\top$ and K_{D_j} and K_{P_j} are the j -th components along the diagonals of \mathbf{K}_D and \mathbf{K}_P , respectively. Each of these systems is globally exponentially stable, because all K_{D_j} and K_{P_j} are positive. Hence, the overall system (6.62) is globally exponentially stable. Notice from (6.61) that

$$\phi(t) = e_\phi(t) + \phi_d(t). \quad (6.64)$$

Hence, $\phi(t)$ is bounded since both $e_\phi(t)$ and $\phi_d(t)$ are bounded. The same applies to the first and second derivative of $\phi(t)$. It now remains to show that the velocity vector \mathbf{w} remains bounded¹. In other words, the so-called tracking dynamics (6.50) produces bounded solutions for time-varying inputs $\phi(t)$, $\dot{\phi}(t)$ and $\ddot{\phi}(t)$ with $\phi(t)$ satisfying (6.64). To this end, we note that the expression for $\dot{\mathbf{w}}$ in (6.50) is locally Lipschitz and employ Theorem 4.18 in Khalil (2002) (see Appendix A.1 for the theorem) in the following. Hence, we define the continuously differentiable function

$$V(t, \mathbf{w}) = \frac{1}{2} \mathbf{w}^\top \mathbf{M}_{11}(\phi(t)) \mathbf{w}. \quad (6.65)$$

Thus,

$$\frac{1}{2} \lambda_{M, \min} \|\mathbf{w}\|^2 \leq V(t, \mathbf{w}) \leq \frac{1}{2} \lambda_{M, \max} \|\mathbf{w}\|^2 \quad (6.66)$$

where $\|\cdot\|$ denotes the Euclidean norm, $\lambda_{M, \min}$ is the infimum over all t of the smallest eigenvalue of $\mathbf{M}_{11}(\phi(t))$ and $\lambda_{M, \max}$ is the supremum over all

¹For notational simplicity, we omit the dependence on t and $\phi(t)$ in the following when it is appropriate.

t of the largest eigenvalue of $\mathbf{M}_{11}(\phi(t))$. Since $\mathbf{M}_{11}(\phi)$ is continuous and positive definite for all ϕ and since $\phi(t)$ belongs to a compact set for all t , then such $\lambda_{M,max} \geq \lambda_{M,min} > 0$ do exist. What remains is to find a continuous positive definite function $W(\mathbf{w})$ such that

$$\dot{V}(t, \mathbf{w}) \leq -W(\mathbf{w}), \quad \forall \|\mathbf{w}\| \geq \mu > 0, \quad \forall t \geq 0, \quad (6.67)$$

for some $\mu > 0$ and all $\mathbf{w} \in \mathbb{R}^3$. This will be the task for the remainder of this section.

Consider

$$\dot{V} = \mathbf{w}^T \mathbf{M}_{11} \dot{\mathbf{w}} + \frac{1}{2} \mathbf{w}^T \dot{\mathbf{M}}_{11} \mathbf{w}. \quad (6.68)$$

Inserting for $\mathbf{M}_{11} \dot{\mathbf{w}}$ from (6.50), we find that

$$\dot{V} = \mathbf{w}^T \left(\mathbf{g}_1 - \mathbf{M}_{12} \ddot{\phi} - \mathbf{h}_1 \right) + \frac{1}{2} \mathbf{w}^T \dot{\mathbf{M}}_{11} \mathbf{w}. \quad (6.69)$$

The velocities \mathbf{w} need to be extracted from \mathbf{g}_1 in (6.48) and \mathbf{h}_1 in (6.46) to satisfy the above inequality (6.67). To this end, we rewrite $\boldsymbol{\nu}$ in (6.30) as

$$\boldsymbol{\nu} = \mathbf{P}_w \mathbf{w} + \mathbf{P}_\phi \dot{\phi}. \quad (6.70)$$

where \mathbf{P}_w is found in (6.42) and \mathbf{P}_ϕ is found in (6.18).

We start by investigating \mathbf{h}_1 which is the term in \dot{V} that needs to be elaborated the most to prove the inequality (6.67). By inserting the identity (6.70) into the expression (6.46) for \mathbf{h}_1 , we find that

$$\mathbf{h}_1 = \mathbf{P}_w^T \mathbf{C} \left(\dot{\theta}, \phi, \dot{\phi} \right) \mathbf{P}_w \mathbf{w} + \mathbf{P}_w^T \mathbf{C} \left(\dot{\theta}, \phi, \dot{\phi} \right) \mathbf{P}_\phi \dot{\phi}. \quad (6.71)$$

It will later be shown that the term $\mathbf{P}_w^T \mathbf{C} \left(\dot{\theta}, \phi, \dot{\phi} \right) \mathbf{P}_w \mathbf{w}$ is cancelled from \dot{V} because of skew-symmetry, however $\dot{\theta}$ and thus an expression for \mathbf{w} , needs to be extracted from the second term in \mathbf{h}_1 and this will be elaborated on in the following. By using $\mathbf{T}_q \mathbf{P}_\phi = \mathbf{0}$ ($\dot{\mathbf{I}}_{n-2 \times n-2} = 0$) together with the expression for $\mathbf{C} \left(\dot{\theta}, \phi, \dot{\phi} \right)$ in (6.39), we write a part of the right-most expression in (6.71) as

$$\mathbf{P}_w^T \mathbf{C} \mathbf{P}_\phi = \mathbf{P}_w^T \mathbf{T}_q^T \mathbf{C}_P \mathbf{P}_\phi, \quad (6.72)$$

where we also have used that $\mathbf{T}_q \mathbf{P}_\phi = \mathbf{P}_\phi$. Define \mathbf{c}_j to be the j -th column of \mathbf{C}_P and

$$\mathbf{p}_j^T = \left[\mathbf{0}_{1 \times j-1} \quad 1 \quad \mathbf{0}_{1 \times n-j} \right] \in \mathbb{R}^n \quad (6.73)$$

is a row vector of zeros with 1 on its j -th column. From the expression for the elements of \mathbf{C}_P in (6.16) we find that

$$\mathbf{c}_j = \mathbf{D}_{P_j} \dot{\mathbf{q}}, \quad (6.74)$$

where

$$\mathbf{D}_{P_j} = \{c_{ijk}\} = \begin{bmatrix} c_{1j1} & c_{1j2} & \cdots & c_{1jn} \\ c_{2j1} & c_{2j2} & & \vdots \\ \vdots & & \ddots & \vdots \\ c_{nj1} & \cdots & \cdots & c_{njn} \end{bmatrix}, \quad (6.75)$$

and c_{ijk} is given by (6.17). Hence, from (6.74), together with the previous expression (6.16) for \mathbf{C}_P , we see that

$$\mathbf{C}_P = \sum_{j=1}^n \mathbf{c}_j \mathbf{p}_j^T = \sum_{j=1}^n \mathbf{D}_{P_j} \dot{\mathbf{q}} \mathbf{p}_j^T. \quad (6.76)$$

By calculating the partial derivatives of the elements of \mathbf{M}_P in (6.17) we find from the expression for the elements of \mathbf{D}_{P_j} that its first two columns are zero. Hence, by comparing $\dot{\mathbf{q}}$ in (6.3) with $\boldsymbol{\nu}$ in (6.30) we obtain

$$\mathbf{D}_{P_j} \dot{\mathbf{q}} = \mathbf{D}_{P_j} \boldsymbol{\nu}. \quad (6.77)$$

We can now continue to expand \mathbf{C}_P by inserting the identities (6.70) and (6.77) into (6.76):

$$\mathbf{C}_P = \sum_{j=1}^n \mathbf{D}_{P_j} \boldsymbol{\nu} \mathbf{p}_j^T = \sum_{j=1}^n \mathbf{D}_{P_j} \left(\mathbf{P}_w \mathbf{w} + \mathbf{P}_\phi \dot{\phi} \right) \mathbf{p}_j^T. \quad (6.78)$$

By inserting the new expression for \mathbf{C}_P (6.78) into (6.72) and noticing that $\mathbf{p}_j^T \mathbf{P}_\phi = \mathbf{0}_{1 \times n-3}$ for $j = 1, \dots, 3$ we obtain

$$\mathbf{P}_w^T \mathbf{C}_P \mathbf{P}_\phi = \mathbf{P}_w^T \mathbf{T}_q^T \sum_{j=4}^n \left\{ \mathbf{D}_{P_j} \mathbf{P}_w \mathbf{w} + \mathbf{D}_{P_j} \mathbf{P}_\phi \dot{\phi} \right\} \mathbf{p}_j^T \mathbf{P}_\phi. \quad (6.79)$$

Inspecting the terms with \mathbf{w} in the summation (6.79) for $j = 4, \dots, n$ we find that

$$\mathbf{p}_{\phi_{j-3}}^T = \mathbf{p}_j^T \mathbf{P}_\phi, \quad (6.80)$$

where

$$\mathbf{p}_{\phi_{j-3}}^T = [\mathbf{0}_{1 \times j-3-1} \quad 1 \quad \mathbf{0}_{1 \times n-j}] \in \mathbb{R}^{n-3} \quad (6.81)$$

is a row vector of zeros with 1 on its $j - 3$ -column, and we note that

$$\dot{\phi}_k = \mathbf{p}_{\phi_k}^T \dot{\phi}, \quad (6.82)$$

where ϕ_k is the k -th element of $\phi \in \mathbb{R}^{n-3}$. In addition, let

$$\mathbf{D}_{B_j}(\phi) = \mathbf{T}_q^T \mathbf{D}_{P_j} \in \mathbb{R}^{n \times n}, \quad j = 4, \dots, n, \quad (6.83)$$

to explicitly show that we are dealing with a matrix with elements only dependent on ϕ . By inserting (6.80) into (6.79), and employing the identities (6.82) and (6.83), we now find the second term of \mathbf{h}_1 in (6.71) as

$$\mathbf{P}_w^T \mathbf{C} \mathbf{P}_\phi \dot{\phi} = \mathbf{P}_w^T \sum_{j=4}^n \mathbf{D}_{B_j} \left(\mathbf{P}_\phi \dot{\phi} \dot{\phi}_{j-3} + \mathbf{P}_w \dot{\phi}_{j-3} \mathbf{w} \right) \quad (6.84)$$

and we have managed to extract \mathbf{w} .

We are now ready to insert \mathbf{h}_1 into \dot{V} in (6.69):

$$\begin{aligned} \dot{V} = & \mathbf{w}^T \mathbf{g}_1 + \mathbf{w}^T (-\mathbf{M}_{12}) \ddot{\phi} + \frac{1}{2} \mathbf{w}^T \dot{\mathbf{M}}_{11} \mathbf{w} + \\ & \mathbf{w}^T \mathbf{P}_w^T (-\mathbf{C}) \mathbf{P}_w \mathbf{w} + \\ & \mathbf{w}^T \mathbf{P}_w^T \sum_{j=4}^n (-\mathbf{D}_{B_j}) \mathbf{P}_\phi \dot{\phi} \dot{\phi}_{j-3} + \\ & \mathbf{w}^T \mathbf{P}_w^T \sum_{j=4}^n (-\mathbf{D}_{B_j}) \mathbf{P}_w \dot{\phi}_{j-3} \mathbf{w}. \end{aligned} \quad (6.85)$$

Define

$$\mathbf{k}_1(\phi, \ddot{\phi}) = (-\mathbf{M}_{12}) \ddot{\phi}, \quad (6.86)$$

$$\mathbf{k}_2(\phi, \dot{\phi}) = \mathbf{P}_w^T \sum_{j=4}^n (-\mathbf{D}_{B_j}) \mathbf{P}_\phi \dot{\phi} \dot{\phi}_{j-3}, \quad (6.87)$$

and note that the following inequality holds:

$$\mathbf{w}^T \mathbf{k}_i \leq \|\mathbf{w}\| \|\mathbf{k}_i\|. \quad (6.88)$$

We also have that

$$\mathbf{w}^T \mathbf{P}_w^T (-\mathbf{D}_{B_j}(\phi(t))) \mathbf{P}_w \dot{\phi}_{j-3}(t) \mathbf{w} \leq \lambda_{D_{B_j}, \max}(\phi(t)) \left| \dot{\phi}_{j-3}(t) \right| \|\mathbf{w}\|^2, \quad (6.89)$$

where $\lambda_{D_{B_j}, \max}(\phi)$ is the largest eigenvalue of $\frac{1}{2} \left(\mathbf{P}_w^T (-\mathbf{D}_{B_j}(\phi)) \mathbf{P}_w + (\mathbf{P}_w^T (-\mathbf{D}_{B_j}(\phi)) \mathbf{P}_w)^T \right)$. Since ϕ only enters $\mathbf{D}_{B_j}(\phi)$ through $\cos(\phi_j)$ and $\sin(\phi_j)$, then there exists a constant $\bar{\lambda}_{D_{B_j}, \max}$ such that

$$\lambda_{D_{B_j}, \max}(\phi) \leq \bar{\lambda}_{D_{B_j}, \max} \quad \forall \phi. \quad (6.90)$$

From (6.90) we define the constant

$$\bar{\lambda}_D = \sum_{j=4}^n \bar{\lambda}_{D_{B_j, max}}. \quad (6.91)$$

We now have from (6.89)-(6.91) that the latter expression in (6.85) satisfies

$$\begin{aligned} \mathbf{w}^T \mathbf{P}_w^T \sum_{j=4}^n (-\mathbf{D}_{B_j}) \mathbf{P}_w \dot{\phi}_{j-3} \mathbf{w} &\leq \sum_{j=4}^n \lambda_{D_{B_j, max}} (\phi) \left| \dot{\phi}_{j-3}(t) \right| \|\mathbf{w}\|^2 \\ &\leq \bar{\lambda}_D \|\mathbf{w}\|^2 \sup_{t \geq 0} \max_{j=4, \dots, n} \left| \dot{\phi}_{j-3}(t) \right| \\ &\leq \bar{\lambda}_D \|\mathbf{w}\|^2 \sup_{t \geq 0} \left\| \dot{\phi}(t) \right\|, \end{aligned} \quad (6.92)$$

where we have employed (see, e.g., Khalil, 2002)

$$\max_{j=4, \dots, n} \left| \dot{\phi}_{j-3}(t) \right| = \left\| \dot{\phi}(t) \right\|_{\infty}, \quad \left\| \dot{\phi}(t) \right\|_{\infty} \leq \left\| \dot{\phi}(t) \right\|_2. \quad (6.93)$$

By using the inequality (6.92) together with (6.86)-(6.89), we can simplify inequality (6.85) as

$$\begin{aligned} \dot{V} \leq & \mathbf{w}^T \mathbf{g}_1 + \frac{1}{2} \mathbf{w}^T \left(\dot{\mathbf{M}}_{11} - 2\mathbf{P}_w^T \mathbf{C} \mathbf{P}_w \right) \mathbf{w} + \\ & \|\mathbf{w}\| \|\mathbf{k}_1\| + \|\mathbf{w}\| \|\mathbf{k}_2\| + \bar{\lambda}_D \|\mathbf{w}\|^2 \sup_{t \geq 0} \left\| \dot{\phi}(t) \right\|. \end{aligned} \quad (6.94)$$

The above expression (6.94) can be further simplified since

$$\frac{1}{2} \mathbf{w}^T \left(\dot{\mathbf{M}}_{11} - 2\mathbf{P}_w^T \mathbf{C} \mathbf{P}_w \right) \mathbf{w} = \frac{1}{2} \mathbf{w}^T \mathbf{P}_w^T \left(\dot{\mathbf{M}} - 2\mathbf{C} \right) \mathbf{P}_w \mathbf{w} = 0, \quad (6.95)$$

where we have used that $\dot{\mathbf{M}}_{11} = \mathbf{P}_w^T \dot{\mathbf{M}} \mathbf{P}_w$ and that $(\dot{\mathbf{M}} - 2\mathbf{C})$ is skew-symmetric. The latter property is proved in, e.g., Spong and Vidyasagar (1989).

The remaining term $\mathbf{w}^T \mathbf{g}_1$ in \dot{V} arises from friction, see (6.48), and will now be investigated. We start by partitioning the complete friction term \mathbf{g} in (6.44) as

$$\mathbf{g} = \mathbf{g}_V + \mathbf{g}_C \implies \mathbf{g}_1 = \mathbf{g}_{V_1} + \mathbf{g}_{C_1} \in \mathbb{R}^3, \quad (6.96)$$

where

$$\mathbf{g}_{V_1} = \mathbf{P}_w^T \mathbf{g}_V, \quad \mathbf{g}_{C_1} = \mathbf{P}_w^T \mathbf{g}_C, \quad (6.97)$$

and the viscous friction \mathbf{g}_V and Coulomb friction \mathbf{g}_C are

$$\mathbf{g}_V = \sum_{i=3}^n \left((\mathbf{J}_{v_{F_i}})^T {}_{B_3} \mathbf{F}_{F_i}^V + (\mathbf{J}_{v_{R_i}})^T {}_{B_3} \mathbf{F}_{R_i}^V \right), \quad (6.98)$$

$$\mathbf{g}_C = \sum_{i=3}^n \left((\mathbf{J}_{v_{F_i}})^T {}_{B_3} \mathbf{F}_{F_i}^C + (\mathbf{J}_{v_{R_i}})^T {}_{B_3} \mathbf{F}_{R_i}^C \right). \quad (6.99)$$

By inserting the constitutive law (6.26) for the viscous friction force and employing the identities ${}_{B_3} \mathbf{F}_{\xi_i}^V = {}_{B_3} \mathbf{F}_{\xi_i}^{P_V}$ and ${}_{B_3} \mathbf{F}_{\xi_i}^V = \mathbf{R}_{B_i}^{B_3} {}_{B_i} \mathbf{F}_{\xi_i}^V$ for $\xi_i = F_i, R_i$, we find that

$$\mathbf{g}_V = - \sum_{i=3}^n \sum_{\xi=R,F} \frac{F_{N_i}}{2} \left((\mathbf{J}_{v_{\xi_i}})^T \mathbf{R}_{B_i}^{B_3} \mathbf{D}_{\mu_V} {}_{B_i} \mathbf{v}_{\xi_i} \right). \quad (6.100)$$

Using ϕ and $\boldsymbol{\nu}$ in expression (6.31) for ${}_{B_i} \mathbf{v}_{\xi_i}$ and inserting the result into \mathbf{g}_V in (6.98), we get

$$\mathbf{g}_V = - \sum_{i=3}^n \sum_{\xi=R,F} \underbrace{\left((\mathbf{J}_{v_{\xi_i}})^T \mathbf{R}_{B_i}^{B_3} \frac{F_{N_i}}{2} \mathbf{D}_{\mu_V} \left(\mathbf{R}_{B_i}^{B_3} \right)^T \mathbf{J}_{v_{\xi_i}} \right)}_{\mathbf{M}_{V_i}(\phi)} \boldsymbol{\nu}. \quad (6.101)$$

We now find the pleasant fact that

$$\mathbf{M}_V(\phi) = \sum_{i=3}^n \mathbf{M}_{V_i}(\phi) \quad (6.102)$$

is positive definite as shown in Appendix A.2. Intuitively, this is the case since \mathbf{M}_V is the analogy to a mass matrix of interconnected point masses at any time-instant and such a mass matrix is positive definite. We formulate \mathbf{g}_{V_1} defined in (6.97) by using the expression for $\boldsymbol{\nu}$ in (6.70) and inserting \mathbf{M}_V (6.102) into (6.101), then

$$\mathbf{g}_{V_1} = -\mathbf{P}_w^T \mathbf{M}_V \mathbf{P}_w \mathbf{w} - \mathbf{P}_w^T \mathbf{M}_V \mathbf{P}_\phi \dot{\phi}. \quad (6.103)$$

We now move on to the Coulomb friction term \mathbf{g}_C since we have managed to extract \mathbf{w} from \mathbf{g}_{V_1} . The Coulomb friction force $\mathbf{F}_{\xi_i}^C$ is found as the smooth approximation of the Coulomb friction force in (6.21) on page 115 and by inserting this into the expression for \mathbf{g}_C in (6.99), we obtain

$$\mathbf{g}_C = - \sum_{i=3}^n \frac{1}{\pi} \sum_{\xi_i=F_i, R_i} \left((\mathbf{J}_{v_{\xi_i}})^T F_{N_i} \mathbf{R}_{B_i}^{B_3} \mathbf{D}_{\mu_C} \arctan \left(\frac{{}_{B_i} \mathbf{v}_{\xi_i}}{\epsilon_C} \right) \right). \quad (6.104)$$

By employing the expressions above that constitute \mathbf{g}_1 , we find from (6.94) that the function \dot{V} can now be written

$$\begin{aligned} \dot{V} \leq & -\mathbf{w}^T \mathbf{P}_w^T \mathbf{M}_V \mathbf{P}_w \mathbf{w} + \mathbf{w}^T \left(-\mathbf{P}_w^T \mathbf{M}_V \mathbf{P}_\phi \dot{\phi} \right) + \\ & \mathbf{w}^T \mathbf{g}_{C_1} + \|\mathbf{w}\| (\|\mathbf{k}_1\| + \|\mathbf{k}_2\|) + \bar{\lambda}_D \|\mathbf{w}\|^2 \sup_{t \geq 0} \left\| \dot{\phi}(t) \right\|. \end{aligned} \quad (6.105)$$

Define the vectors

$$\mathbf{k}_3 = -\mathbf{P}_w^T \mathbf{M}_V \mathbf{P}_\phi \dot{\phi}, \quad (6.106)$$

$$\mathbf{k}_4 = \mathbf{g}_{C_1}(\phi, B_i \mathbf{v}_{\xi_i}). \quad (6.107)$$

In addition, define the positive definite matrix

$$\bar{\mathbf{M}}_V = \mathbf{P}_w^T \mathbf{M}_V \mathbf{P}_w \in \mathbb{R}^{3 \times 3}. \quad (6.108)$$

The matrix $\bar{\mathbf{M}}_V$ is positive definite since it is the upper left submatrix of the positive definite matrix \mathbf{M}_V . By inserting the identities (6.106)-(6.108) into (6.105), we obtain

$$\begin{aligned} \dot{V} \leq & -\mathbf{w}^T \bar{\mathbf{M}}_V \mathbf{w} + \bar{\lambda}_D \|\mathbf{w}\|^2 \sup_{t \geq 0} \left\| \dot{\phi}(t) \right\| + \\ & \|\mathbf{w}\| (\|\mathbf{k}_1\| + \|\mathbf{k}_2\| + \|\mathbf{k}_3\| + \|\mathbf{k}_4\|). \end{aligned} \quad (6.109)$$

Recall that due to (6.64) $\dot{\phi}(t)$ and $\ddot{\phi}(t)$ are bounded, i.e. there exists a constant \tilde{c}_ϕ such that $\left\| \dot{\phi}(t) \right\| \leq \tilde{c}_\phi$ and $\left\| \ddot{\phi}(t) \right\| \leq \tilde{c}_\phi$ for all $t \geq 0$. For the expressions for \mathbf{k}_1 , \mathbf{k}_2 , \mathbf{k}_3 and \mathbf{k}_4 in (6.86), (6.87), (6.106) and (6.107), this implies that there exists a constant $\bar{k}_5 > 0$ such that

$$\|\mathbf{k}_1\| + \|\mathbf{k}_2\| + \|\mathbf{k}_3\| + \|\mathbf{k}_4\| \leq \bar{k}_5, \quad \forall t \geq 0. \quad (6.110)$$

Since ϕ enters $\bar{\mathbf{M}}_V(\phi)$ only through $\cos(\phi_j)$ and $\sin(\phi_j)$ and $\bar{\mathbf{M}}_V(\phi)$ is continuous and positive definite, then there exists a constant $\bar{\lambda}_{M_V, \min} > 0$ such that

$$\bar{\lambda}_{M_V, \min} \|\mathbf{w}\|^2 \leq \mathbf{w}^T \bar{\mathbf{M}}_V \mathbf{w}, \quad \forall \phi. \quad (6.111)$$

Hence,

$$\dot{V} \leq - \left(\bar{\lambda}_{M_V, \min} - \bar{\lambda}_D \sup_{t \geq 0} \left\| \dot{\phi}(t) \right\| \right) \|\mathbf{w}\|^2 + \|\mathbf{w}\| \bar{k}_5. \quad (6.112)$$

Since $\left\| \dot{\phi}(t) \right\|$ is bounded, the inequality (6.112) implies that V does not grow faster than an exponent, i.e. solutions $\mathbf{w}(t)$ of the system (6.50) are

defined for all $t \geq 0$. Moreover, since $\dot{\phi}(t) - \dot{\phi}_d(t) \rightarrow 0$ as $t \rightarrow 0$ there exists a time instant T_* such that $\|\dot{\phi}(t)\| \leq 2c_{\dot{\phi}}$ where $c_{\dot{\phi}}$ is the upper bound of $\|\dot{\phi}_d(t)\|$ from the formulation in Theorem 6.1. Choose this $c_{\dot{\phi}} > 0$ such that $\bar{\lambda}_{M_V, min} - \bar{\lambda}_D 2c_{\dot{\phi}} > 0$ and define the constant

$$\bar{k}_6 = \bar{\lambda}_{M_V, min} - \bar{\lambda}_D 2c_{\dot{\phi}} > 0. \quad (6.113)$$

Then, for $t \geq T_*$ we have

$$\dot{V} \leq -\bar{k}_6 \|\mathbf{w}\|^2 + \bar{k}_5 \|\mathbf{w}\|. \quad (6.114)$$

We are now close to satisfying the inequality (6.67) which we stated in the beginning of this section. Define a constant η such that $0 < \eta < 1$. We can then rewrite (6.114) as

$$\dot{V} \leq -(1 - \eta)\bar{k}_6 \|\mathbf{w}\|^2 - \eta\bar{k}_6 \|\mathbf{w}\|^2 + \bar{k}_5 \|\mathbf{w}\|.$$

Hence, the positive definite function $W(\mathbf{w})$ in (6.67) is

$$W(\mathbf{w}) = (1 - \eta)\bar{k}_6 \|\mathbf{w}\|^2, \quad (6.115)$$

and we obtain

$$\dot{V} \leq -W(\mathbf{w}), \quad (6.116)$$

for

$$\eta\bar{k}_6 \|\mathbf{w}\|^2 \geq \bar{k}_5 \|\mathbf{w}\|, \quad (6.117)$$

The positive constant μ in (6.67) needs to be found such that (6.116) is satisfied. From (6.117) we find that

$$\|\mathbf{w}\| \geq \frac{\bar{k}_5}{\eta\bar{k}_6} = \mu. \quad (6.118)$$

Hence, we have that

$$\dot{V} \leq -W(\mathbf{w}), \quad \|\mathbf{w}\| \geq \mu > 0, \quad (6.119)$$

for $t \geq T_*$. This implies that for $t \geq T_*$ the solutions $\mathbf{w}(t)$ are bounded. Moreover, (6.112) ensures that solutions $\mathbf{w}(t)$ of the system (6.50) are defined for all $t \geq 0$, i.e. solutions can not escape to infinity in the time-interval from 0 to T_* . These two latter results together with (6.66) prove that \mathbf{w} is bounded. This concludes the proof of Theorem 6.1. ■

6.4 Summary

This chapter considers the problem of model-based control of the joints of a snake robot without wheels. The potential range of applications of snake robots is wide and delicate operations such as inspection and maintenance in industrial environments or performing search and rescue operations require precise control of snake robot joints. To this end, we present a controller that asymptotically stabilizes the joints of the snake robot to a desired reference trajectory. We have developed a planar model of the snake robot based on the standard dynamics of a robot manipulator. This model is denoted the process plant model. Such a formulation is advantageous for control since it allows us to employ the large variety of control techniques available for robot manipulators. Our controller is based on input-output linearization of a control plant model, which is a reformulation of the process plant model, of the snake robot dynamics also developed in this chapter. The reformulation of the process plant model is necessary to make the stability analysis more manageable and exponential stability of the desired trajectories for the joints is shown with a formal Lyapunov-based stability proof. The performance of the controller will be tested in simulation on both a smooth and non-smooth planar snake robot model in Chapter 7.

Chapter 7

Simulation and Experimental Results

7.1 Introduction

The ability to investigate a system by simulation, as opposed to testing the actual system, is a powerful tool for testing and evaluating the system as long as the model represents the actual system dynamics with sufficiently high accuracy. In general, simulations allow one to evaluate the system without risking damaging it or wearing it out. In addition, it is easier to try out various initial conditions since the actual system does not need to be manipulated for each trial.

In this chapter we present simulations of the various snake robot models presented throughout this thesis. The algorithms for the time-stepping method presented for the 2D and 3D model have been implemented for numerical treatment of the models and we investigate whether the time-stepping method is suitable for simulating the models. We have qualitative knowledge from previously published papers on snake robots of how a snake robot should perform during various motion patterns. In this chapter we will test a selection of these gaits to see whether the snake robot moves as expected with our models. Moreover, to further validate the model, experimental results will be presented for the NTNU/SINTEF snake robot ‘Aiko’ and these results will be compared to the simulation results from both the 2D model presented in Chapter 3 and the 3D model presented in Chapter 4. We will also investigate how the 2D and 3D model performs for obstacle aided locomotion with small obstacles. Such small obstacles may, in general, cause problems due to numerical issues and that the obstacles

may get stuck in the snake robot due to the design of the exterior of the snake robot. In addition, the 3D model will be tested for a motion pattern based on rolling to motivate the inclusion of the exoskeleton defined for contact with the ground surface. Moreover, the input-output linearizing controller presented in Chapter 6 will be evaluated for an idealized snake robot model and for a model that includes various degrees of noise. These results will be compared to an ordinary PD-controller. Both of these latter models are based on the ‘robot model’ developed in Chapter 6.

This chapter includes the results from Transeth *et al.* (2007a), Transeth *et al.* (2008b) and Transeth *et al.* (2008a) and it is organized as follows. A snake robot called ‘Aiko’ – the basis for the models – is presented in Section 7.2 together with all simulation parameters. Simulation results are given in Section 7.3. The experimental setup is presented in Section 7.4 and simulations with back-to-back comparisons to experimental results are given in Section 7.5. A summary of the most important findings in this chapter is given in Section 7.6.

7.2 Simulation Parameters

The snake robot ‘Aiko’ is used as a basis for the mathematical models. In this section we will describe the most important characteristics of the snake robot and use these to determine the parameters for the models presented in Chapters 3, 4 and 6. In addition, we will present the simulation parameters.

7.2.1 A Description of Aiko and Model Parameters

Aiko is depicted in Figure 7.1 and has eleven (red) links connected by ten two DOF joints. As seen from the figure, the snake robot has cylindrical links and a half-sphere made of metal as a ‘head’ which may be removed. Aiko weighs 7.5 kg and each link is 12.2 cm long. The snake robot was developed¹ at the NTNU/SINTEF Advanced Robotics Laboratory in Trondheim, Norway in the year 2006.

¹General requirements for Aiko were formulated by Pål Liljebäck and Aksel Andreas Transeth. These requirements included that the snake robot should have 2 DOF joints, be able to lift three or more of its links and have an as smooth as possible exoskeleton to be able to use obstacles to move forward and/or move over them. Moreover, the exoskeleton should be cylindrical to be able to easily move by e.g. lateral rolling. Sverre Brovoll and Kristoffer Nyborg Gregertsen developed Aiko based on these requirements during S. Brovoll’s Master’s degree and K. N. Gregertsen’s summer job at SINTEF under the supervision of P. Liljebäck, A. A. Transeth and Amund Skavhaug. A detailed description of the design and components of Aiko is found in Brovoll (2006).



Figure 7.1: The NTNU/SINTEF snake robot ‘Aiko’.

As mentioned above, we employ Aiko as a basis for choosing the model parameters. These parameters are given in Table 7.1 for the 2D model from Chapter 3, the 3D model from Chapter 4 and the model from Chapter 6. The latter will be referred to as the ‘robot model’ since it is based on the standard robot manipulator model form. The moment of inertia for a link was calculated by assuming that each link was a cylinder of uniform mass with height L_i and radius L_{SC_i} . The coefficient of friction between the snake robot and the ground was calculated from the measurements obtained by connecting Aiko to a scale and then pulling the scale along one of the particle boards Aiko is moving on during the experiments.

7.2.2 Simulation Parameters and Reference Joint Angles

The numerical integration of the various models requires first deciding the simulation parameters, such as step-length and different tolerances. Most of the remaining parameters needed to perform numerical integration of the models presented in Chapters 3, 4 and 6 are given in Table 7.2. Some parameters are varied in the upcoming presentation of the simulations to test how the snake robot behaves. The new values will then be clearly stated.

We employ the same expressions to find the reference angles for the joints in this chapter as given in Chapter 2. However, we repeat the expressions here to ease the readability of the upcoming sections. The reference angle for joint i for the sideways (yaw) $\phi_{h_i,d}$ and up-down (pitch) $\phi_{v_i,d}$ motion of each joint are given by

$$\phi_{h_i,d}(t) = A_h \sin(\omega_h t + (i-1)\beta_h) \quad (7.1)$$

$$\phi_{v_i,d}(t) = A_v \sin(\omega_v t + (i-1)\beta_v + \beta_0), \quad (7.2)$$

Parameter	Value	Description
L_i	0.122 m	link length
L_{GS_i}	0.0393 m	length employed to lump the contact point between the link into two points for the 3D model and the ‘robot model’
L_{SC_i}	0.0525 m	link radius
m_i	$\frac{7.5}{11} \approx 0.682$ kg	link mass
Θ_{ψ_i}	1.32×10^{-3} kg m ²	moment of inertia around transversal axis through CG of link
Θ_{ρ_i}	9.40×10^{-4} kg m ²	moment of inertia around longitudinal axis through CG of link
g	9.81 m/s ²	acceleration of gravity
μ_{T_x}	0.2	Coulomb friction coefficient longitudinal to a link
μ_{T_y}	0.2	Coulomb friction coefficient lateral to a link

Table 7.1: Parameter values for the various mathematical models.

where the various components are further described in Section 2.4. The parameter γ_h , which controls the direction of forward motion, is removed from (7.1) since it will always be set to zero in the following simulations and experiments.

Remark 7.1 (Choice of motion pattern parameters) *In the following sections each motion pattern will most often be simulated with a single set of motion pattern parameters. In general, the specific choice of parameters for a motion pattern is dependent on the design of the snake robot (for example link length and maximum joint angle), and the desired behaviour of the snake robot (e.g. desired speed or turning ratio). For example, the snake robot needs at least three contact points between its body and the obstacles for obstacle aided locomotion with lateral undulation shown in Section 7.3.4. Hence, the shape of the snake robot must have ‘enough curves’ so that locomotion is possible. The curvature of the snake robot in the horizontal plane is controlled with the parameters A_h and β_h .*

A ‘soft start’ approach is employed for simulations and experiments where the initial orientation of each joint is not equal to the desired angles

Parameter	2D model	3D model	Robot model
n	11 links	11 links	13 DOF
r_N	\emptyset	0.1	\emptyset
r_T	1.3	0.01	0.2
r_V	\emptyset	0.05	\emptyset
r_H	0.01	0.01	\emptyset
Δt	2.5×10^{-4} s	2.5×10^{-4} s	2.5×10^{-4} s
μ_V	\emptyset	0.02	\emptyset
(μ_{C_x}, μ_{C_y})	\emptyset	\emptyset	(0.2, 0.5)
(μ_{V_x}, μ_{V_y})	\emptyset	\emptyset	(0.1, 0.2)
ϵ_C	\emptyset	\emptyset	0.001
k_P	800	800	200
k_D	2	2	1.2
K_{P_j}	\emptyset	\emptyset	4.0×10^5
K_{D_j}	\emptyset	\emptyset	2.4×10^3
ϵ	2×10^{-5}	2×10^{-5}	1×10^{-5}

Table 7.2: Simulation parameters for the 2D model in Chapter 3, the 3D model in Chapter 4, and the model based on minimal coordinates (the ‘robot model’) presented in Chapter 6. K_{P_j} and K_{D_j} are the elements on the diagonal of the matrices \mathbf{K}_P and \mathbf{K}_D in (6.60), respectively, and k_P , k_D are the gains of the PD-controllers. See Algorithm 3.1 on page 64 and Algorithm 4.2 on page 94 for the usage of the parameters r_N , r_T , r_V and r_H . The step size in the numerical integration is given by Δt . The various friction coefficients are given by μ_V , μ_{C_x} , μ_{C_y} , μ_{V_x} and μ_{V_y} . The tolerance for the Modified Newton Algorithm is given by ϵ . The steepness of the approximation to the sign-function is given by ϵ_C .

$\phi_{h_i,d}(t)$ and $\phi_{v_i,d}(t)$ at the time the snake robot is started. An example of such a situation is if we want the snake robot to move forward by lateral undulation, but it has a straight posture at $t = 0$. The ‘soft start’ approach is utilized to avoid large steps in the reference signal which causes large actuator torques. To this end, we consider all the reference signals $\phi_{h_i,d}(t)$ and $\phi_{v_i,d}(t)$ separately and calculate them from the expressions (7.1) and (7.2) for the reference angles, respectively. However, we set $\phi_{h_i,d}(t) = 0$ until $|\phi_{h_i,d}(t)| \leq 3\pi/180$ and $\phi_{v_i,d}(t) = 0$ until $|\phi_{v_i,d}(t)| \leq 3\pi/180$ for the first time after start-up. After the reference value is within the allowed start value, the reference angle is found from (7.1) or (7.2).

7.3 Simulations without Experimental Validation

In this section we present simulations of the models presented in this thesis. First, we simulate the 3D model where the snake robot is raised above the ground surface and then dropped onto it. This scenario challenges the numerical stability of the numerical treatment since large sudden contact forces arise simultaneously. Moreover, we check whether the snake robot moves forward by lateral undulation with anisotropic friction, as it is expected to. Second, we test if the defined exoskeleton works well for U-shaped lateral rolling and sidewinding, and we investigate what happens if the friction between the snake robot and the ground surface is changed for these motion patterns. Third, obstacle aided locomotion is simulated with small obstacles. Such small obstacles pose difficulties both in the numerical treatment and that the snake robot may get stuck. We investigate how our 2D and 3D models perform for these challenges and we also enquire into how the snake robot behaves if one obstacle that is necessary for forward motion is moved. Moreover, contact forces between the snake robot and the obstacles are plotted and compared to the discussion on obstacle aided locomotion given in Chapter 5. Finally, the input-output linearizing controller is compared to a conventional PD-controller, both implemented on the ‘robot model’. In addition to the comparison, we will also inquire into whether the ‘robot model’ is suitable for numerical treatment. Note that for simulation of the 2D model presented in Chapter 3 we always employ the PD-controller given in the same chapter due to its simplicity and since the very high accuracy in the control of the snake robot joints offered by the input-output linearizing controller is not crucial for testing the validity of the 2D model. Moreover, the PD-controllers given for the 3D model from Chapter 4 are always employed when simulating the 3D model.

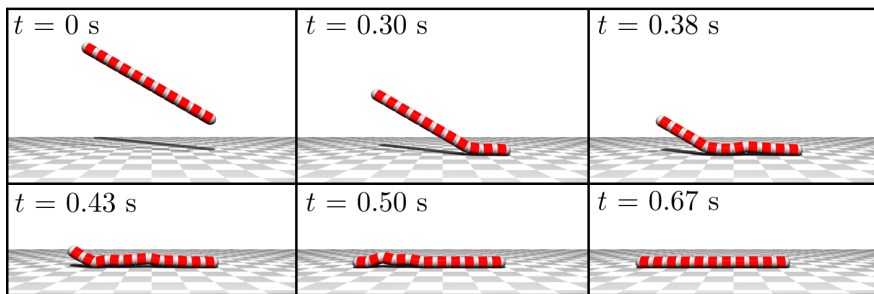


Figure 7.2: Snake robot during drop.

These simulations are given without an experimental validation due to several reasons. For example, the snake robot Aiko does not have the orthotropic friction property needed to move forward by lateral undulation. Moreover, we have not focused on producing the necessary measurement equipment to accurately measure the position of Aiko during 360 degree roll around the longitudinal axis of the robot or to implement the input-output linearizing controller. Instead, the focus has been on developing the theory around these aspects of locomotion and snake robot control.

All simulations are run in Matlab R2006a on an Intel Xeon 3 GHz processor with 2 GB RAM.

7.3.1 3D Model: Drop and Lateral Undulation

In this section the snake robot is dropped from an inclined angle and then set to move across the ground surface with lateral undulation. We include this simulation to show that the numerical treatment is stable even for such non-ideal situations and to show that the model of the snake robot moves forward by lateral undulation when the friction coefficient μ_{T_y} transversal to the snake robot links is higher than the friction coefficient μ_{T_x} longitudinal to each link ($\mu_{T_y} > \mu_{T_x}$).

Particular simulation parameters: $\mu_{T_y} = 0.5$, $r_N = 0.01$.

Motion pattern parameters: Lateral undulation

$$\begin{aligned} A_h &= \frac{40\pi}{180} \text{ rad}, & \omega_h &= \frac{80\pi}{180} \frac{\text{rad}}{\text{s}}, & \beta_h &= \frac{-50\pi}{180} \text{ rad}, \\ A_v &= 0 \text{ rad}, & \omega_v &= 0 \frac{\text{rad}}{\text{s}}, & \beta_v &= 0 \text{ rad}, & \beta_0 &= 0 \text{ rad}. \end{aligned} \quad (7.3)$$

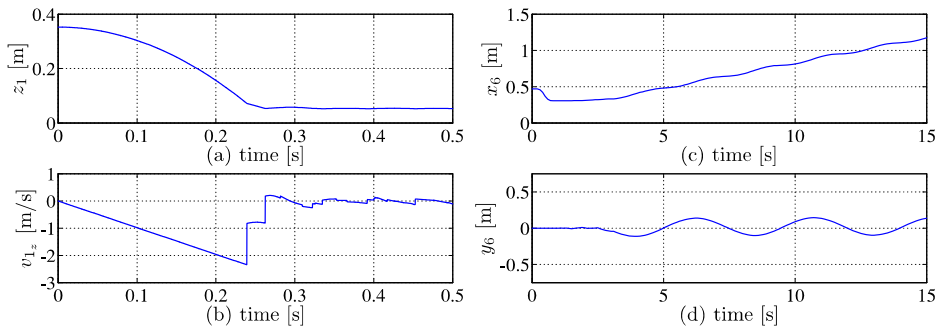


Figure 7.3: (a) Position and (b) velocity of link 1 during drop and (c)-(d) the position of link 6 during lateral undulation after the drop.

The snake robot is dropped at an inclined angle of 30 degrees with respect to the floor. The centre of gravity of the lowest link is lifted to $z_1 = 0.3525$ m (i.e. $L_{SC_i} + 0.3$ m) above the floor and all actuators are turned off until it has come to a resting position on the ground (we set $\tau_{v_i} = \tau_{h_i} = 0$ for $t \in [0, 1)$ s). Figure 7.2 shows the configuration of the snake robot while it drops to the floor. Figure 7.3 depicts (a) the vertical position z_1 of link 1 and (b) its drop velocity v_{1z} (parallel to the e_z^I -axis). As seen in Figure 7.3 (b), velocity jumps occur when the link hits the ground. We observe that the link does not immediately come to a resting position after having hit the ground even though the impact is modelled as completely inelastic. This is because the remaining links reach the ground shortly after and the impulses that arise from their impacts with the ground are transmitted through the joints so that v_{1z} also becomes discontinuous at the impact times. To be able to simulate the hard impact with the ground, we needed to reduce the value of r_N compared to the rest of the simulations of the 3D model. We see from Figure 7.3 (c) that the snake robot slides a little backward during, and shortly after, its impact with the ground. This is because it was dropped from an inclined angle.

At $t = 1$ the actuators are turned on again and the snake robot begins to progress forward with the motion pattern lateral undulation where the ‘soft start’ approach described in Section 7.2.2 is employed. The position of the CG of link 6 is depicted in Figure 7.3 (c)-(d) and we see that the snake robot moves forward along the e_x^I -axis. This is the expected qualitative motion of a snake robot moving by lateral undulation with the anisotropic friction property described above. We note that even though the snake

robot started from a non-idealized posture (after the drop), the numerical treatment was stable.

7.3.2 3D Model: U-shaped Lateral Rolling

The motion pattern lateral rolling with a U-shape is simulated in this section. To this end, we show that 360 degree rotations around the longitudinal axis of the snake robot does not pose a problem for the snake robot model and that the snake robot moves sideways on the ground surface due to the exoskeleton. In addition, we investigate what happens for U-shaped lateral rolling when the friction coefficients are changed.

Particular simulation parameters: Three combinations of friction coefficients: $(\mu_{T_x}, \mu_{T_y}) = (0.2, 0.2)$, $(\mu_{T_x}, \mu_{T_y}) = (0.2, 0.5)$ and $(\mu_{T_x}, \mu_{T_y}) = (0.01, 1)$.

Motion pattern parameters: U-shaped lateral rolling

$$\begin{aligned} A_h &= \frac{10\pi}{180} \text{ rad}, & \omega_h &= \frac{80\pi}{180} \frac{\text{rad}}{\text{s}}, & \beta_h &= 0 \text{ rad}, \\ A_v &= A_h, & \omega_v &= \omega_h, & \beta_v &= 0 \text{ rad}, & \beta_0 &= \frac{90\pi}{180} \text{ rad}. \end{aligned} \quad (7.4)$$

The snake robot is started with all links on the ground in a curved posture where the joint angles are set initially to $\phi_{h_i}(0) = \phi_{h_i,d}(0) = 0$ rad and $\phi_{v_i}(0) = \phi_{v_i,d}(0) = 10 \frac{\pi}{180}$ rad. With these initial conditions and the motion pattern parameters in (7.4), the snake robot rolls towards the ‘edge’ of the U-shape (see Figure 2.5 (a) for an illustration of a shape of a snake robot during U-shaped lateral rolling). We see from Figure 7.4 that the snake robot moves further for $(\mu_{T_x}, \mu_{T_y}) = (0.2, 0.5)$ than for $(\mu_{T_x}, \mu_{T_y}) = (0.2, 0.2)$ in an almost straight line. This suggests that the snake robot slips less and can utilize more of the rolling motion for locomotion. However, for $(\mu_{T_x}, \mu_{T_y}) = (0.01, 1)$ we observe that the path taken by the snake robot clearly deviates to the left in Figure 7.4 (a). This is because the snake robot sometimes starts sliding along its longitudinal axis due to the low friction coefficient μ_{T_x} and the difference $\mu_{T_x} \ll \mu_{T_y}$. This latter results suggests that for a snake robot with $\mu_{T_x} \ll \mu_{T_y}$ (implemented, for example, by adding small caster wheels in a tight grid on each link) a different shape of the snake robot may be advantageous to better control the direction of locomotion.

We have seen from the simulations that the snake robot moves sideways during U-shaped lateral rolling. This is possible since we have chosen to define an exoskeleton as a contact surface with the ground. If only, for

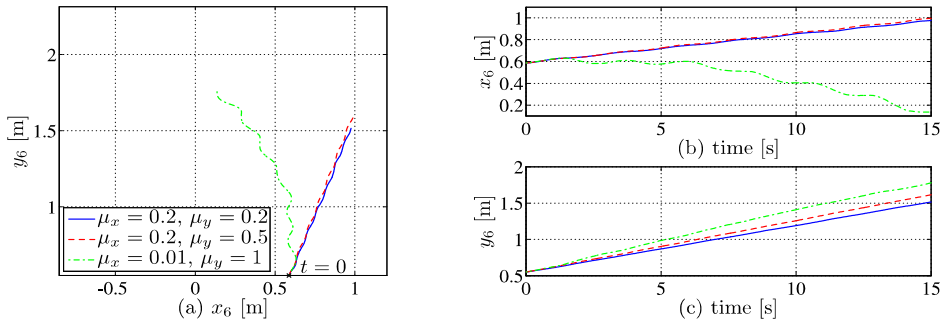


Figure 7.4: Position of link 6 during U-shaped lateral rolling along the e_x^I - and e_y^I -axis.

example, the centre of gravity would have been chosen as a contact point, then the rolling motion would not have been possible.

7.3.3 3D Model: Sidewinding

In this section we simulate sidewinding for various friction coefficients.

Particular simulation parameters: Three combinations of friction coefficients: $(\mu_{T_x}, \mu_{T_y}) = (0.2, 0.2)$, $(\mu_{T_x}, \mu_{T_y}) = (0.2, 0.5)$ and $(\mu_{T_x}, \mu_{T_y}) = (0.01, 1)$.

Motion pattern parameters: Sidewinding

$$\begin{aligned} A_h &= \frac{30\pi}{180} \text{ rad}, & \omega_h &= \frac{80\pi}{180} \frac{\text{rad}}{\text{s}}, & \beta_h &= \frac{-50\pi}{180} \text{ rad}, \\ A_v &= \frac{10\pi}{180} \text{ rad}, & \omega_v &= \omega_h, & \beta_v &= \beta_h, & \beta_0 &= \frac{90\pi}{180} \text{ rad}. \end{aligned} \quad (7.5)$$

We employ the ‘soft start’ approach and set the snake robot to move by sidewinding from a straight posture. Figure 7.5 depicts the position the centre of gravity of link 6 during locomotion. We see from Figure 7.5 (a) and (c) that the snake robot slides backward along the y -axis at certain intervals for $(\mu_{T_x}, \mu_{T_y}) = (0.01, 1)$. This is mostly due to the fact that one of the parts of the snake robot that instantaneously is supposed to be lying stationary on the ground slides longitudinally due to the large difference $\mu_{T_y} > \mu_{T_x}$ in friction coefficients. In addition, we observe that the direction of locomotion is dependent on the friction coefficients.

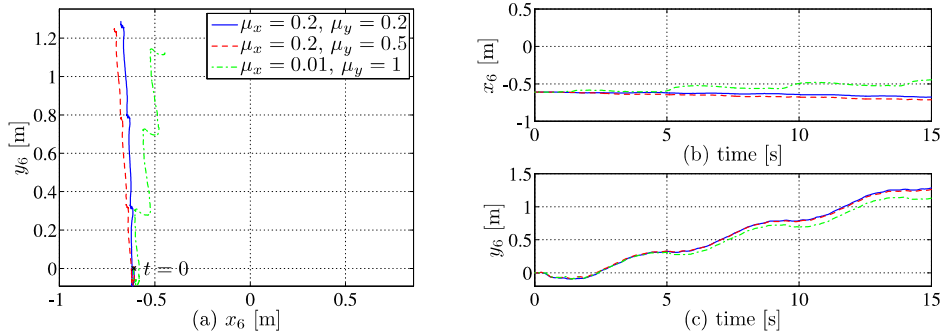


Figure 7.5: Position of link 6 during sidewinding for three different friction scenarios.

j	L_{H_j} [m]	H_j ([m], [m])	j	L_{H_j} [m]	H_j ([m], [m])
1	0.02	(0.087, 0.077)	6	0.01	(1.927, 0.110)
2	0.03	(0.474, 0.082)	7	0.02	(2.394, 0.153)
3	0.01	(0.882, 0.101)	8	0.03	(2.705, 0.061)
4	0.02	(1.201, 0.097)	9	0.01	(3.119, 0.128)
5	0.03	(1.671, 0.178)			

Table 7.3: Radii and positions of obstacles in the simulations in Section 7.3.4 for simulations of the 2D and 3D model.

7.3.4 2D and 3D Model: Obstacle Aided Locomotion

In this section we test and compare the 2D and 3D model for obstacle aided locomotion with small obstacles of various sizes. The contact forces between the snake robot and obstacles will be investigated and compared to the discussion on obstacle aided locomotion given in Chapter 5. In addition, we will investigate what happens if the position of one of the obstacles is altered.

Particular simulation parameters: We include $\nu = 9$ obstacles in the simulation. For each obstacle $j = 1, \dots, 9$ (counting from left to right in Figure 7.6 (a)) the radius L_{H_j} and position of the centre H_j in the (e_x^I, e_y^I) -plane are as given in Table 7.3. These parameters are used in the first set of simulations. Then all obstacles are rearranged for a discussion on what happens if the motion pattern parameters are changed. Finally, the obstacles are placed at their initial position, but the 3rd obstacle is first moved and then removed as explained below.

Motion pattern parameters: Lateral undulation is mostly performed with parameters given by (7.3) and employed for obstacle aided locomotion in this section. However, a short discussion is added for two other sets of motion pattern parameters. These parameters are indicated below.

The snake robot is set to move with the same motion pattern parameters as for lateral undulation shown in Section 7.3.1. However, in this section, the snake robot will move forward not due to the orthotropic friction property employed in Section 7.3.1, but, instead, due to pushing against obstacles.

The snake robot is initially (at $t = 0$ s) in a curved posture where its joints angles are found from $\phi_{h_i,d}(0)$ and $\phi_{v_i,d}(0)$ (the latter only applies to the 3D model). This posture is the basis for placing the obstacles along the natural path of the snake robot according to the criteria described in Section 5.5. In short, that is the obstacles are placed according to the initial configuration of the snake robot based on the choice of motion pattern parameters A_h , ω_h and β_h and they are located close to the centre line of the sinus-like shape of the snake robot. We employ various small obstacles to show that problems such as getting stuck do not occur for our choice of exoskeleton for the snake robot links. This problem is avoided since we use a moving contact point for contact between a snake robot link and an obstacle. Hence, every point on the exoskeleton of the snake robot is a possible contact point with an obstacle and therefore cannot an obstacle penetrate the snake robot body.

The position of link 6 of the snake robot is depicted in Figure 7.6, where in Figure 7.6 (a) we also see the obstacles placed according to Table 7.3. We see that the results from the 2D and 3D model are more or less identical. Hence, the results suggest that we may choose the 2D model to optimise the simulation time when purely planar snake robot motion is investigated, while the 3D model can be employed when additional 3D motion is required. Moreover, we observe that the snake robot is able to traverse the obstacle course, and its path is almost sine-like except from some ‘bumps’ here and there. These ‘bumps’ arise from the definition of the exoskeleton of the links. Each link has a hemisphere (or a semicircle for the 2D model) at the two ends of the link, and the ‘bumps’ in the path comes from that the obstacles are so small that they may slide slightly into the small ‘crack’ between two adjacent links.

We also conclude from the simulation result that the numerical treatment of the 2D and 3D model with the time-stepping method worked well

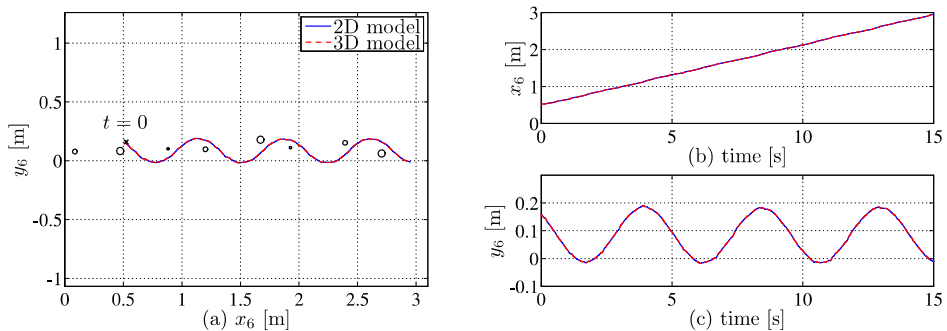


Figure 7.6: Position of link 6 during obstacle aided locomotion. The perimeters of the obstacles $j = 1, \dots, 9$ (counting from left to right) are depicted as black circles in (a).

even though the obstacles were small. Hence, the contact impulses between the snake robot was calculated accurate enough to keep the obstacles from penetrating the exoskeleton of the snake robot.

In the following, we will use the simulation results from the 2D model to investigate the contact forces between the snake robot and the obstacles. To this end, we quantify the instantaneous value of the contact force that acts on the snake robot from the obstacles as

$$\mathbf{F}_H = \frac{1}{\Delta t} \sum_{b \in \mathcal{H}_M} (\mathbf{W}_{HM})_b P_{H_b} \quad (7.6)$$

where \mathcal{H}_M is the set of active contact points, P_{H_b} is contact impulse during the current time-interval and $(\mathbf{W}_{HM})_b$ gives the direction of the contact impulse (see the explanation of (3.67) on page 62). Hence, \mathbf{F}_H consists of both forces and possible impulses and it is found from the integral of the forces during each time-step.

Figure 7.7 (a) depicts the snake robot at $t = 5.2$ s in an environment with circular obstacles. The snake robot is moving to the right and we observe that this is also the direction in which the sum of forces, acting on the snake robot from the obstacles points. The contact forces found from (7.6) are concurrent with the illustration of obstacle aided locomotion in Figure 5.3, hence it is the sum of contact forces that results in the forward (toward the right in Figure 7.7 (a)) motion of the snake robot.

The linear contact forces found from (7.6) are plotted from $t = 4.5$ s to $t = 5.5$ s for links 2, 7 and 10 in Figure 7.7 (b)-(c). Hence, the figures depict

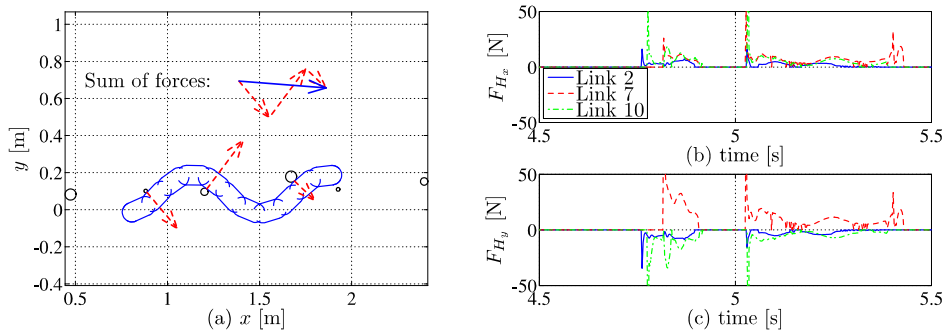


Figure 7.7: (a) Outline of snake robot at $t = 5.2$ s during obstacle aided locomotion. The contact forces are indicated with red arrows and the snake robot is moving toward the right. (b)-(c) The contact forces on links 2, 7 and 10.

how the contact forces develop around the time the snake robot is in the position illustrated for in Figure 7.7 (a). The spikes in the forces arise from impulses due to the link colliding with an obstacle. The maximum values of these spikes have not been included in the plot to depict the contact force during the rest of the time-interval more accurately. We see from Figure 7.7 (c) that the average applied force to link 7 is higher than to the other two links. Comparing to Figure 7.7 (a), we see that the difference is because link 7 is the only link that is subjected to a force pointing in the negative e_y^I -direction. Hence, this contact force balances the contact forces acting on the other two links (links 2 and 10).

We see from Figure 7.7 that the snake robot is barely ‘curled enough’ to be able to reach and take advantage of the necessary three contact points discussed in Chapter 5. To this end, a comment on the choice of the motion pattern parameters A_h , β_h , and ω_h is imminent. We see from Figure 7.9 (a) that a too low value of β_h results in that the snake robot (with only 11 links) is no longer able to be in contact with three push-points simultaneously, and this renders it unable to move forward by lateral undulation. However, if we increase the value of β_h as illustrated in Figure 7.9 (b), then snake robot is able to move forward, but the angle between the link in contact with an obstacle and the direction of progress becomes small. Hence, most of the reaction force from the obstacle will point normal to the direction of motion (since it points transversal to the link), and this leads to very inefficient locomotion since larger joint torques are needed to propel the

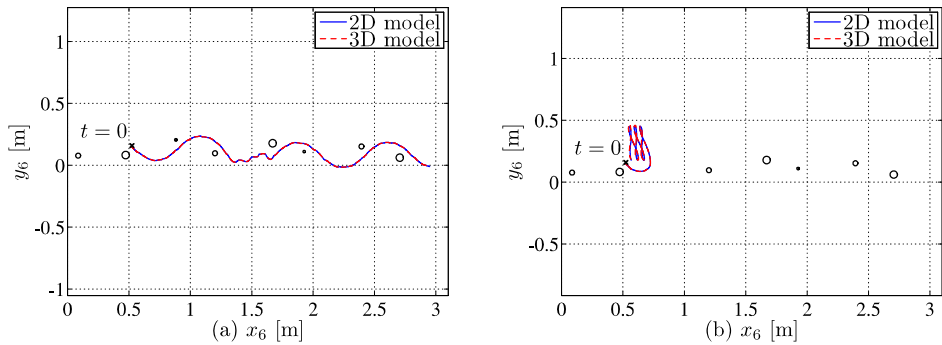


Figure 7.8: Position of link 6 of the snake robot during a 15 second simulation with the 3rd obstacle (a) moved upwards and (b) removed from the simulation.

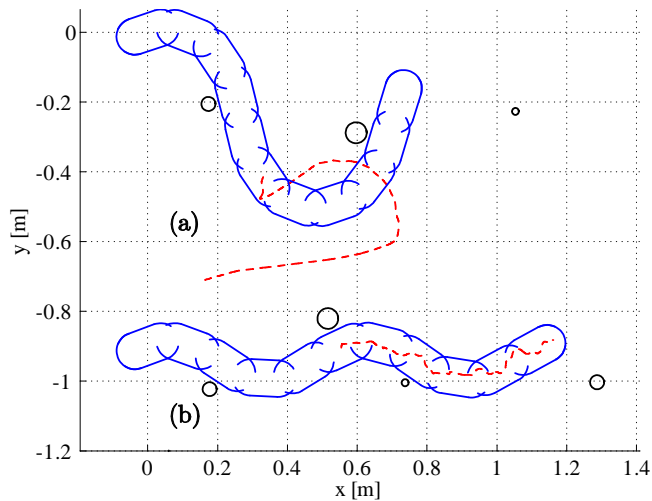


Figure 7.9: Posture at $t = 0$ s of snake robot for (a) $\beta_h = -30\pi/180$ rad and (b) $\beta_h = -70\pi/180$ rad. The positions of the middle links from $t = 0$ s to $t = 5$ s are indicated by the (red) dashed line.

snake robot forward (i.e. the contact force needs to be higher to obtain the same amount of force tangential to the direction of motion with three push-points). Moreover, the increased contact force will result in that a real snake robot can get easier stuck since the friction forces between the obstacles and the snake robot will increase. The increase in contact force for a posture with ‘more curls’ corresponds to the increase in the constraint forces on the wheels for a wheeled snake robot with the same posture during lateral undulation (without obstacles) as described in Date *et al.* (2000). The angle between the link in contact with the obstacle and the direction of locomotion will also be smaller if we choose to keep β_h at its original value, but lower the amplitude of joint oscillation A_h . Hence, the value of β_h should be chosen such that the snake robot is just ‘curled enough’ to utilize the push-points for locomotion, and the amplitude of oscillation should, up to some extent, be chosen as high as possible to obtain a large angle between the direction of locomotion and the links in contact with the push-points. However, it is important to be aware of that if the obstacles are located far from each other, a larger value of β_h is necessary to allow the snake robot to reach all three necessary push-points. Moreover, the wave propagation speed is dependent on β_h , so this also has to be taken into account when choosing the motion pattern parameters.

It now remains to comment on the value of the angular frequency ω_h . For an infinitely strong snake robot (both with respect to actuator torques and outer shell) and fixed obstacles that do not yield, a larger ω_h will result in a greater forward speed since the propagating wave is propagated faster toward the back of the snake robot. However, the larger the speed, the harder the impact between the snake robot and the obstacles will become. Also, the speed will be limited by the amount of torque the actuators are able to produce. Moreover, the obstacles may not always be fixed to the ground, such as a chair or a small rock, hence it may in some cases be necessary to limit the forces exerted on the obstacle to keep them fixed. This is also achieved by limiting the angular frequency ω_h . From the discussion above, we choose to keep the motion parameters as they are for the upcoming experiments.

In addition to the perfect scenario depicted in Figure 7.8 (with respect to the connection between the motion pattern and obstacle placement) described above, we set the obstacles according to Table 7.3 and briefly explore the scenario where obstacle 3 (counting from the beginning of the track) is partly and completely removed. First, obstacle 3 is moved one snake robot link diameter along the e_y^I -axis. Thus, its new position in the

horizontal (e_x^I, e_y^I) -plane is

$$H_{3,moved} : (0.882 \text{ m}, 0.101 \text{ m} + 2L_{SC1}). \quad (7.7)$$

Figure 7.8 (a) illustrates the path of link 6 of the snake robot when it moves through the altered track. The simulation shows that the snake robot started sliding to the left and rotating anti-clockwise when it was supposed to collide with the moved obstacle. However, the deviation from its original path was not substantial enough to hinder the snake robot in reaching obstacle 4 sufficiently accurate to continue its path along the track. Secondly, obstacle 3 is completely removed and the new path of the snake robot is depicted in Figure 7.8 (b). Then the snake robot also rotated anti-clockwise and slid to the left when it was supposed to hit obstacle 3, but this time it was unable to complete the track since it never collided with the third obstacle, and therefore never regained the third contact point needed for locomotion as discussed in Section 5.5.

We might expect that the snake robot fails to move through the entire track with one obstacle moved or missing since the desired joint pattern is preprogrammed. However, the results displayed in Figure 7.8 show that some repositioning of the obstacles is possible. Hence, a control algorithm for intelligent obstacle aided locomotion does not need to provide a motion pattern fully accurate compared to the position of the obstacles. The snake robot still manages to move through the track even if it touches some obstacles in a far from perfect manner. The downside can be that larger joint torques are needed as discussed in Section 5.3.

7.3.5 Robot Model: Comparison of Controllers

The input-output linearizing controller presented in Chapter 6 will be compared to a PD-controller in this section. The comparison will be performed for the (planar) process plant model, or ‘robot model’, presented in the same chapter for both a smooth approximation to Coulomb friction, and Coulomb friction described by a set-valued force law. In addition, we will see how the controllers perform when noise is introduced in the system and the suitability of the ‘robot model’ for numerical treatment is discussed.

Particular simulation parameters: None. All parameters are given in Tables 7.1 and 7.2. Note that (μ_{C_x}, μ_{C_y}) and (μ_{V_x}, μ_{V_y}) are employed in the robot model instead of (μ_{T_x}, μ_{T_y}) since we here have viscous friction in addition to Coulomb friction.

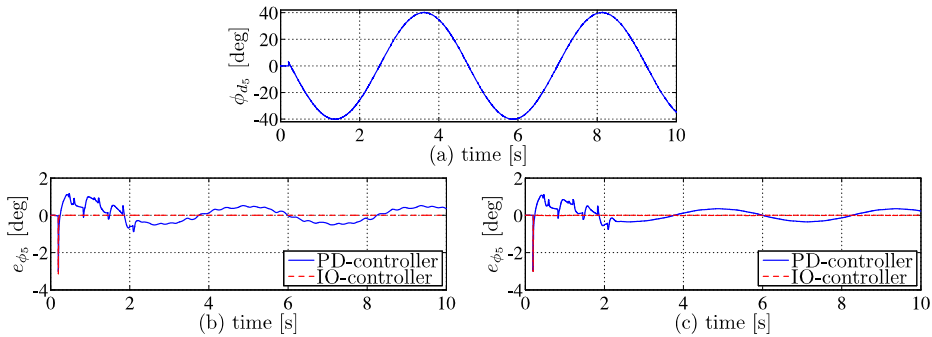


Figure 7.10: (a) Reference angle for joint 5 during lateral undulation, (b) the tracking error for the same joint with the smooth approximation to the Coulomb friction and (c) the tracking error from simulations with the set-valued Coulomb friction model.

Motion pattern parameters: Lateral undulation with parameters given by (7.3).

The PD-controller is implemented for the robot model as

$$\tau = -k_D (\dot{\phi} - \dot{\phi}_d) - k_P (\phi - \phi_d), \quad (7.8)$$

where the elements of the vector of desired joint angles $\phi_d \in \mathbb{R}^{10}$ are denoted ϕ_{d_i} and found from $\phi_{h_i,d}$ in (7.1), and $\phi \in \mathbb{R}^{10}$ is the vector of snake robot joint angles found directly from the generalized coordinates as shown in (6.3), (6.30).

The time-stepping method is used for numerical integration of the ‘robot model’ with set-valued Coulomb friction, while the Matlab solver `ode15s` is used for the smooth version of the model. We remember from Chapter 6 that the control torque found from the input-output linearizing controller is based only on the smooth friction model, even for control of the ‘robot model’ with set-valued Coulomb friction. The initial posture of the snake robot is straight and then it is set to move with lateral undulation using the ‘soft start’ approach.

First, the simulations were run without noise and the tracking error for one of the two middle joints is depicted in Figure 7.10 for both the smooth and the non-smooth model. Figure 7.10 (b) suggests that the input-output linearization controller yields an asymptotically stable error dynamics, while the PD-controller results in a periodic tracking error. We

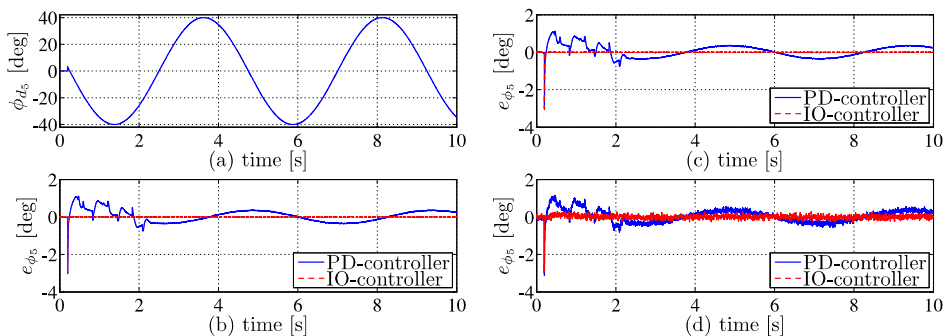


Figure 7.11: Tracking error for joint 5 from the simulations of the non-smooth model with added noise according to Table 7.4 and (7.9)-(7.14). The reference signal for the joint is depicted in (a).

see from Figure 7.10 (c) that the tracking error for the controller based on input-output linearization (i.e. the ‘IO-controller’) is almost negligible despite that we are no longer able to cancel the snake robot dynamics correctly with the feedback term (since the feedback term is based on a smooth friction model and the plant itself is simulated as set-valued). In addition, we find that the tracking error for the PD-controller has a sinus-like shape and almost similar in form but somewhat smaller in amplitude compared to the case with smooth Coulomb friction.

We have earlier argued that the ‘robot model’ is not suitable for simulating a snake robot due to its high number of degrees of freedom. However, in this section we employ this model for simulation anyway so this point naturally needs some attention. The time-stepping method with the Modified Newton Algorithm is designed to simulate systems with set-valued force laws and possibly discontinuous velocities. However, with the smooth approximation of the friction model, we no longer have such a system and a conventional numerical integrator can be employed instead. Moreover, since the Modified Newton Algorithm is no longer employed, we need a new way to handle the bilateral constraints (introduced by the joints). One way of doing this is to employ *minimal* coordinates instead (as done in the ‘robot model’). The minimal coordinate representation of the snake robot gives us the system matrices employed in the feedback law more or less directly. Hence, we do not have to calculate these matrices from scratch in addition to system matrices developed for a model with non-minimal coordinates. The 2D model in Chapter 3 is implemented with a single point on each link

for the frictional contact between the snake robot and the ground surface. This choice has been made to simplify the model. Hence, such an approach results in that a purely rotational motion of a link does not give rise to any friction forces. Such friction forces are needed in the stability analysis provided of the closed-loop system with the input-output linearizing controller. Due to the reasons mentioned above, we decided to try to use the smooth ‘robot model’ for simulation in order to compare the PD-controller and the input-output linearizing controller. We see from Figure 7.10 that we were able to simulate the system with this approach, and to avoid having to redesign the 2D model to add a friction point and viscous friction, we also used the same approach (now with the time-stepping method) for the ‘robot model’ with set-valued friction. Using this approach was okay, *but only since we needed a very limited number of simulation trials*. This is because the simulations were *very* time-consuming. This was partly due to the stiff differential equations which was due to the smooth approximation of the Coulomb friction, but also due to the enormous amount of terms that had to be calculated in order to find the system matrices. To illustrate the difference in computation time, we mention that it took between 20 and 50 minutes to simulate a single second of snake robot motion with the ‘robot model’, and with the 2D model it only took about 0.5 minutes to simulate a second of snake robot motion.

For the second set of simulations, we add various degrees of noise to the measurements and test the IO-controller and the PD-controller in simulations where the set-valued Coulomb friction model is employed. The noise is added to the measurements with the Matlab function `rand` which produces approximately normally distributed numbers between 0 and 1. Furthermore, the states utilized in the feedback (i.e. the measurements with noise) are found from

$${}^I\mathbf{r}_{O_2,m} = {}^I\mathbf{r}_{O_2} + N_p (2 \text{rand}(2, 1) - 1), \quad (7.9)$$

$$\theta_m = \theta + N_\theta (2 \text{rand}(1, 1) - 1), \quad (7.10)$$

$$\phi_m = \phi + N_\phi (2 \text{rand}(10, 1) - 1), \quad (7.11)$$

$${}^I\mathbf{v}_{O_2,m} = {}^I\mathbf{v}_{O_2} + N_v (2 \text{rand}(2, 1) - 1), \quad (7.12)$$

$$\dot{\theta}_m = \dot{\theta} + N_{\dot{\theta}} (2 \text{rand}(1, 1) - 1), \quad (7.13)$$

$$\dot{\phi}_m = \dot{\phi} + N_{\dot{\phi}} (2 \text{rand}(10, 1) - 1), \quad (7.14)$$

where N_p , N_θ , N_ϕ , N_v , $N_{\dot{\theta}}$ and $N_{\dot{\phi}}$ are positive constants in \mathbb{R} , and `rand(a, b)` is the Matlab function `rand` that returns a matrix (or vector) with a rows and b columns with approximately normally distributed numbers between 0 and 1.

Figure 7.11	N_p [m]	N_θ [rad]	N_ϕ [rad]	N_v [m/s]	$N_{\dot{\theta}}$ [rad/s]	$N_{\dot{\phi}}$ [rad/s]
(b)	0.1	$0.1 \frac{\pi}{180}$	$0.1 \frac{\pi}{180}$	0.1	$0.1 \frac{\pi}{180}$	$0.1 \frac{\pi}{180}$
(c)	10	$10 \frac{\pi}{180}$	$0.1 \frac{\pi}{180}$	10	$10 \frac{\pi}{180}$	$0.1 \frac{\pi}{180}$
(d)	10	$10 \frac{\pi}{180}$	$1 \frac{\pi}{180}$	10	$10 \frac{\pi}{180}$	$1 \frac{\pi}{180}$

Table 7.4: Amplitude of noise added to the measurement employed in the feedback control corresponding to Figure 7.11.

Figure 7.11 depicts the reference signal together with the tracking errors for three simulations with various degrees of noise added to the measurements employed for feedback. Table 7.4 contains the amplitudes of the added noise. We see from Figure 7.11 (b) and (c) that as long as the noise added to the joint angle measurements is kept low, then both the results utilizing the IO-controller and PD-controller, respectively, are not significantly affected. Hence, even though the feedback term in the IO-controller contains the velocity of the snake robot, it is not significantly affected by adding noise to these states. We note from Chapter 6 that we assume that all states are available for feedback.

We see from Table 7.2 that a very high gain is used to control the joints accurately with the IO-controller. Hence, a large error will yield a large control torque that may not be possible to realize in an actual snake robot. Hence, this is an aspect that needs to be considered for future work. However, when the error is small (which is the case a short time after the snake robot has been started), then the IO-controller will not require any substantially larger control torque than the PD-controller. It is also important to comment on that the gains for the PD-controller and the IO-controller are not directly comparable since certain elements of the mass matrix (and its inverse) are included by multiplication before the final control torque is found from the IO-controller (see 6.58 on page 121). A possible solution to the problem of high control torques during start up, is to employ a reference generator which accounts for the initial posture of the snake robot, and then guides the actual reference angle to the desired reference signal without causing it to jump. To investigate the difference in gains between the PD-controller and the IO-controller, we tried to increase the PD-controller gains. However, this resulted in such a stiff system that we were not able to simulate it within a reasonable amount of time (a 20 second simulation was aborted since it did not complete within 48 hours). We have also tried adding a ± 10 N saturation on the control

torques for both controllers. Then, the results from stationary motion were the same as Figure 7.10. However, the steps in the reference signal during the ‘soft start’ approach caused a very short, but larger tracking error for the IO-controller compared to the PD-controller.

The smooth approximation of the Coulomb friction in the ‘robot model’ results in a stiff system. This is not advantageous for simulation and the 2D model in Chapter 3 should be employed instead. However, since the non-linear control techniques employed in this thesis are only fit to cope with smooth non-linear systems, the smooth approximation is necessary for the controller synthesis and stability proof. Hence, the system matrices for the minimal coordinates needed to be found in order to implement the control law (which is constructed from these matrices). Thus, even though the simulation time became much longer due to the need for small time step due to e.g. the stiff system equations, it was more effective to simulate the robot model since few simulations were required to compare the controllers.

7.4 Experimental Setup

The snake robot ‘Aiko’ is employed in the experiments, and the snake robot together with the rest of the equipment used to conduct the experimental tests are described in the following.

Each 2 DOF cardan joint of the snake robot is actuated by two 6 W DC-motors. Each the two joint angles are controlled by simple P-controllers implemented on an Atmel ATmega128 microcontroller. Hence, 10 micro-controllers are employed to control the 10 cardan joints. The snake robot joint reference angles are sent with a frequency of 10 Hz from a PC running Matlab, via a CAN-bus (a particular type of field bus) to the micro-controllers. The position of the centre of the middle link (link 6) is tracked using a Vicon MX Motion Capture System with 4 cameras (MX3) together with Matlab Simulink. The Vicon programme (Vicon iQ 2.0) runs on a computer with 4 Intel Xeon 3 GHz processors and 2 GB RAM. This was the same computer employed for the simulations. The logging of motion data is synchronized in time through a TCP-connection between the PC that controls Aiko and the Vicon-computer at the start-up of the transmission of the desired snake robot joint angles. Data logging is performed at 20 Hz. We have chosen to let the position of link 6 represent the position of the snake robot, because then we filter any transient behaviour of the snake robot that might occur at its ends and it is thus easier to focus on the general motion of the snake robot.

The snake robot moves on particle boards in the experiments. For obstacle aided locomotion, we attach obstacles cut from a PVC-pipe with 0.25 m outer diameter to the particle boards.

7.5 Simulations with Experimental Validation

In this section we compare the simulation and experimental results for sidewinding, lateral undulation with isotropic friction and obstacle aided locomotion. We compare both qualitative and quantitative motion to see to what degree the models can reproduce the motion of the snake robot ‘Aiko’. Moreover, by investigating obstacle aided locomotion, we can learn whether the exoskeleton defined for the snake robot models resembles the actual contact between Aiko and the obstacles in the experimental setup. We employ the PD-controllers described in Chapters 3 and 4 for the 2D and 3D models, respectively. The experimental setup described in Section 7.4 has been employed to conduct and log the experiments.

7.5.1 3D Model: Sidewinding

In this section we compare the simulation results with the 3D model and the experimental results with Aiko for the sidewinding motion pattern.

Particular simulation parameters: None.

Motion pattern parameters: Sidewinding with the same parameters as in (7.5).

The resulting motion of sidewinding with isotropic friction for the 3D model of the snake robot is compared to experimental results found by letting Aiko move with the same motion pattern. We employ the motion pattern parameters given by (7.5) and calculate the reference angles for the joints from (7.1)-(7.2) with the ‘soft start’ approach starting from a straight and flat posture along the e_x^I -axis as performed in the sidewinding simulations above. The motion of the model and Aiko is depicted in Figure 7.12. As seen from the picture, Aiko was not wearing its metal head. The head was removed in order to obtain a uniform weight distribution along the snake robot.

Figure 7.13 displays the position of link 6 for the 3D model and Aiko during sidewinding. We observe from the figure that the model almost follows the e_y^I -axis, while Aiko steadily turns somewhat to the right. In addition,

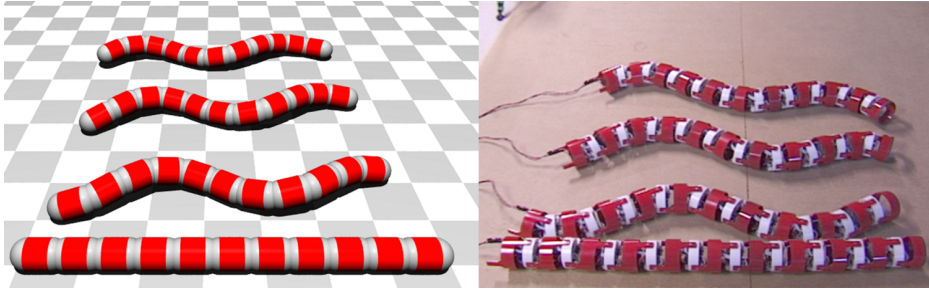


Figure 7.12: Comparison of simulation (left) and experimental (right) results during sidewinding. The pictures are taken at $t = 0, 5, 10, 15$ s and the snake robot is moving upwards in the illustration.

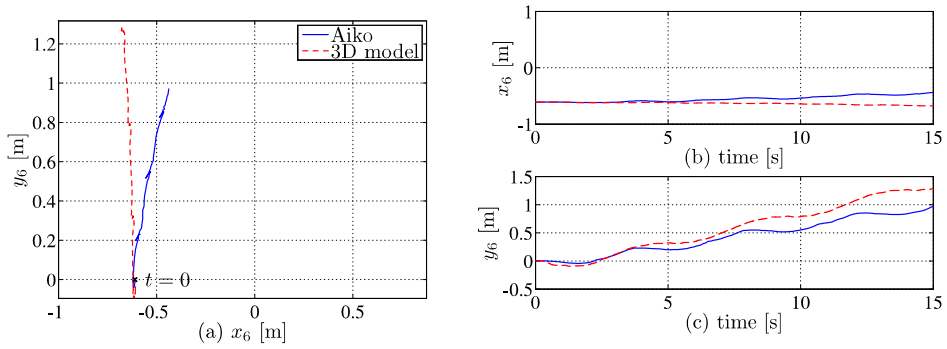


Figure 7.13: Position of link 6 of Aiko and the 3D model during sidewinding.

the model covers a greater distance than Aiko. We see from Figure 7.13 (a) and (c) that the simulation results have the same trend and approximate frequency as the experimental results along the e_y^I -axis. The variance in trend is more noticeable along the e_x^I -axis due to the turning motion of Aiko. Moreover, a slight initial difference between the orientation of Aiko and the model may contribute to the discrepancy in heading. We discuss in Section 7.5.4 various reasons for the differences between the simulation and experimental results. For sidewinding, we believe that one of the most important differences between the 3D model and Aiko is that Aiko is not able to control its joints accurately (see Section 7.5.4). This is particularly the case for vertical (lifting) motion which is needed for sidewinding. By inspecting Aiko closely during the experiment, we noticed that a large part of its body was touching the ground at the same time. This was not the case in the simulation where the snake robot joints were accurately controlled which resulted in that most of its body was lifted from the ground during locomotion.

We observe from Figure 7.13 (a) and (c) that Aiko sometimes slides a little backward (at $t \approx 5$ s, $t \approx 9$ s and $t \approx 14$ s). We have tried to reproduce this phenomenon in the 3D model by lowering and increasing the friction coefficient. We have also tried a variety of anisotropic friction properties, but without any luck. The backward motion might be a result of Aiko not being able to lift its links properly. The links that are supposed to be lifted and moved forward are instead sometimes dragged along the particle board. This results in a friction force which acts in the opposite direction of the desired motion of the snake robot. This friction force might result in that some of the parts that are supposed to be lying stationary on the ground are instead pushed slightly backward.

7.5.2 2D and 3D Model: Lateral Undulation with Isotropic Friction

Although not commonly used, it is possible for a snake robot to use lateral undulation for locomotion when the friction between the ground and the snake robot is isotropic (Nilsson, 2004). In this section we compare the simulation results from the 2D and 3D model with experimental results of Aiko moving by lateral undulation.

Particular simulation parameters: None.

Motion pattern parameters: Lateral undulation with motion pattern parameters given by (7.3).

We let the snake robot start in a curved posture with its centre line approximately along the e_x^I -axis and with its initial joints angles given by $\phi_{h_i,d}(0)$ and $\phi_{v_i,d}(0)$ (the latter only applies for the 3D model). Figure 7.14 shows both the position of link 6 calculated from the mathematical models in the simulations and the position logged from the experiment with Aiko. We see that the snake robot moves slowly backward along the e_x^I -axis in both the simulations and the experiment. The average speed of the link is approximately -1 cm/s. From Figure 7.14, we see that the both the 2D and the 3D model display the same trend in motion as observed in the experiment. However, the distance travelled is greater in the simulations and we have tried varying the friction coefficients (while still keeping isotropic friction), but this did not seem to have a noticeable effect on the simulation results. The difference in distance travelled may come as a consequence of that the model of the snake robot is able to control its joints more accurately and thus is able to reach the maximum amplitude (40 degrees) when it is required to do so. Also, this suggest that the frictional contact between the snake robot and the ground surface is more complex than what has been accounted for in the 2D and 3D model. This statement is also backed up by the fact that the simulation results differs between the 2D and the 3D model. The differences between the two simulations come from that two contact points are employed for contact between a link and the ground surface in the 3D model, while only a single point is employed for contact in the 2D model. We have defined two contact points in the 3D model to avoid that parts of a link penetrates the ground surface while still keeping the contact model as simple as possible (if just a single point would have been chosen for the contact point, then the remaining parts of the link can penetrate the ground surface during 3D motion).

In the following section we will see how the snake robot moves with the exact same motion pattern in an environment with obstacles. To compare the two trials, we have kept the metal sphere head on Aiko for the lateral undulation on an isotropic surface as shown in this section.

7.5.3 2D and 3D Model: Obstacle Aided Locomotion

The simulation and experimental results for obstacle aided locomotion will be compared in this section. By doing this we can check whether the models are appropriate test beds for such locomotion, and we can also get an indication of whether the definition of the exoskeleton of the models is valid for describing Aiko.

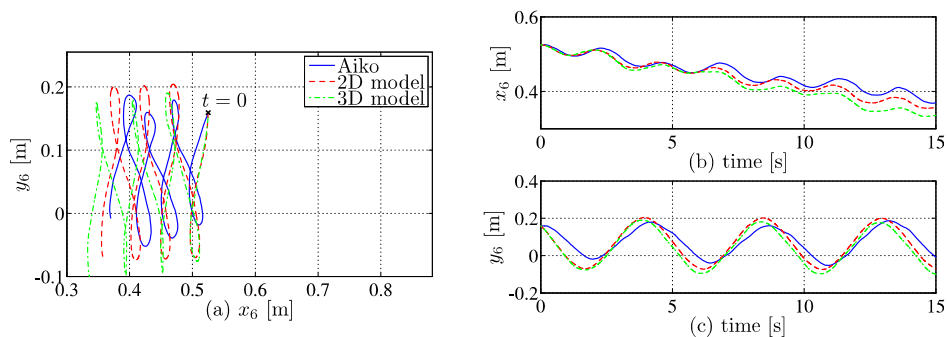


Figure 7.14: Position of link 6 during lateral undulation with isotropic friction.

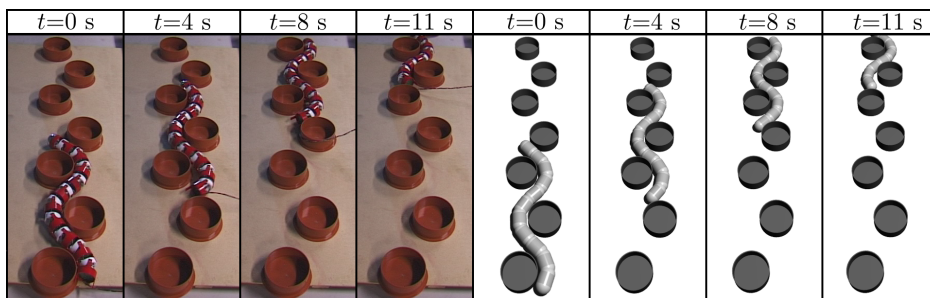


Figure 7.15: Experimental results with Aiko (left) and simulation results from the 3D model (right) during obstacle aided locomotion by lateral undulation.

j	L_{H_j} [m]	H_j ([m], [m])	j	L_{H_j} [m]	H_j ([m], [m])
1	0.125	(0.051, 0.176)	5	0.125	(1.530, 0.178)
2	0.125	(0.422, 0.003)	6	0.125	(1.888, 0.002)
3	0.125	(0.804, 0.185)	7	0.125	(2.259, 0.175)
4	0.125	(1.153, 0.003)			

Table 7.5: Radii and positions of obstacles in the simulations and experiments described in Section 7.5.3.

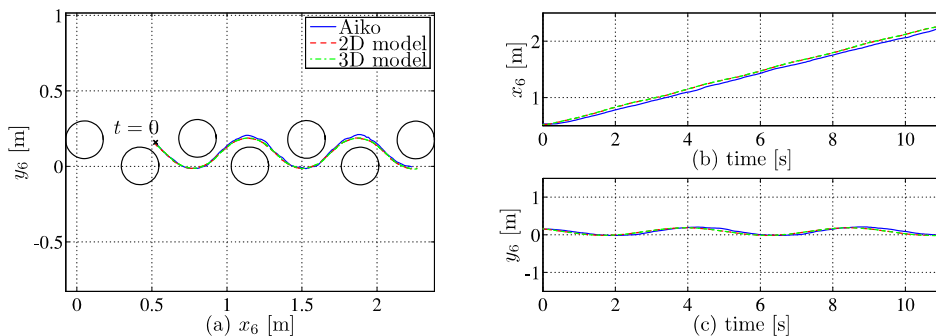


Figure 7.16: Position of link 6 during obstacle aided locomotion in the simulations and the experiment. The circumferences of the obstacles $j = 1, \dots, 7$ (counting from left to right) are depicted as black circles in (a).

Particular simulation parameters: We include $\nu = 7$ obstacles in the simulation and the experiment. For each obstacle $j = 1, \dots, 7$ (counting from left to right in Figure 7.16 (a)) the radius L_{H_j} and position of the centre H_j in the (e_x^I, e_y^I) -plane are given in Table 7.5.

Motion pattern parameters: Lateral undulation with the same parameters as in (7.3).

The snake robot is set to move with exactly the same motion pattern as in the previous subsection, however now it moves forward and much faster because of the obstacles that it pushes against. Moreover, Aiko is wearing its metal sphere head to easier slide along the obstacles it encounters.

The position of link 6 during the simulations and the experiment are shown in Figure 7.16. The positions and diameter of the obstacles are decided based on the elaboration in Section 5.5 and this is the same principle as is employed in Section 7.3.4. The positions of the obstacles were used to synchronize (in the horizontal plane) the positions logged in the experiment and the positions found in the simulations of link 6. The snake robot was placed among the obstacles before it started to move as shown in Figure 7.15 for $t = 0$. Its initial joint angles were set to $\phi_{h_i}(0) = \phi_{h_i,d}(0)$ and $\phi_{v_i}(0) = \phi_{v_i,d}(0)$. We see from Figure 7.16 that Aiko traverses its path among the obstacles along the e_x^I -axis with approximately the same speed (ca. 15 cm/s) in the simulation as in the experiments. This is 15 times faster compared to the results without the obstacles. Moreover, the snake robot moves in the opposite direction (with respect to the wave that

is propagated down its body) of what it did for isotropic friction without obstacles. Hence, for isotropic friction without obstacles, the direction of motion is the same as the direction the undulating wave that is propagated along the snake robot body. However, for anisotropic friction or with obstacles to push against the direction of motion is opposite to the direction of the propagating wave. Another way of increasing the velocity of a snake robot on a flat surface is to attach passive caster wheels on the underside of the robot. However, this approach has some downsides. First of all the snake robot is more dependent on the ground surface condition for effective locomotion. Secondly, the speed of the snake robot is dependent on the angular frequency of oscillation ω_h in (7.1) for a fixed δ_h and A_h (Saito *et al.*, 2002), and for a snake robot with passive wheels, the side-ways slip has to be considered when setting ω_h (Wiriyacharoensunthorn and Laowattana, 2002). However, the only practical limitation for the wave propagation speed during obstacle aided locomotion with obstacles that do not yield (i.e. move or deform when pushed against) is the speed of the joint motors. It is therefore possible to gain great velocities using obstacle aided locomotion.

7.5.4 Discussion of the Experimental Validation

We have seen in the previous section (Section 7.4) that the simulation and experimental results range from being very similar, as for the obstacle aided locomotion in Figure 7.16, to comparing fairly well for sidewinding in Figure 7.13. Consequently, the differences between the results need to be addressed and in the following we will list the most important possible sources for the discrepancies:

- Aiko has a noticeable free play in the joints of about 3 – 5 degrees. This results in that the control of the joint angles is not completely accurate and the joint angle might not be able to reach its desired angle.
- The dynamics of the actuators are not modelled. Hence, the actuators in the models are extremely strong and fast, and we are able to accurately control the actual angle of each joint to its desired position. This is not the case for Aiko where the joint motors sometimes saturate and are unable to track the desired joint movement precisely. This particularly the case for vertical motion where larger joint torques are required.

- For each cardan joint, there is in fact a 2.2 cm distance between the rotational axis for the yaw and pitch motion. However, to simplify the kinematics in the models, we have assumed that the axes cross each other.
- The varnish on the underside of the snake robot has been worn down. This has resulted in that the snake robot links slide easier when rolled slightly to one of the sides where the varnish is still present.
- The Stribeck effect (see e.g. Leine and Nijmeijer (2004)) is not included in the model of the friction. The Stribeck effect states, roughly speaking, that the friction forces acting on a body is reduced just after it is set in motion.
- The exoskeleton of Aiko is not modelled precisely since this would require a great deal of consideration of the geometry of the snake robot and this would severely complicate the expressions for the contact with the environment.
- There was a time-delay between the start-up of the snake robot and the start-up of the logging of position data. This delay ranged between 50 and 150 ms.
- We note from the simulations with lateral undulation with isotropic friction that there is a difference between the simulation results of the 2D and the 3D model. The difference may arise from the fact that two friction points are employed for each link in the 3D model to handle the impacts with the ground in an effective manner, while the 2D model only has one friction point per link to minimize the complexity of the model and thus increase the simulation speed.

Even though there are several sources of error, we see from the plots that compare the simulation and experimental results that the models give a satisfactory qualitative description of the snake robot dynamics. Especially the direction of locomotion is fairly accurate. For the sidewinding motion pattern, the model is not accurate enough to predict the quantitative motion of the snake robot for an extended period of time, however it gives a clear indication of how the snake robot will move. This suggests that the model can be used to develop and test new 3D motion patterns for snake robots that can later be implemented on physical snake robots such as Aiko.

7.6 Summary

In this chapter we present back-to-back comparisons between simulation and experimental results for a selection of serpentine motion patterns. Furthermore, additional simulation results are given to test motion patterns and controllers which we have not been able to investigate in the experimental setup with our snake robot Aiko.

We see from the simulations that the time-stepping method with the Modified Newton Euler Algorithm is suitable for numerical treatment of the 2D and 3D model. This is verified, partly by simulating that the snake robot drops to the ground and then moves forward by lateral undulation. For this scenario the numerical treatment is stable even though the contact forces are large and many impacts occurs.

The 3D model gives realistic simulation results of both sidewinding and U-shaped lateral rolling. To this end, the snake robot moves sideways by lateral rolling due to our choice of an exoskeleton for contact with the ground surface. Also, our choice of possible contact points between the snake robot and the ground surface keeps all parts of the snake robot links from penetrating the ground surface. Sidewinding and U-shaped lateral rolling are tested for a selection of anisotropic friction properties, and we have found that the friction coefficients influence the direction of motion for these motion patterns.

The 2D and 3D model are both simulated for obstacle aided locomotion with small obstacles. Even though the obstacles are small, the snake robot does not get stuck since we define the contact between the snake robot and an obstacle as a *moving contact point*. This results in that every point on the body of the snake robot can come in contact with the obstacle and the obstacle is kept from penetrating the exoskeleton of the snake robot. We see that the simulation results compare very well for obstacle aided locomotion simulated with the 2D and the 3D model. For lateral undulation with isotropic friction (without obstacles), there is a noticeable difference between the models. However, both models move in the same direction (backward) and the qualitative motion of the snake robot is the same.

By comparing Figure 7.3 with Figure 7.16, we note that the snake robot is able to move much faster by pushing against obstacles than while moving on a planar surface with 2.5 times higher friction coefficient transversal to the snake robot than along it. To obtain the same speed on a planar surface without obstacles, the snake robot would probably have to be equipped with wheels which complicates its design. In addition, the use of obstacles renders the snake robot less dependent on the condition of the ground surface. This

is a desirable mobility property.

The simulations given in Section 7.3.5 show that the input-output linearizing controller proposed in Chapter 6 performs better than a conventional PD-controller for the controller gains used in that section, even when noise up to a certain level is introduced in the measurements. However, there is an issue with the choice of controller gains since it was not possible to simulate very high gains for the PD-controller due to the resulting stiff model. However, for stationary motion, the IO-controller performed better than the PD-controller even though the applied torques were almost the same in magnitude.

From the back-to-back comparisons between simulation and experimental results, we conclude that the 2D and 3D model both give a realistic picture of how a real snake robot will move, especially for obstacle aided locomotion. Many aspects of Aiko have not been included in the mathematical model, which results in noticeable differences between the simulation and experimental results. However, the goal of the modelling is not to exactly reproduce the behaviour of the specific snake robot Aiko. To achieve such an exact similarity, more elaborate system identification has to be performed and the exterior of the snake robot has to be more accurately modelled. Instead of presenting a model with such a very close resemblance to Aiko, we present models for synthesis and testing of planar and 3D motion patterns for snake robots in general.

In this chapter we employ three mathematical models: the 2D model from Chapter 3, the 3D model from Chapter 4 and the ‘robot model’ presented in Chapter 6. Each model is developed for its own specific purpose. First, the 2D model is developed to have an as simple model as possible for fast implementation and simulation of planar motion pattern. Second, the 3D model is for synthesis and testing of a combination of planar and 3D motion patterns. Third, the ‘robot model’ is developed to implement and investigate stability of state-feedback controllers. However, it is not advantageous to use for simulation, partly since it takes approximately 20 to 50 times more time to simulate the planar ‘robot model’ than it takes to simulate the 2D model. We see in this chapter that the roles of the different models are successfully fulfilled.

Chapter 8

Conclusions and Further Work

The results presented in this thesis are steps towards developing snake robots that can navigate efficiently in narrow and chaotic environments, for example, when searching for people in a collapsed building or inspecting narrow and possibly dangerous areas of industrial plants. In addition, the results can be used to better understand the working principle of biological snake locomotion.

A novel 2D mathematical model based on the framework of non-smooth dynamics and convex analysis is developed for a snake robot in a certain environment with external objects. The model describes planar snake robot motion where the snake robot can push against external obstacles over the entire surface of each link. Hence, the effect of the contact forces between the obstacle and each link is modelled properly. It is shown how to easily incorporate the normal contact and friction forces into the system dynamics (which is represented by a certain equality of measures) by set-valued force laws. Moreover, it is shown how to include the correct directional Coulomb friction.

The 2D model is extended to a 3D non-smooth mathematical model based on the same methodology. However, for the 3D model we need to add the ability to lift the links above the ground. Moreover, a 3D surface has to be defined for each link. Such a surface is used for contact with the ground surface and obstacles. This allows us to simulate 3D motion patterns such as sidewinding and lateral rolling. Even though the resulting dynamics is hybrid (there may be velocity jumps due to impacts), there is no need for an explicit switch between system equations (when for example

an impact between the snake robot and an obstacle occurs) since both the non-impulsive forces and the impact impulses are covered by the same force law and since we use the time-stepping method for the numerical treatment. The same principles apply for the 2D model. The use of the non-minimal absolute coordinates results in a constant and diagonal mass matrix and an effective way of writing the system equations. Such a constant mass matrix is beneficial in the numerical treatment since it needs only to be inverted once and not for each time-step during simulation.

A control law for tracking control of the joints of a snake robot without wheels is proposed. Since the friction between the snake robot and the ground surface is essential for locomotion for wheel-less snake robots, a kinematic model is not sufficient and a non-linear model including both the kinematics and dynamics of a planar snake robot and its contact with the ground surface is presented. To this end, we develop a planar model of the snake robot based on the standard dynamics of a robot manipulator. This model is denoted the process plant model or ‘robot model’ and it is initially implemented with a smooth approximation of Coulomb friction. Such a formulation is advantageous for control since it allows us to employ the large variety of control techniques available for robot manipulators. Furthermore, a control plant model is developed from the process plant model. This model allows for an infinite number of rotations of the snake robot without affecting the controller. Based on this control plant model, a non-linear controller is developed using input-output linearization. It is proved using Lyapunov stability theory that this controller asymptotically stabilizes the desired motion for the snake robot joints. Furthermore, simulations with this controller applied to the process plant model suggest that the closed-loop system is able to track the desired joint angles. In addition, the same process plant model is implemented with a set-valued force law for describing the Coulomb friction instead of its smooth approximation. The simulations suggest that the proposed controller is also effective for the snake robot model with set-valued Coulomb friction. Moreover, it took a very long time to simulate the process plant model compared to the 2D model, and this suggests that it is not suitable for simulation.

Experiments are performed with the snake robot ‘Aiko’ built at the NTNU/SINTEF Advanced Robotics Laboratory and a back-to-back comparison between simulation and experimental results is presented. The comparison illustrate a quantitative similarity between motion of the 2D and 3D model and the snake robot ‘Aiko’ during obstacle aided locomotion. Moreover, the results show that the model needs to be extended to accurately describe in particular 3D motion of Aiko. We are able to make a more

exact model of Aiko, and thus deal with issues such as actuator dynamics and a more accurate description of the exoskeleton of Aiko. However, this will require a considerable amount of time and effort and such a specific model is beyond the scope of this thesis. Instead, we have focused on giving a more general description of a snake robot where Aiko has been employed as an inspiration for the design. The simulation results of the 2D and 3D model illustrate that qualitatively the motions of the simulated snake robot and Aiko are the same. Hence, the 2D and 3D models constitute a test bed for future controller design and motion pattern design.

This thesis provides a framework based on non-smooth dynamics for the development, analysis and testing of forthcoming motion planning and control approaches to obstacle aided locomotion. Further work on snake robots is important to eventually develop a snake robot that can navigate by itself through an unknown and cluttered environment. In the following we list a number of challenges to overcome before such a scenario is realizable. 1) Develop motion patterns based on the mathematical model and observations of real snakes that enable a snake robot to detect and exploit obstacles by itself for efficient locomotion. Such a motion pattern will enable a snake robot to move within, for example, a shattered building while looking for people. 2) Extend the 3D model to include other ground shapes such as stairs. This will be advantageous for the development of the motion pattern mentioned in '1)'. 3) Construct a smooth exterior of the snake robot. This will enable the snake robot to easily slide beside or over any irregularities on the ground. 4) Construct robust snake robots with powerful and lightweight joint actuators to be able to continue being operational even though some of the joint actuators break down. 5) Develop a snake robot exterior that can withstand large temperatures during, for example, a fire-fighting operation, while still keeping the temperature within the snake robot relatively low. 6) Develop control algorithms that consider the global positions of all parts of a snake robot. This will help keep the snake robot from unintentionally colliding with its environment during operations in fragile environments.

A snake robot has limited payload capability, poor power efficiency and a high number of degrees of freedom. Nevertheless, the snake robot has the potential of great traversability and capability of inspecting narrow places. It can also be made very robust to dirt and dust by covering the robot completely with a shell. These properties make the research on snake robots worth-while. It is hoped that this thesis will help to promote further research on the fascinating topic of snake robots.

Bibliography

- Alart, P. and A. Curnier (1991). A mixed formulation for frictional contact problems prone to Newton like solution methods. *Comput. Method. Appl. M.* **92**, 353–375.
- Ayers, J., C. Wilbur and C. Olcott (2000). Lamprey robots. In: *Proc. Int. Symp. Aqua Biomechanisms*.
- Bauchot, R. (1994). *Snakes: A Natural History*. Sterling Publishing Company.
- Bayraktaroglu, Z. Y., A. Kilicarslan, A. Kuzucu, V. Hugel and P. Blazevic (2006). Design and control of biologically inspired wheel-less snake-like robot. In: *Proc. IEEE/RAS-EMBS Int. Conf. Biomedical Robotics and Biomechatronics*. pp. 1001–1006.
- 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.
- Bayraktaroglu, Z. Y., F. Butel, P. Blazevic and V. Pasqui (1999). A geometrical approach to the trajectory planning of a snake-like mechanism. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. pp. 1322–1327.
- Bloch, A. M., J. Baillieul, P. Crouch and J. Marsden (2003). *Nonholonomic Mechanics and Control*. Springer-Verlag.
- Bloch, A. M., P. S. Krishnaprasad, J. E. Marsden and R. M. Murray (1995). Nonholonomic mechanical systems with symmetry. Technical report. California Institute of 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.

- Brogliato, B. (1999). *Nonsmooth Mechanics*. 2 ed. Springer. London.
- Brovoll, Sverre (2006). Design and implementation of a snake robot actuated by dc-motors. Master's thesis. Norwegian University of Science and Technology. In Norwegian.
- Burdick, J. W., J. Radford and G. S. Chirikjian (1993). A 'sidewinding' locomotion gait for hyper-redundant robots. In: *Proc. IEEE Int. Conf. Robotics and Automation*. pp. 101–106.
- Chen, L., Y. Wang, S. Ma and B. Li (2004). Studies on lateral rolling locomotion of a snake robot. In: *Proc. IEEE Int. Conf. Robotics and Automation*. pp. 5070–5074.
- Chernousko, F. (2005). Modelling of snake-like locomotion. *Appl. Math. Comput.* **164**(2), 415–434.
- Chernousko, F. L. (2000). Snake-like motions of multibody systems over a rough plane. In: *Proc. 2nd Int. Conf. Control of Oscillations and Chaos*. pp. 321–326.
- Chirikjian, G. 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 (1991). Kinematics of hyper-redundant robot locomotion with applications to grasping. In: *Proc. IEEE Int. Conf. Robotics and Automation*. pp. 720–725.
- Chirikjian, G. S. and J. W. Burdick (1993). Design and experiments with a 30 DOF robot. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Vol. 3. pp. 113–119.
- Chirikjian, G. S. and J. W. Burdick (1994a). A hyper-redundant manipulator. *IEEE Robotics & Automation Magazine* **1**(4), 22–29.
- Chirikjian, G. S. and J. W. Burdick (1994b). A modal approach to hyper-redundant manipulator kinematics. *IEEE Trans. Robot. Autom.* **10**(3), 343–354.
- Chirikjian, G. S. and J. W. Burdick (1995). The kinematics of hyper-redundant robot locomotion. *IEEE Trans. Robot. Autom.* **11**(6), 781–793.

- Choset, H. (2007). Snake robots at Carnegie Mellon University. <http://www.snakerobot.com/>. Online. Accessed 29 September 2007.
- Choset, H. and J. Y. Lee (2001). Sensor-based construction of a retract-like structure for a planar rod robot. *IEEE Trans. Robot. Autom.* **17**(4), 435–449.
- Choset, H., J. Luntz, E. Shamma, T. Rached, D. Hull and C. C. Dent (2000). Design and motion planning for serpentine robots. In: *Proc. SPIE Smart Structures and Materials: Smart Electronics and MEMS*. Vol. 3990. Newport Beach, CA, USA. pp. 148 – 155. Serpentine robots;.
- 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). Locomotion control of a snake robot with constraint force attenuation. In: *Proc. American Control Conference*. pp. 113–118.
- Do Carmo, M. P. (1976). *Differential Geometry of Curves and Surfaces*. Prentice-Hall. Englewood Cliffs, New Jersey.
- Dowling, K. (1999). Limbless locomotion: learning to crawl. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Vol. 4.
- Dowling, K. J. (1997). Limbless Locomotion. Learning to Crawl with a Snake Robot. PhD thesis. Carnegie Mellon University.
- Egeland, O. and J. T. Gravdahl (2002). *Modeling and Simulation for Automatic Control*. Marine Cybernetics. Trondheim, Norway.
- Erkmen, I., A. M. Erkmen, F. Matsuno, R. Chatterjee and T. Kamegawa (2002). Snake robots to the rescue!. *IEEE Robotics & Automation Magazine* **9**(3), 17–25.
- Fossen, T. I. (2002). *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*. Marine Cybernetics. Trondheim, Norway.
- Glocker, Ch. (2001). *Set-Valued Force Laws, Dynamics of Non-Smooth Systems*. Vol. 1 of *Lecture Notes in Applied Mechanics*. Springer-Verlag. Berlin.

- Glocker, Ch. and C. Studer (2005). Formulation and preparation for numerical evaluation of linear complementarity systems in dynamics. *Multibody System Dynamics* **13**, 447–463.
- Grabec, I. (2002). Control of a creeping snake-like robot. In: *Proc. 7th Int. Workshop on Advanced Motion Control*. pp. 526–513.
- Gray, J. (1946). The mechanism of locomotion in snakes. *J. Exp. Biol.* **23**(2), 101–120.
- 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.
- Henning, W., F. Hickman and H. Choset (1998). Motion planning for serpentine robots. In: *Proc. ASCE Space and Robotics*.
- Hirose, S. (1993). *Biologically Inspired Robots: Snake-Like Locomotors and Manipulators*. Oxford University Press. Oxford.
- Hirose, S. and A. Morishima (1990). Design and control of a mobil robot with an articulated body. *Int. J. Rob. Res.* **9**(2), 99–114.
- Hirose, S. and M. Mori (2004). Biologically inspired snake-like robots. In: *Proc. IEEE Int. Conf. Robotics and Biomimetics*. pp. 1–7.
- Hirose, S. and Y. Umetani (1976). Kinematic control of active cord mechanism with tactile sensors. In: *Proc. 2nd RoMAnSy Symp.* Warsaw. pp. 241–252.
- 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.
- Kelly, S. D. and R. M. Murray (1995). Geometric phases and robotic locomotion. *J. Robotic Systems* **12**(6), 417–431.
- Khalil, H. K. (2002). *Nonlinear Systems*. 3rd ed. Prentice Hall. International Edition.
- Kolmanovskiy, I. and N. H. McClamroch (1995). Developments in nonholonomic control problems. *IEEE Contr. Syst. Mag.* **15**(6), 20–36.

- 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.
- Le Saux, C., R. I. Leine and Ch. Glocker (2005). Dynamics of a rolling disk in the presence of dry friction. *Journal of Nonlinear Science* **15**(1), 27–61.
- Leine, R. I. and H. Nijmeijer (2004). *Dynamics and Bifurcations of Non-Smooth Mechanical Systems*. Vol. 18 of *Lecture Notes in Applied and Computational Mechanics*. Springer Verlag. Berlin.
- Lewis, M. A. and D. M. Zehnpfennig (1994). R7: A snake-like robot for 3-D visual inspection. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. Vol. 2. Munich Germany. pp. 1310–1317.
- Liljebäck, P. (2004). Modular snake-robot: Modeling, implementation and control of a modular and pressure based snake-robot. Master’s thesis. Norwegian University of Technology and Science. Trondheim, Norway. In Norwegian.
- Liljebäck, P., Ø. Stavdahl and K. Y. Pettersen (2005). Modular pneumatic snake robot: 3D modelling, implementation and control. In: *Proc. 16th IFAC World Congress*.
- Lu, Y., S. Ma, B. Li and L. Chen (2003). Ground condition sensing of a snake-like robot. In: *Proc. IEEE Int. Conf. Robotics, Intelligent Systems and Signal Processing*. Vol. 2. pp. 1075–1080.
- Ma, S. (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, S. (2001). Analysis of creeping locomotion of a snake-like robot. *Adv. Robotics* **15**(2), 205–224.
- Ma, S. and N. Tadokoro (2006). Analysis of creeping locomotion of a snake-like robot on a slope. *Autonomous Robots* **20**, 15–23.
- Ma, S., H. Araya and L. Li (2001a). Development of a creeping snake-robot. In: *Proc. IEEE Int. Symp. Computational Intelligence in Robotics and Automation*. pp. 77–82.

- Ma, S., N. Tadokoro, B. Li and K. Inoue (2003a). Analysis of creeping locomotion of a snake robot on a slope. In: *Proc. IEEE Int. Conf. Robotics and Automation*. pp. 2073–2078.
- Ma, S., W. J. Li and Y. Wang (2001b). A simulator to analyze creeping locomotion of a snake-like robot. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Vol. 4.
- Ma, S., 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, S., Y. Ohmameuda, K. Inoue and B. Li (2003b). 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.
- Mattison, C. (2002). *The Encyclopaedia of Snakes*. Cassell Paperbacks. London.
- McIsaac, K. A. and J. P. Ostrowski (1999). A geometric approach to anguilliform locomotion: Modelling of an underwater eel robot. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Vol. 4. pp. 2843–2848.
- McIsaac, K. A. and J. P. Ostrowski (2000). Motion planning for dynamic eel-like robots. In: *Proc IEEE Int. Conf. Robotics and Automation*. Vol. 2. pp. 1695–1700.
- McIsaac, K. A. and J. P. Ostrowski (2003a). A framework for steering dynamic robotic locomotion systems. *Int. J. Robot. Res.* **22**(2), 83–97.
- McIsaac, K. A. and J. P. Ostrowski (2003b). Motion planning for anguilliform locomotion. *IEEE Trans. Robot. Autom.* **19**(4), 637–625.
- Miller, G. (2002). *Neurotechnology for Biomimetic Robots*. Chap. Snake Robots for Search and Rescue, pp. 271–284. MIT Press Cambridge, MA, USA. Cambridge/London.
- Miller, G. (2007). Dr. Miller’s snake robots. <http://www.snakerobots.com/>. Online. Accessed 29 September 2007.

- Moreau, J. J. (1988). Unilateral contact and dry friction in finite freedom dynamics. In: *Non-Smooth Mechanics and Applications* (J. J. Moreau and P. D. Panagiotopoulos, Eds.). Vol. 302 of *CISM Courses and Lectures*. pp. 1–82. Springer Verlag. Wien.
- 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.
- Murray, R. M., Z. Li and S. S. Sastry (1994). *A Mathematical Introduction to Robotic Manipulation*. 1st ed. CRC Press. Florida, USA.
- Nijmeijer, H. and A. van der Schaft (1990). *Nonlinear Dynamical Control Systems*. Springer-Verlag. New York.
- Nilsson, M. (1997a). Ripple and roll: Slip-free snake robot locomotion. In: *Proc. Mechatronical Computer Systems for Perception and Action*. Piza, Italy.
- Nilsson, M. (1997b). Snake robot free climbing. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Vol. 4. pp. 3415–3420.
- Nilsson, M. (1998). Snake robot – free climbing. *IEEE Contr. Syst. Mag.* **18**(1), 21–26.
- Nilsson, M. (2004). Serpentine locomotion on surfaces with uniform friction. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. pp. 1751–1755.
- Ohmameuda, Y. and S. Ma (2002). Control of a 3-dimensional snake-like robot for analysis of sinus-lifting motion. In: *Proc. 41st SICE Annual Conference*. Vol. 3. pp. 1487–1491.
- 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 (1996). Gait kinematics for a serpentine robot. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Vol. 2. pp. 1294–1299.
- Ostrowski, J. and J. Burdick (1998). The geometric mechanics of undulatory robotic locomotion. *Int. J. Robot. Res.* **17**(7), 683–701.

- 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.
- 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.
- Rockafellar, R. T. (1970). *Convex Analysis*. Princeton Landmarks in Mathematics. Princeton University Press. Princeton, New Jersey.
- Saito, M., M. Fukaya and T. Iwasaki (2002). Serpentine locomotion with robotic snakes. *IEEE Contr. Syst. Mag.* **22**(1), 64–81.
- Shan, Y. and Y. Koren (1993). Design and motion planning of a mechanical snake. *IEEE Trans. Syst. Man Cyb.* **23**(4), 1091–1100.
- Spong, M. W. (1996). Energy based control of a class of underactuated mechanical systems. In: *Proc. 13th IFAC World Congress*. Vol. F. pp. 431–436.
- Spong, M. W. and M. Vidyasagar (1989). *Robot Dynamics and Control*. Wiley & Sons Inc.
- Taney, I., T. Ray and A. Buller (2005). Automated evolutionary design, robustness, and adaptation of sidewinding locomotion of a simulated snake-like robot. *IEEE Trans. Robot.* **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.
- Transth, A. A. and K. Y. Pettersen (2006). Developments in snake robot modeling and locomotion. In: *Proc. IEEE Int. Conf. Control, Automation, Robotics and Vision*. Singapore. pp. 1393–1400.
- Transth, A. A. and K. Y. Pettersen (2008). A survey on snake robot modeling and locomotion. *Robotica*. Submitted.

- Transeth, A. A., N. van de Wouw, A. Pavlov, J. P. Hespanha and K. Y. Pettersen (2007a). Tracking control for snake robot joints. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. San Diego, CA, USA. pp. 3539–3546.
- Transeth, A. A., P. Liljebäck and K. Y. Pettersen (2007b). 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. pp. 2582–2589.
- Transeth, A. A., R. I. Leine, Ch. Glocker and K. Y. Pettersen (2006a). Non-smooth 3D modeling of a snake robot with external obstacles. In: *Proc. IEEE Int. Conf. Robotics and Biomimetics*. Kunming, China. pp. 1189–1196.
- Transeth, A. A., R. I. Leine, Ch. Glocker and K. Y. Pettersen (2006b). Non-smooth 3D modeling of a snake robot with frictional unilateral constraints. In: *Proc. IEEE Int. Conf. Robotics and Biomimetics*. Kunming, China. pp. 1181–1188.
- Transeth, A. A., R. I. Leine, Ch. Glocker and K. Y. Pettersen (2008a). 3D snake robot motion: Non-smooth modeling, simulations, and experiments. *IEEE Trans. Robot.* Accepted.
- Transeth, A. A., R. I. Leine, Ch. Glocker, K. Y. Pettersen and P. Liljebäck (2008b). Snake robot obstacle aided locomotion: Modeling, simulations, and experiments. *IEEE Trans. Robot.* Accepted.
- 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.
- Xinyu, L. and F. Matsuno (2003). Control of snake-like robot based on kinematic model with image sensor. In: *Proc. IEEE Int. Conf. Robotics, Intelligent Systems and Signal Processing*. Vol. 1. pp. 347–352.
- Yamada, H. and S. Hirose (2006a). Development of practical 3-dimensional active cord mechanism ACM-R4. *Journal of Robotics and Mechatronics* **18**(3), 1–7.

- Yamada, H. and S. Hirose (2006b). Study on the 3D shape of active cord mechanism. In: *Proc. IEEE Int. Conf. Robotics and Automation*. pp. 2890–2895.
- Yamada, H., S. Chigisaki, M. Mori, K. Takita, K. Ogami and S. Hirose (2005). Development of amphibious snake-like robot ACM-R5. In: *Proc. 36th Int. Symp. Robotics*.
- Ye, C., S. Ma, B. Li and Y. Wang (2004a). 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 and Y. Wang (2004b). Twist-related locomotion of a 3D snake-like robot. In: *Proc. IEEE Int. Conf. Robotics and Biomimetics*. pp. 589–594.

Appendix A

An additional Proof and a Theorem for Chapter 6

This appendix includes a theorem and a proof referred to in Chapter 6. First, we include a theorem from Khalil (2002). Then we prove that \mathbf{M}_V in (6.102) on page 127 is positive definite.

A.1 Theorem on Boundedness

Theorem 4.18 on page 172 in Khalil (2002) is employed in Chapter 6 to show boundedness of the variable \mathbf{w} . The remainder of this section is an excerpt from Khalil (2002) and describes the theorem. We include it in this thesis for the convenience of the reader. Consider the system

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}), \tag{A.1}$$

where $\mathbf{f} : [0, \infty) \times D \rightarrow \mathbb{R}^m$ is piecewise continuous in t and locally Lipschitz in \mathbf{x} on $[0, \infty) \times D$, and $D \subset \mathbb{R}^m$ is a domain that contains the origin.

Theorem A.1 (Theorem 4.18 in Khalil (2002)) *Let $D \subset \mathbb{R}^m$ be a domain that contains the origin and $V : [0, \infty) \times D \rightarrow \mathbb{R}$ be a continuously differentiable function such that*

$$\alpha_1(\|\mathbf{x}\|) \leq V(t, \mathbf{x}) \leq \alpha_2(\|\mathbf{x}\|) \tag{A.2}$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial \mathbf{x}} \mathbf{f}(t, \mathbf{x}) \leq -W_3(\mathbf{x}), \quad \forall \|\mathbf{x}\| \geq \mu > 0 \tag{A.3}$$

$\forall t \geq 0$ and $\forall \mathbf{x} \in D$, where α_1 and α_2 are class \mathcal{K} functions and $W_3(\mathbf{x})$ is a continuous positive definite function. Take $r > 0$ such that $B_r \subset D$ and

suppose that

$$\mu < \alpha_2^{-1}(\alpha_1(r)). \quad (\text{A.4})$$

Then there exists a class \mathcal{KL} function β and for every initial state $\mathbf{x}(t_0)$, satisfying $\|\mathbf{x}(t_0)\| \leq \alpha_2^{-1}(\alpha_1(r))$, there is $T \geq 0$ (dependent on $\mathbf{x}(t_0)$ and μ) such that the solution of (A.1) satisfies

$$\|\mathbf{x}(t)\| \leq \beta(\|\mathbf{x}(t_0)\|, t - t_0), \quad t_0 \leq t \leq t_0 + T \quad (\text{A.5})$$

$$\|\mathbf{x}(t)\| \leq \alpha_2^{-1}(\alpha_1(\mu)), \quad t \geq t_0 + T. \quad (\text{A.6})$$

Moreover, if $D = \mathbb{R}^m$ and α_1 belongs to class \mathcal{K}_∞ , then (A.5) and (A.6) hold for any initial state $\mathbf{x}(t_0)$, with no restrictions on how large μ is.

Note that (A.5) and (A.6) show that $\mathbf{x}(t)$ is uniformly bounded for all $t \geq t_0$ and uniformly ultimately bounded with the ultimate bound $\alpha_1^{-1}(\alpha_2(\mu))$.

A.2 Positive Definiteness of \mathbf{M}_V

In this section we will show that $\mathbf{M}_V(\phi)$ in (6.102) on page 127 is positive definite. We see from (6.101)-(6.102) in Chapter 6 that

$$\mathbf{M}_V(\phi) = \sum_{i=3}^n \mathbf{M}_{V_i}(\phi), \quad (\text{A.7})$$

where

$$\mathbf{M}_{V_i}(\phi) = \sum_{\xi=R,F} \left((\mathbf{J}_{v_{\xi i}})^T \mathbf{R}_{B_i}^{B_3} \frac{F_{N_i}}{2} \mathbf{D}_{\mu_V} \left(\mathbf{R}_{B_i}^{B_3} \right)^T \mathbf{J}_{v_{\xi i}} \right), \quad (\text{A.8})$$

and

$$\mathbf{D}_{\mu_V} = \begin{bmatrix} \mu_{V_x} & 0 \\ 0 & \mu_{V_y} \end{bmatrix}. \quad (\text{A.9})$$

Consider the chain of massless rods with body-fixed frames $B_i = \{O_i, \mathbf{e}_x^{B_i}, \mathbf{e}_y^{B_i}\}$ attached to the rear part of each rod. The chain may move freely in the horizontal plane. Assume that two point masses (i.e. without moment of inertia) are fixed to rod i such that they do not coincide. A structure with two such rods is depicted in Figure A.1 where we notice that the first rod is number 3 due to the inertial position and orientation of the chained structure. Denote the centre of the inertial frame

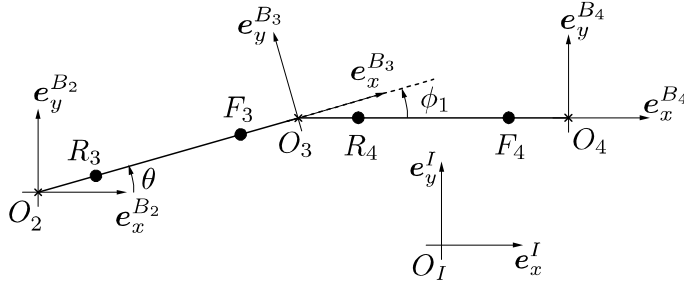


Figure A.1: Chain of two massless rods with four point masses fixed to the rods.

$I = \{O_I, e_x^I, e_y^I\}$ as O_I . Assume that the mass matrix of each point mass connected to rod i is

$$\mathbf{M}_i = \frac{F_{N_i}}{2} \mathbf{D}_{\mu\nu}. \quad (\text{A.10})$$

Hence, each point mass has more kinetic energy when moving longitudinal to the rod than transversal to it. An analogy to such a property is the added mass matrix for a completely submerged rigid body with a particular shape (see Fossen (2002) for more on added mass). Now we will proceed to develop the complete mass matrix for the chain of rods and its kinetic energy.

Define the generalized coordinates for the system of connected masses as

$$\mathbf{q} = [x_{O_2} \quad y_{O_2} \quad \theta \quad \phi_1 \quad \cdots \quad \phi_{n-3}]^T \in \mathbb{R}^n, \quad (\text{A.11})$$

where $O_2 = (x_{O_2}, y_{O_2})$ is the position of the rearmost part of rod 3 which coincides the origin of frame B_2 ; θ is the anti-clockwise rotation of frame B_3 with respect to the inertial frame, and ϕ_j , $j = 1, \dots, n-3$, are the relative angles between the adjacent rods. Let ${}_{B_3}\mathbf{v}_{O_2}$ be the translational velocity of O_2 given in frame B_3 . Let the translational velocities of the front and rear mass of rod i expressed in frame B_3 be

$${}_{B_3}\mathbf{v}_{F_i} = \mathbf{J}_{v_{F_i}}\boldsymbol{\nu}, \quad {}_{B_3}\mathbf{v}_{R_i} = \mathbf{J}_{v_{R_i}}\boldsymbol{\nu}, \quad (\text{A.12})$$

where

$$\boldsymbol{\nu} = \begin{bmatrix} {}_{B_3}\mathbf{v}_{O_2} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix}. \quad (\text{A.13})$$

Moreover, $\mathbf{J}_{v_{F_i}} \in \mathbb{R}^{2 \times n}$ and $\mathbf{J}_{v_{R_i}} \in \mathbb{R}^{2 \times n}$ are equal to the Jacobians (6.32)-(6.33) by defining the rods to be of the same length as the links of the snake

robot presented in Chapter 6 and that the point masses in this section are placed at the same points as the contact points with the ground defined for the links in Chapter 6. The velocities of the point masses are expressed from (A.12) in their corresponding frames as

$${}_{B_i}\mathbf{v}_{F_i} = \left(\mathbf{R}_{B_i}^{B_3}\right)^T {}_{B_3}\mathbf{v}_{F_i}, \quad {}_{B_i}\mathbf{v}_{R_i} = \left(\mathbf{R}_{B_i}^{B_3}\right)^T {}_{B_3}\mathbf{v}_{R_i}, \quad (\text{A.14})$$

where $\mathbf{R}_{B_i}^{B_3}$ is the rotation matrix that transforms a velocity vector from being described in frame B_i to being described in frame B_3 by the transformation ${}_{B_3}\mathbf{v}_{F_i} = \mathbf{R}_{B_i}^{B_3} {}_{B_i}\mathbf{v}_{F_i}$. The total kinetic energy K of the chained structure of connected rods can now be found as

$$K = \sum_{i=3}^n \left(\frac{1}{2} {}_{B_i}\mathbf{v}_{F_i}^T \mathbf{M}_i {}_{B_i}\mathbf{v}_{F_i} + \frac{1}{2} {}_{B_i}\mathbf{v}_{R_i}^T \mathbf{M}_i {}_{B_i}\mathbf{v}_{R_i} \right), \quad (\text{A.15})$$

and by inserting (A.10), (A.12) and (A.14) into (A.15), we find that

$$K = \frac{1}{2} \boldsymbol{\nu}^T \underbrace{\sum_{\xi_i=R_i, F_i} \sum_{i=3}^n \left((\mathbf{J}_{v_{F_i}})^T \mathbf{R}_{B_i}^{B_3} \frac{F_{N_i}}{2} \mathbf{D}_{\mu_V} \left(\mathbf{R}_{B_i}^{B_3}\right)^T \mathbf{J}_{v_{F_i}} \right)}_{\mathbf{M}_K} \boldsymbol{\nu}. \quad (\text{A.16})$$

Since the kinetic energy of the system is always positive for $\|\boldsymbol{\nu}\| \neq 0$ we have that \mathbf{M}_K is positive definite. Moreover, by comparing \mathbf{M}_V given in (A.7) to \mathbf{M}_K indicated in (A.16) we see that they are equal. Hence, \mathbf{M}_V is also positive definite, which concludes the proof.