Doctoral theses at NTNU, 2023:405

Jostein Løwer

Snakes on a Plane: Modeling, Estimation and Locomotion for Planar Snake Robots in Cluttered Environments

Doctoral thesis

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

NTNU
Norwegian University of
Science and Technology

Jostein Løwer

Snakes on a Plane: Modeling, Estimation and Locomotion for Planar Snake Robots in Cluttered Environments

Thesis for the Degree of Philosophiae Doctor

Trondheim, December 2023

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

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

© Jostein Løwer

ISBN 978-82-326-7508-1 (printed ver.)
ISBN 978-82-326-7507-4 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

IMT-report 2023-23 -W

Doctoral theses at NTNU, 2023:405

Printed by NTNU Grafisk senter

Every great story seems to begin with a snake.

- Nicolas Cage

ABSTRACT

SNAKE robots are mechanisms designed to mimic biological snakes, and aspire to inherit the robustness and stability of their biological counterparts. As of yet, this is a largely unrealized potential. This thesis aims to explore various topics in modeling, state estimation and control pertaining to snake robot locomotion in cluttered environments. The thesis is divided into three constituent parts, each exploring one topic.

Chapter 2 concerns the design and development of the Boa snake robot, a next-generation sensor-driven snake robot research platform. In addition to embedded gyroscopes and accelerometers in each link, the Boa includes a constraint force measurement system capable of measuring the interaction forces acting between adjacent links in the body of the robot. In the Boa, this system has been improved and made less prone to mechanical failure. This chapter includes the design and development of mechanical and mechatronic parts for the robot, on-board electronics, sensor systems, and software.

Chapter 3 is dedicated to modeling and estimation leveraging the on-board sensors in the Boa snake robot. Here, we aim to show that intrinsic sensor data can be used to create meaningful estimates of the robots' interaction with its environment in the form of contact point and contact force estimates. We show how this data can also be used to create a simple model of the robots' otherwise complex dynamics. Finally, we show how the Unscented Kalman Filter (UKF) can be applied to perform real-time estimates of the manipulator Jacobian matrix during locomotion. This method is shown to outperform a similar method adapted from soft robotics in both statistical properties and execution time.

Chapter 4 is dedicated to the theory of form closure, a mathematical concept that has found widespread use in computing grasps for prehensile robots to grip objects with different geometry. In this part, we present a tutorial on the foundational theory of form closure, and expand the theory to the field of snake robots. We show how form closure can be used to identify the *form closed region* of the robots' configuration space. In this region, the robot can be modeled a fully actuated system, which is shown to be beneficial to modeling, control and locomotion.

PREFACE

Before you lies my PhD Thesis, submitted the 16$^{\text{th}}$ of June 2023, which constitutes the main part of my professional career from 2020 to 2023. In this preface, I want to leave a small memorandum for any aspiring researchers and give my heartfelt thanks to the people who made all of this possible.

I am utterly fascinated and totally bewildered by robots. From the invention of the bronze axe to the advent of steam engines, man has repeatedly discovered new ways of making ourselves redundant. The latest in this series of discoveries is that of robots. They have become an ubiquitous part of our daily lives as they vacuum our homes, mow our lawns and build our cars. On nearly all levels, robots are able to physically out-compete man in strength, durability and precision. However, after decades of intensive research, robots are nowhere near the dexterity and cognitive abilities of humans. What truly makes the study of robots fascinating is not their mechanical construction, but their ability to sense, think and act. This is quite unpoetically described by Kevin Drum:

"We're talking about cognitive abilities, not the fact that they're made of metal instead of flesh and powered by electricity instead of chicken nuggets."

Snake robots are no different. Prototype snake robot platforms have been built and studied since the 70's but the endeavor of controlling these robots is, in my opinion, still in its very infancy. What makes these robots so fascinating is that they are so unlike anything else we see in mainstream robotics. Unlike most robots that actively avoid contact with its environment, snake robots are entirely dependent on this contact for locomotion and for executing tasks. Because of this, the control of snake robots alludes to some unorthodox control methods. By furthering our understanding of how these robots work, they may some day be deployed in the field to help locate survivors in collapsed buildings, or do maintenance nuclear reactors. Only the future will tell!

I believe my motivation for this work was fueled by the same curiosity that made our primordial ancestors bang rocks together to create fire. It is not necessarily for a greater purpose, or to push the boundaries of science, but a simple and almost child-like curiosity. The inability to take "we don't know" for an answer. Honestly, I have had a lot of fun doing this PhD. Yes, it had it's ups and downs, but who wouldn't get a bit bummed when you get slaughtered by peer reviewers

or your expensive circuit board catches fire on your desk?

Over the past three years I have read books on topology and abstract algebra one day, and spent countless hours soldering in the lab another day. In a sense, this PhD has been truly interdisciplinary. I have learned so much along the way, and had the opportunity to challenge myself with something new every day. I want to encourage anyone who considers doing a PhD, but who is hesitant because they're unsure if they have the aptitude for it. With a positive attitude, a creative soul, and a hearthy sprinkle of insanity, you're already 80% of the way there.

There are many people who needs to be thanked for this work. First I would like to thank my office mates **Josef**, **Mauhing**, **Amer** and **Trond** for keeping me sane and caffeinated. A warm thanks to **Irja**, my partner-in-crime, who has always been available for a manic whiteboard session or just a friendly chat. I also want to thank all my friends and colleagues at **Studentersamfundet** for their continued love and support.

I want to give a particular thanks to three people who have been defining figures in both my work and my personal life over the past years:

My first thanks goes to **Nina**, my partner and number one fan. You never doubted me for a second, and stuck with me through the highest highs and the lowest lows. Although you may not realize it, I don't think all of this would have been possible without you. Thanks for believing in me even when I couldn't do so myself.

My second thanks goes to **Damiano**, my co-supervisor. Your energy and positive outlook has been a guiding light for me. You approach everything with a passion and an energy which is truly inspiring. Thanks for the time and effort you have put into my work, and the many smiles you have given me over the past few years!

My final and most heartfelt thanks goes to my main supervisor **Øyvind**. Our relation has truly been one of the main driving forces of my work. Every time I have stepped into your office, I have left feeling inspired, motivated and appreciated. When working with you, it always feels colloquial in its best meaning. You have taught me a lot, not only about our research but also about being human. I will genuinely miss our weekly meetings, and hope that we meet in the future as colleagues or friends.

-Jostein Løwer, June 16th, 2023

Contents

CHAPTER

ONE

INTRODUCTION

1.1 Motivation

Snake robotics remains a field of great research interest, because of the astounding locomotion capability and adaptability of their biological counterparts. Biological snakes are capable of moving efficiently in a vast array of environments [1]. They are able to move in rugged terrain such as stone runs and cluttered forest floors and propel themselves in sand and dirt; environments that are typically challenging for other creatures to traverse. Some species of snakes have even been observed to leverage their nimble and powerful bodies to climb trees [2].

In addition to their locomotor skills, snakes can utilize their bodies to perform an array of different tasks. This might include tethering themselves to objects in their environment, or gripping objects such as the members of the *Boidae* family might do when suffocating prey. These capabilities are interesting for snake robots to emulate as the robots might use these capabilities to grip and move objects, and interact with constrained bodies such as levers, valves and handles.

This serves as the inspiration to build a robot that can emulate these capabilities, as it would be able to adapt to a wide array of different tasks in vastly different environments.

Snake robots have the possibility to serve as a viable alternative to legged or wheeled robots in particularly challenging terrain or in constrained environments [3]. They have been suggested as appropriate instruments in search-and-rescue missions in dangerous and cluttered environments such as in collapsed mines or buildings, where they could navigate in small constrained spaces that are insurmountable for wheeled or legged robots. Current research also includes diagnostic or service missions in environments that are dangerous to humans, such as in nuclear reactors or high-energy physics facilities [3]. Snake robots have also

shown promise in aquatic environments where, when equipped with thrusters or propellers, they might serve as a viable alternative to traditional Unmanned Underwater Vehicles (UUVs) [4].

As of the time of writing, the potential of snake robots is largely unrealized. Several robotic platforms emulating the physical capabilities of biological snakes have been constructed [3], however the control of the robots remain the limiting factor in realizing their potential. Unlike wheeled or legged robots that operate best in homogeneous and predictable environments, snake robots might leverage cluttered or unstructured environment to improve their locomotion capabilities.

This thesis seeks to further the study into one form of snake robot locomotion in cluttered terrains called Obstacle Aided Locomotion (OAL) [5]. In the study of OAL we investigate how snake robots might leverage obstacles in its environment to produce propulsion. As snake robotics is a young field in rapid development, with many unsolved problems, we limit our study to planar snake robots that are limited in motion to only moving along a level plane. The goal of this thesis is to explore methods that can directly aid in OAL, such as control strategies, modeling and path planning, and methods that implicitly aid in OAL, such as state estimation and sensing.

Ultimately, this thesis is intended as a stepping stone towards the greater goal of creating a complete and robust OAL-based control strategy, allowing snake robots to move with the same agility as their biological counterparts in cluttered and rugged environments.

1.2 Constituent Publications and Work

The work underlying this thesis was produced in the following publications, that are numbered according to their appearance in this thesis

Journal Papers (First Author)

- Article II: **Improved Jacobian Matrix Estimation Applied to Snake Robots** (Jostein Løwer, Damiano Varagnolo and Øyvind Stavdahl)[6], Frontiers in Robotics and AI 2023

- Article IV: (*Review pending*) **Snakes On a Plane: Form Closure and Constrainedness in Planar Snake Robots** (Jostein Løwer, Irja Gravdahl, Damiano Varagnolo and Øyvind Stavdahl), IEEE Robotics and Automation Letters (RA-L) 2023

- Article V: (*Review pending*) **Undulatory Snake Robot Locomotion in Cluttered Environments Using Form Closure** (Jostein Løwer, Irja Gravdahl, Damiano Varagnolo and Øyvind Stavdahl), IEEE Robotics and Automation Letters (RA-L) 2023

Conference Papers (First Author)

- Article I: **Proprioceptive contact force and contact point estimation in a stationary snake robot** (Jostein Løwer, Irja Gravdahl, Damiano Varagnolo and Øyvind Stavdahl), [7], IFAC Symposium on Robot Control (SYROCO) 2021

- Article III: **A Novel Model for Link Dynamics in Planar Snake Robots Using Internal Constraint Force Sensing** (Jostein Løwer, Irja Gravdahl, Damiano Varagnolo and Øyvind Stavdahl), [8], IEEE Conference on Control Technology and Applications (CCTA) 2023

Other Publications, Not Included in the Thesis

In addition to the aforementioned publication, we include a publication that the author contributed to, that is not considered an essential part of this thesis or its constituent research.

- **Modeling for Hybrid Obstacle-Aided Locomotion (HOAL) of Snake Robots** (Irja Gravdahl, Øyvind Stavdahl, Atussa Koushan, Jostein Løwer and Kristin Ytterstad Pettersen) IFAC International Conference on Mathematical Modeling (MATHMOD), 2023

Supplementary Material

Chapter 2, which concerns the design and construction of a novel snake robot, is accompanied by a closed repository containing additional information on the software and hardware designs for the snake robot [9].

1.3 The Structure of the Thesis

The PhD research that comprises this thesis visited a number of different topics, which is reflected in the structure of the thesis. In this section, we discuss not only the structure of the thesis, but also how its different topics constitute one coherent work. This thesis is written as a combination of an article collection and a monograph, using the style that is deemed most appropriate for communicating the content of each section. Excluding this introductory chapter, the thesis consists of three chapters, each dedicated to one aspect of the research.

Chapter 2 is written monographically and concerns the design and development of the Boa Snake Robot, a next-generation sensor-driven snake robot research platform. This part outlines the design and development of mechanical parts, mechatronic parts, on-board electronics, sensor systems, and software for the robot. This chapter is intended as a part of the complete documentation for the Boa where the software and hardware design files can be found in the code repository accompanying this thesis [9].

Chapter 3 is dedicated to modeling and estimation leveraging the on-board sensors in the Boa snake robot, and is written as an article collection of Articles I-III. Article I aims to show how intrinsic sensor data can be used to create meaningful estimates of the snake robots' interaction with its environment in the form of contact point and contact force estimates. Article II shows how the Unscented Kalman Filter (UKF) can be applied to perform real-time estimates of the manipulator Jacobian matrix during locomotion. The method outperforms a convex optimization based method adapted from soft robotics in both statistical properties and execution time. Finally, Article III shows how intrinsic sensor data from a snake robot can be used to create a simplified model of a snake robots' otherwise complex dynamics.

Chapter 4 is dedicated to the theory of form closure, a mathematical concept that has found widespread use in computing appropriate grasps for robot manipulators to grip objects of differing geometry. Section 4.2 is written monographically as a tutorial on the foundational theory of form closure. It is intentionally written in a more colloquial style than the remainder of the thesis in order to promote intuition, interest and understanding. The tutorial is intended as a stepping stone for Article IV that expands this theory to the field of snake robots and Article V, that leverages these findings to identify a region of the robots' configuration space where it is fully actuated

The Articles are near-verbatim copies of the original manuscripts, with only minor alterations. The changes include a unified citation section for all the articles, and changes in the cross-referencing between the articles. We note that the articles were written at different times, with different themes in mind. It is likely that there are some discrepancies or overlap in notation between the different articles. We refer to the introductory section of each article for a definition of the notation used in each respective article.

1.4 Contributions

The primary contributions of this thesis are:

- The design and construction of the Boa snake robot, with particular emphasis on its sensor systems.
- A contact force and contact point estimator, derived from the findings in [4]. This includes experimental results verifying its efficacy in the stationary case.
- The use of the Unscented Kalman Filter as a means of estimating the Jacobian matrix in a snake robot.
- A simple and partially linear model of a snake robots' dynamics, leveraging the sensor information available in the Boa.
- The adaptation of form closure to snake robot locomotion, and the consequent use of form closure to identify regions of the robots' configuration space that are beneficial to locomotion.
- The concept of form boundedness: a generalization of form closure that exhibits many of the same properties, but under relaxed conditions.
- A demonstration showing that form closure and form boundedness can be used as conditions to ensure predictable lateral undulation in cluttered environments.

CHAPTER

TWO

DESIGN OF THE BOA SNAKE ROBOT

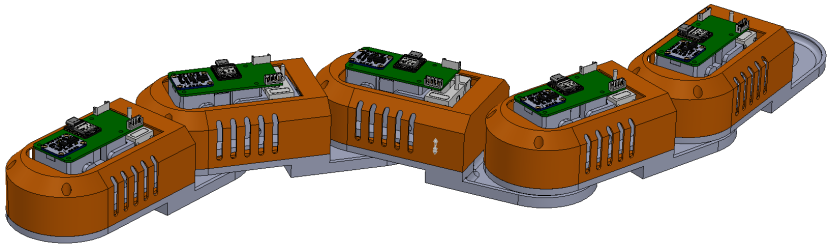


Figure 2.1: A Boa snake robot with $N = 5$ links.

2.1 Developmental Background and Pedigree

The Boa snake robot is built to actively engage in research tied to the concept of Obstacle Aided Locomotion (OAL) [5]. The previous iteration of snake robots built at the Department of Engineering Cybernetics (ITK), named Mamba [4], was built in the time span from 2010-2013. The design and construction of its predecessors are reviewed in [10].

The Mamba snake robot featured a novel measurement system, intended to measure the constraint forces between consecutive links in the robot, in order to estimate the contact forces between the robot and its environment without the use of external sensors. The underlying theory of this estimation was first conceived in [4], and is further elaborated on in Article I.

Previous attempts have been made at using external sensors (c.f. Article I), but the resulting mechanisms have have often proved to be too fragile, as the snake robots' exterior is subject to wear and large forces when interacting with the environment.

Preliminary research in the beginning stages of the PhD work [10] indicated that the sensor system in the Mamba snake robot did not perform as intended. The sensors were fragile and prone to mechanical warping and hysteresis, making it difficult to produce meaningful constraint force measurements from the Mamba.

As of 2023, the Mamba has been an active research platform for 11 years. A large number of the parts used in the construction of the Mamba are no longer commercially available, and the last functional parts are becoming weathered and are likely to break in the near future.

The fragility of the sensor system, the aging of the Mamba and other minor issues related to control software and servoing led to the decision to develop a novel snake robot platform specially designed for OAL. The Boa snake robot is the result of this development. Boa is built from the ground up, taking heavy inspiration from previous snake robots at ITK and other research institutions [3]. The Boa robot differs from previous robots at ITK in three major ways:

- It is planar: While previous robots were designed to move in three dimensions, the Boa is intentionally designed to only operate in two dimensions. This implies that the Boa cannot climb obstacles or lift its head off the ground. Being planar also limits which gaits the Boa can perform to produce motion. Lifting parts of the robot off the ground requires a large amount of motor torque, and a high level of structural integrity, while keeping the weight of the links low. Issues such as weight and structural strength are comparably smaller issues when working only on a plane. As the robot is planar, its top side can be uncovered, making it easier to access electronics, sensors and motors. The uncovered top-side also makes it easier to provide sufficient cooling to the motors, as overheating was a significant issue in the Mamba snake robot [10]. By making the robot planar, less time was spent on the design and construction of the robot, making it possible to focus on conducting experiments and obtaining valuable data for this thesis.

- The Boa snake robot is specifically designed for OAL. Previous snake robots at ITK were complex mechanisms designed to do a vast array of different tasks. Many of these robots had different detachable modules, making it possible to e.g. mount wheels or tracks on the snake robots. The Mamba was, initially, designed to be waterproof, allowing it to be submerged in order to study snake locomotion in water. All though these robots were designed to perform many different tasks, they arguably suffered in performance in that all this functionality had to be accommodated in a single form factor. The Boa is specifically designed for research into OAL, making it as capable as possible in this single task.

- Boa includes a similar constraint force estimation system as the one included in the Mamba. The design flaws of the Mamba were taken into consideration when developing Boa, in order to create a system capable of accurately measuring constraint forces. Preliminary research (c.f. Article I), indicates that the sensor system on the Boa works as intended.

The following sections describe the design and development of the Boa snake robot in detail. All schematics, design files and software relating to the Boa can be found in a repository [9], and we will refer to the repository when necessary.

2.2 Design Principles

At the beginning of the development of the Boa snake robot, a set of design principles were formed to serve as guidelines during the development process. The design principles are specified in no particular order, and have no strict prioritization. It is important to note that these are not strict rules, but rather guidelines intended to facilitate in decision-making during the design of the robot and its software.

Principle 1: Commercial Off-The-Shelf (COTS)

The Boa snake robot should be built from commercially available parts, with as few custom-built solutions as possible.

Custom solutions are often costly, both in development time and in expenses. In the case that a custom built solution breaks, it is oftentimes difficult to repair, and spare parts can be difficult to acquire. An example of the implementation of this design criterion can be seen in the choice of off-the-shelf modules for the onboard IMU and microcontroller in each link. Where it is possible, most parts for the Boa have been bought from the same main suppliers, making it easy to acquire spare parts in the case that the robot is broken or in need of maintenance.

Principle 2: User Friendliness

Using the Boa snake robot for research purposes should be fairly easy for anyone with only a bare minimum of experience with high-level programming.

Consequently, it should be easy for more advanced programmers and engineers to improve or modify both the hardware and software running on the Boa. Ideally, a user with no previous experience should be able to use the Boa within a single workday. An example of the implementation of this design criterion is that the control and sensors electronics in the Boa is built around the Arduino and Raspberry Pi platforms, both of which should be familiar to most students from a computer science, electrical engineering or robotics background. Programming and flashing new firmware onto the links is done via a USB-C port on the link main board, and requires no external debugger or programmer.

Principle 3: Maintainable, Modular and Upgradeable

Replacing the mechanical parts of the Boa should be easy, whether that be for maintenance or upgrades. The Boa should also be easy to assemble and disassemble.

An example of the implementation of this design criterion is the choice of screws for the robot. The robot can be assembled and disassembled with a stan-

dard set of metric allen keys. Two links can be detached by unplugging a single cable and removing a single screw. All non-structural parts of the Boa are 3D-printed and can easily be replaced. These parts are designed so that they can be printed on a consumer-grade low-cost FDM 3D-printer without the need for supports in the print itself. All the structural parts of the Boa are designed to be manufactured by a 3D-printer or by a 4-axis CNC milling machine with practically no alterations.

Principle 4: Future Proof

The latest hardware, firmware and software should be used in all stages of development to prevent the robot from becoming outdated in the near future.

As an example, the top-level software for the Boa is written for Python 3.11 which is the latest release of the programming language as of June 2023. The Boa uses the latest and most powerful version of the Raspberry Pi and a newly released microcontroller board for the sensor systems in the links.

2.3 Design and Construction

The following section discusses the design and construction of the Boa snake robot. Each subsection concerns a different aspect of the Boa, and details the design choices made during the development of the robot.

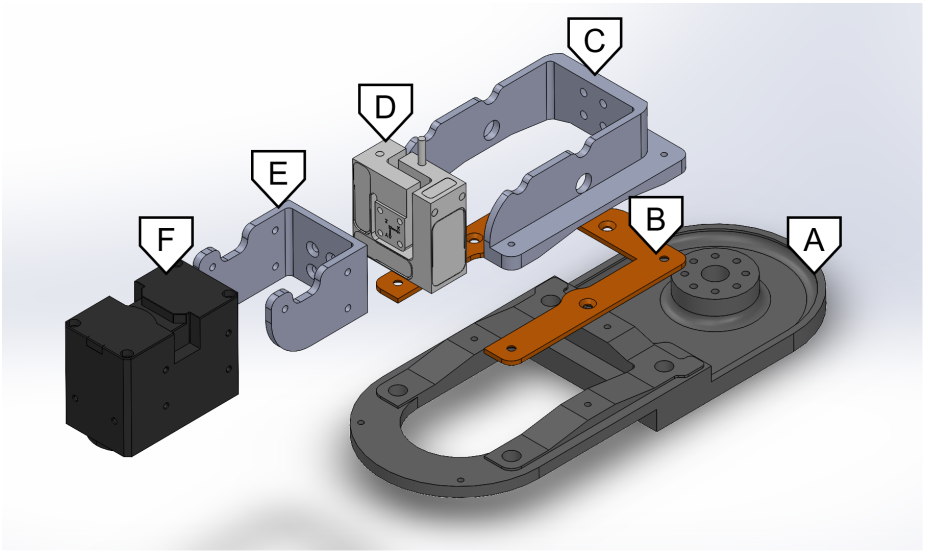


Figure 2.2: An exploded view of all the structural parts of one of the body links. The parts are (A) Chassis, (B) Spring, (C) Bracket, (D) Load Cell, (E) Servo Mount and (F) Servo.

2.3.1 Mechanical Construction

The main mechanical structure of the Boa consists of *links* and *joints*. A link is one of the N rigid body elements that constitute the Boa snake robot and the $N-1$ joints are the mechanisms that interconnect each of the links. All the links on the Boa are mechanically identical except for two of the links, namely the *head link*, which is indexed as link N and the *tail link* which is indexed as link 1. The remainder of the links are termed the *body links*. From a control and sensing perspective, the head and tail links serve special purposes in the Boa, but from a mechanical perspective they are near identical to the body links

One core feature of the Boa is that it only moves in two spacial dimensions along a plane and cannot lift its body from the surface. Accordingly, all the joint axes have the same orientation which is normal to the plane. This implies that the Boa is built to interact with its environment only at the bottom and at the sides. The top of the Boa is not intended to interact with the environment, which makes it a suitable place to put sensitive electronics, wiring and brittle mechanical components.

The Boa is built so that its main structural component is also the component that interacts with its environment. This is as opposed to a design using structural chassis that bears the mechanical load of the system with an outer "shell" that interacts with the environment.

The constituent parts of the Boa can be divided into two categories: The *structural parts* and the *non-structural parts*. The structural parts are designed to efficiently transfer force, and are essential to the mechanical workings of the Boa. The non-structural part serve other purposes such as dust protection or coverage.

The structural parts are in general designed so that they can be manufactured using an array of different manufacturing methods and can be built from a range of materials. All of these parts can be manufactured on a 4-axis CNC milling machine. In such a process, the parts can be manufactured from a range of materials including, but not limited to, stainless steel, aluminum and acetal copolymer (POM-C). In addition the structural parts can be produced by 3D-printing using one of the following methods:

- SLS or MJF printing for production in nylon or other synthetic polymers
- FDM printing for production in plastics
- SLA printing for production in resin compounds
- SLM printing for production in steel, titanium or a range of exotic metal alloys.

The assembly of an entire body link is shown in Figure 2.3, and an exploded view of all the structural parts is shown in Figure 2.2. The main structural part of the Boa is the *chassis*. The chassis is unique in that it is the only part intended to mechanically interact with its environment, which it does at its bottom and on

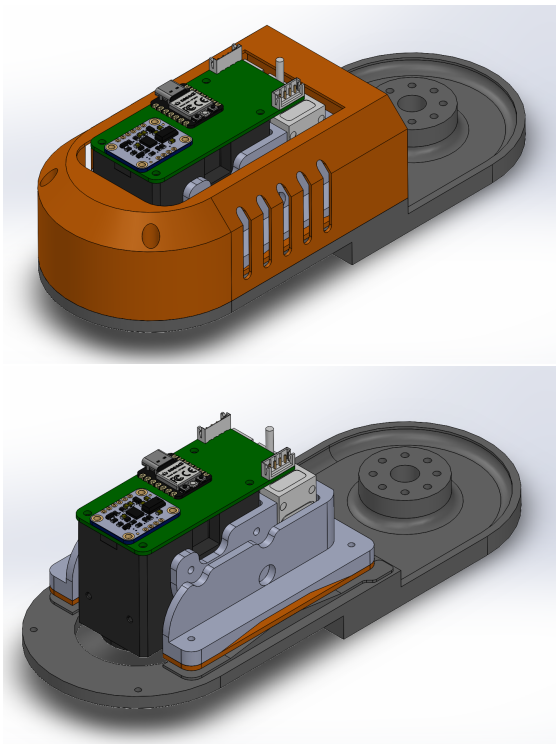


Figure 2.3: An overview image of a single link of the Boa snake robot with and without the link cover, showing the structural parts.

its sides. As the part is integral to the structure of the robot, and interacts with the environment, it was essential that a material was chosen that is both rigid to prevent unintended deformation of the link, and with a low friction coefficient as to not hinder locomotion. The choice fell on POM-C as it satisfies both criteria.

The *spring* and *bracket* together with the chassis form a *beam coupling* and is shown in detail in Figure 2.4. A beam coupling as shown in Figure 2.5, is a compliant mechanism designed to interlink two driveshafts, which due to mechanical inaccuracies or deformations, might not be parallel or co-linear. A single beam coupling is rigid in three degrees of freedom, effectively transferring axial torque and radial forces, while being deformable with respect to axial force and radial torques. Typically two beam couplings are interconnected as shown in Figure 2.5, introducing another degree of freedom also allowing deformation wrt. radial forces.

The beam coupling was included in the design of the Boa to account for inaccuracies in the construction of the robot or its environment. If the surface that the robot rests on is not completely flat and the robot is placed on the surface without any compliance in its body, this could lead to high internal stress in the robot, potentially deforming or breaking its structural components. The beam coupling on the Boa is designed to be rigid when torque is applied about the

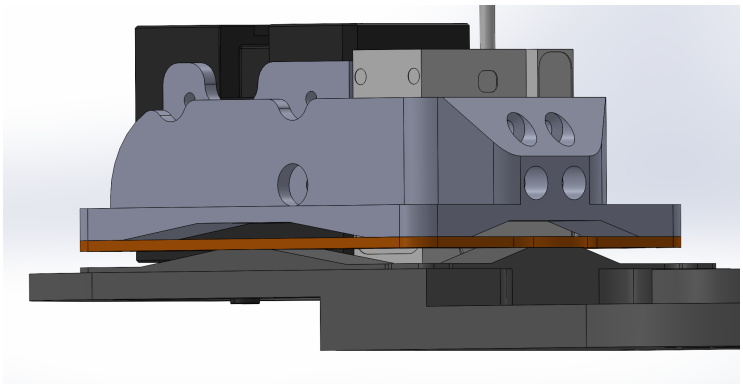


Figure 2.4: An exploded view of all the structural parts of a body link.

Figure 2.5: An industrial grade double beam coupling. (© Regal Rexnord).

joint axis, or when a force is applied normal to the joint axis. Consequently, the beam coupling is compliant when it is subjected to torques normal to the motor axis or forces along the joint axis.

The *load cell* is an off-the-shelf *ME-Messysteme K3D40-50N 3-axis force sensor* that interconnects the bracket and the servo mount. The load cell is intended to measure the interaction forces between the link it is mounted on and the link it is connected to through the servo. The sensor properties of the load cell is covered in greater detail in Section 2.3.4.

The servo mount serves as an interconnecting part between the servo and the load cell and serves no other functional purpose. Both the servo mount and the bracket are intended to be manufactured from either aluminum or steel, as the parts require a high tensile strength in order to not deform under load.

2.3.2 Actuation

Every joint on the Boa is actuated by a *Dynamixel XH540-V150-R* smart servo motor as shown in Figure 2.6. The servo was chosen due to a number of qualities:

- The servo can be powered by a 24 V power bus; a voltage level that is common in commercial robotics and automation engineering, making it simple

Figure 2.6: The Dynamixel XH540-V150-R. (© ROBOTIS Co. Ltd.).

and cheap to create a power supply with sufficient current to power all of the servos.

- The servo has a stall torque of 6.4 Nm making it sufficiently powerful to actuate the joints of the Boa snake robot.
- Although the servo has a gear exchange ratio of 1:150, the cycloidal gearbox design of the servo makes the servos backdriveable. This is a useful feature as it makes the snake robot easier to manipulate and move when doing experimental setups, as well as it protects the mechanical structure of the Boa from impacts that may happen during locomotion.
- The servo is reasonably priced compared to other servos with similar performance from other manufacturers.
- The manufacturer has made a conscious effort to make well-written an thorough documentation for all of their products.
- The servo can operate in a range of different control modes, allowing internal feedback loop control on speed, position, torque or a combination thereof.
- The servo is controlled using a standardized serial protocol over a RS-485 twisted differential pair, making it easy to control the servo and receive status and sensor information.
- It is relatively light and weighs only 165 g, adding minimal weight to the Boa.

Early iterations of the Boa snake robot used the *MyActuator RMD-X6 1:6* servo motor, but it was consciously abandoned as it was prone to failure, was poorly documented, and it did not perform as specified by the manufacturer.

Each servo actuate a single joint and interconnects the servo mount on link i to the chassis of link $i+1$. The servo itself is mounted on link i and not on link $i+1$ as this removes the need for a servo motor taking up space in the head link, which would be more prudent to use for payload such as cameras, manipulators or sensors.

2.3.3 Sensor Systems

The Boa has an array of onboard sensors. This section deals only with sensors that are included in every link of the Boa, as any other sensors included on the Boa are considered as attachments rather than integral parts of the sensor system. The sensor data that can be acquired by the Boa stems from three different sources:

Servo sensor data

Each of the Dynamixel servos can acquire sensor data relating to the actuation of the robot, which can be read from its serial interface. This includes measuring current, temperature, servo power consumption, applied torque, joint angular velocity and joint angle.

IMU sensor data

An IMU (Inertial Measurement Unit) provides odometric data for each of the links. This includes linear acceleration, gyroscopic measurements, and magnetic field readings. The specific sensor used in the Boa is an *Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout* that builds on the *Bosch BNO055* microchip. The Bosch BNO055 features an onboard *ARM Cortex-M0* processor that allows the IMU to execute proprietary data fusion algorithms on its different sensor readings before the data is transferred to the user, reducing some of the computational cost of using the IMU.

Constraint force measurements

Each joint contains an load cell of the type *ME-Messysteme K3D40-50N 3-axis force sensor*. This load cell allows the Boa to measure the constraint force in a joint, which in the Boa is the force exerted on link i by link $i+1$ in the reference frame of link i. The design and construction of the Constraint Force Sensor (CFS) is complex and is treated in its entirety in Section 2.3.4

2.3.4 Design of the Constraint Force Sensor (CFS)

The CFS is a measurement system onboard the Boa is intended to measure the constraint forces between two consecutive links. The CFS consists of three main components: The load cell, an instrumentation amplifier and an Analog-to-Digital Converter (ADC). In this section we will cover the general structure and design of the CFS system.

The K3D40-50N was chosen for the CFS system because of its durability, its ability to measure force on multiple axes, and its small form factor allowing it to fit inside the geometry of the Boa. One disadvantage of the K3D40-50N is that it does not contain any on-board electronics apart from the strain gauges that the load cell uses for force measurements. The load cell contains three Wheatstone bridges, each giving a differential voltage proportional to the force exerted on the sensor on the respective axis. The strain gauge bridges provide a differential voltage in the order-of-magnitude of millivolts, requiring some degree of amplification and digital conversion in order to be useful in the sensor system of the Boa.

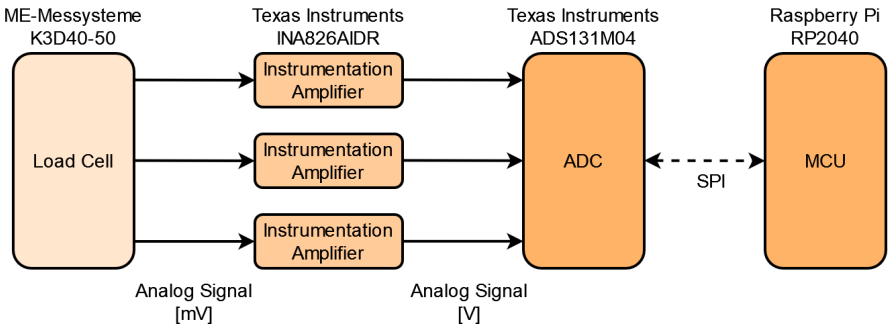


Figure 2.7: A communications diagram showing the design of the CFS subsystem.

Other commercial sensors are available that have built-in electronics in the form of amplifiers, ADCs or processing units, however these sensors were significantly more expensive than the K3D40-50N. This motivated the need to design an amplification system that can convert the small differential voltage from the strain gauge into meaningful force measurements.

The overall design of the CFS subsystem is shown in Figure 2.7. The initial step of the sensor measurement is to convert the small differential voltage from the strain gauge bridges into a higher differential voltage which can be used in an ADC-process. The *Texas Instruments INA826* instrumentation amplifier was chosen for the purpose that amplifies the signal from the load cell by a factor of around $70 : 1$ depending on the settings of the amplifier.

The amplified signal is relayed to an ADC that converts the analog signal to a digital representation. The *Texas Instruments ADS131M04* was chosen for this purpose as it uses a *Sigma-Delta* approach for the ADC conversion, giving it a resolution of 24 bits. The ADC also has four unique ADC channels allowing it to perform multiple conversions at once. This allows for simultaneous sampling of each force axis of the load cell, that is beneficial to the synchronization of the constraint force measurements.

The ADC is connected to the onboard microcontroller in each link through an SPI interface, allowing the MCU to extract sensor data and configure parameters in the ADC such as sampling rate and voltage reference levels.

2.3.5 Communication

This section deals with the communication between the links and the host computer of the Boa. The host computer can be any computer with an RS485 interface. In the case of the Boa, the host computer is not located in the snake robot itself, but is rather connected to the robot via the *umbilical*; a cable tether connected to the tail link of the Boa.

As show in Figure 2.8, each link is connected to the same RS485 bus called the *spine bus*. Each sensor module and each servo interconnects with the spine bus separately and can both transmit and receive data over the bus. The spine is the

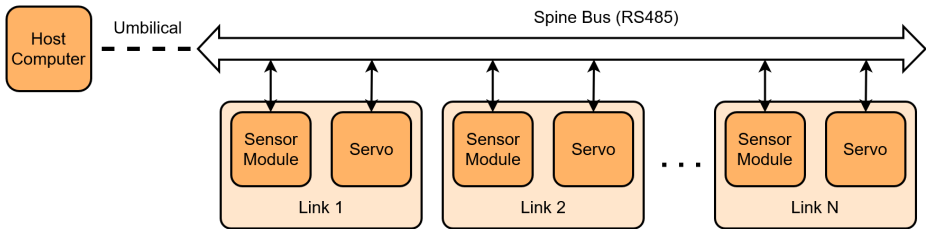


Figure 2.8: A communications diagram showing the composition of the spine bus, and how the host computer and the links attach to the bus.

primary communications channel on the Boa and allows for the host computer to actuate the servos and acquire the necessary sensor data to perform meaningful control strategies.

The bus is configured in a *master/slave* configuration, where the host computer maintains control over the arbitration on the bus at any time, and the sensor modules and servos only reply on the bus when explicitly requested to by the host computer. A typical communications flow on the spine bus is shown in Figure 2.9, where the host computer requests sensor data from the sensors and servos in turn, and sends actuation data to the servos

2.3.6 Embedded Electronics

The entirety of the sensor module is implemented on a single circuit board which is shown in Figure 2.11 whose general structure is shown in Figure 2.10. The circuit board is constructed as a 4-layer circuit board with reserved ground and supply voltage layers. The circuit boards were designed in the software *EasyEDA* and was manufactured by the company *JLCPCB* who subsequently also handled

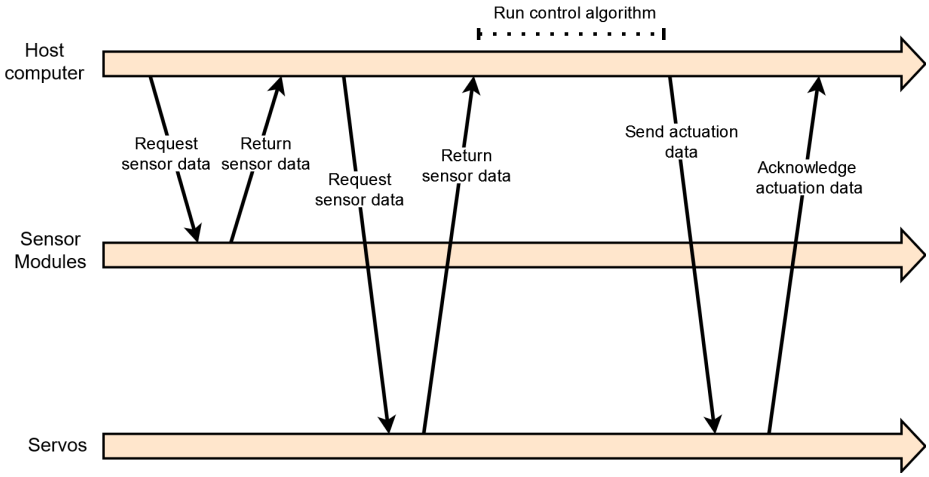


Figure 2.9: A sequence diagram showing the communication of the host computer with the sensor modules and the servos.

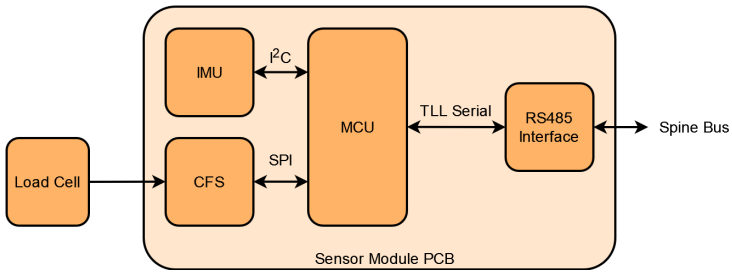


Figure 2.10: A communications diagram showing the composition of the sensor module and how it interconnects with other components.

the placement and soldering of components on the circuit board. When this thesis was written, the design of the circuit board was incomplete and its completion is considered a future work.

2.3.7 Power Electronics

The Boa is powered by a central DC power bus at 24 V as shown in Figure 2.12. The bus is powered by an external power supply that is connected to the power bus through the umbilical tethered to the tail-link of the robot.

The amount of power consumed by the Boa is largely influenced by the number of links in the robot. The servos each consume 2.4 A of current when under stall torque, and the current consumed by the sensor modules is negligible in comparison. Thus, the maximal current consumption of the Boa is approximately $2.4 \times N$ where N is the number of links.

Preliminary tests indicate that during locomotion, the Boa will likely consume less power, and that an average consumption of about 0.6 A for each servo is a reasonable metric to use when choosing the capacity of the power supply. Thus a power supply capable of delivering a continuous current of 10 A at 24 V is sufficient to power a Boa snake robot consisting of 16 links.

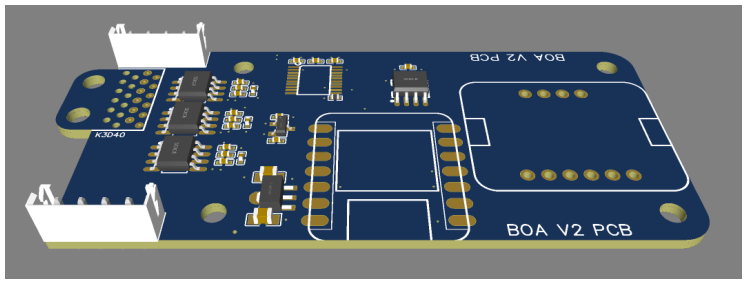


Figure 2.11: The PCB constituting the sensor module on each link in the Boa snake robot, with some of the components mounted.

2.3.8 Software and Firmware

The software and firmware for the Boa Snake robot is divided into two components: The code running on the onboard microcontrollers in each link, and the code running on the host computer. The code running on the onboard microcontrollers is referred to as the *firmware* in the context of the Boa snake robot, and the code running on the host computer is the *software*. This section covers the firmware and software only superficially, and the reader is referred to the source repositories [9] for a complete implementation.

The firmware is a low-level bare-metal code library written in C/C++ for the Arduino electronic prototyping platform. The firmware runs on the onboard Raspberry Pi RP2040 microcontrollers on each link, and coordinates the acquisition of CFS and IMU data from each link and relays it on the spine bus when requested to do so. The firmware is not intended to be modified by an end user of the Boa platform, and should ideally only be modified when adding functionality or when doing code optimization or debugging.

The functionality of the firmware is relatively simple. Because of this, a choice was made to not use a Real-Time Operating System (RTOS) as a basis for the firmware as this likely would have added additional complexity to the code base without any perceivable performance benefits. The choice of using the Arduino platform was made because of its ease-of-use and its familiarity to most engineers within the fields of computer science, robotics or electronics. While using the Arduino platform might slightly impact real-time performance for the firmware, the added benefit of increased legibility and modifiability outweighs this disadvantage.

The software is created in the form of a Python library, which provides an abstracted and high-level interface to all functions of the Boa snake robot. The library exposes the `Boa()` object, that represents a single Boa snake robot connected to the host computer. The tasks of acquiring sensor data or actuating the servos on the Boa snake robot are both done through the same object. The software builds on top of the *Dynamixel SDK* that serves as a middleware to communicate with the servos on the Boa.

The software was written in Python, again for its ease-of-use and the likelihood that Python is familiar to an end user. Python also has ample resources and libraries for numerical computation, optimization, state estimation and control, making it easier to implement control strategies for the Boa. While Python is

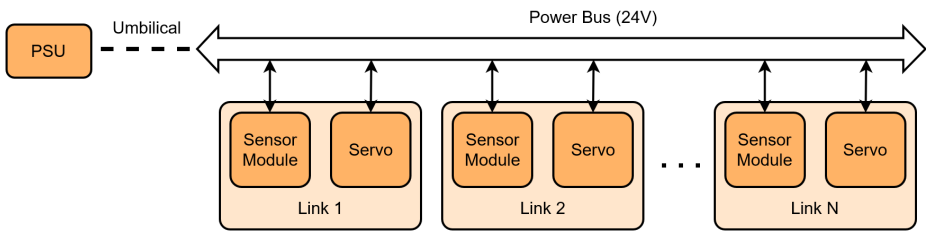


Figure 2.12: The topology of the power bus in the Boa snake robot.

flexible and simple to use, it suffers in performance, and may perform poorly in time-critical or computationally intensive tasks. Because of this, further research may be necessary to determine if Python is a suitable programming language for the Boa snake robot.

CHAPTER

THREE

CONTACT FORCE AND STATE ESTIMATION IN PLANAR SNAKE ROBOTS

3.1 Introduction

This chapter is dedicated to the study of contact force and state estimation in planar snake robots. The chapter consists of three articles, approaching different aspects of this topic. In this section, we discuss the motivation behind each of the three articles and show how the three separate works are related.

In [11], the dynamics of a snake robot is studied in detail. Even in the most forgiving case where the robot is placed in an obstacle-free environment, controlling the robot is a challenging task. When introducing obstacles into the environment, the model is further complicated by the highly non-linear, discontinuous and oftentimes unpredictable dynamics arising when the robot interacts with the obstacles. This leads directly to the motivation for this chapter: In order to create a meaningful control strategy, it is beneficial to continuously acquire as much information as possible on the state of the robot, its movement and its interaction with its environment.

The findings in all three papers build upon the same foundation: The sensor system of the Boa snake robot. The Boa features an array of sensors in each link, which makes it possible to infer information about the robots' behavior and interaction with its environment. More specifically, each link includes an IMU supplying inertial measurements for each link, and a constraint force sensor that measures the interaction force between adjacent links. The goal of this chapter is to show how these measurements might be applied in different ways to aid in the control of the robot.

Article I investigates how we might detect and measure the robots' contact with its environment. The article is based on an assumption that each link is

subject to three force components, which determine the translational dynamics of the link: The forces acting on the link from its two neighboring links in the robot, as well as an external force acting on the link from the robots surroundings The external force component comprises normal and tangential forces between the snake and its environment.

If the acceleration of the link and the constraint forces are known, then by Newtons 2^{nd} law, it should be possible to uniquely infer the contact force from the known data. The article explores this idea, but is limited in scope to only the case where the robot is near stationary in relation to the world frame. A further development of this strategy to the more general case where the robot is in motion is considered a future work, and is addressed in greater detail in Article I.

Article II is based on a recent series of papers exploring the concept of estimating the Jacobian for soft robots in real-time [12, 13]. Soft robots are notoriously hard to model, and their forwards and inverse kinematics can oftentimes be hard to compute. The papers present an estimation method in which the motion of the soft robots' end effector is measured by on-board sensors. This data is paired with the actuator inputs that caused the motion, and the Jacobian that relates the two is computed using convex optimization. The resulting Jacobian is a useful tool as it relates future actuator inputs to end effector motions without the need for an analytic kinematic model of the system

The problem of modeling soft robots may relate to the problem of modeling snake robots in constrained environments. While soft robots are difficult to model due to their complex geometry, snake robots are difficult to model due to their discontinuous contact with the environment. The core idea of Article II is to investigate whether the method for Jacobian estimation was applicable to constrained snake robots as well.

During the investigation of the Jacobian estimation method, it became apparent that the original method did not take the possibility of measurement noise into account. Because of this, Article II also suggests an alternative approach to the Jacobian estimation method from [12, 13] by using the Unscented Kalman Filter (UKF).

In comparison: Article I intends to provide estimates of the snake robots' contact with the environment. Article II intends to bypass the problem of estimating contact entirely, by rather estimating the contacts' effect on the kinematics of the snake robot.

Article III investigates how the constraint force measurements in the Boa can be used to simplify the otherwise complex modeling of snake robots. As the robot forms one long open kinematic chain, the dynamics of each link is non-linearly dependent on the dynamics of all the other links in the chain.

The core idea of Article III is that it may be possible to "disconnect" the dynamics of the separate links by inserting the constraint force measurements into the original dynamic model. In this way, parts of the model may become

near-linear, making it simpler to utilize for control or state estimation purposes.

Article III, as opposed to Article I and II, does not directly approach the problem of estimating interactions between the snake robot and its environment, but rather seeks to simplify the dynamics of the snake robot in its own right.

3.2 Article I: Proprioceptive Contact Force and Contact Point Estimation in a Stationary Snake Robot

Abstract

Measuring contact forces and knowing how and where a robot is interacting with obstacles in its environment is particularly useful for developing physics-based Obstacle-Aided Locomotion strategies for snake robots. The current paradigm for obtaining such measurements is mostly hardware-based, and is achieved through physical sensors that are attached to the outside of the chassis. Since external sensors are subject to wear and tear, it is in general preferable to estimate external forces using solely sensors that may be hidden within the body of the robot. In this paper we contribute towards devising a method for performing such estimation tasks; more precisely, and building on the work of Liljebäck et. al., we analyze the kinematics of the snake robot systems, and propose a method to estimate contact forces and contact points in a case where the robot remains stationary starting from proprioceptive measurements of constraint forces, accelerations, and force balance equations of a rigid body. The efficacy of the estimators in estimating contact point, contact force and direction is verified experimentally

3.2.1 Introduction

Snake robots are mechanisms designed to mimic biological snakes, which aspire to inherit the robustness and stability properties of biological snake locomotion. Like their biological counterparts, mechanical snakes move using an array of different propulsion techniques such as lateral undulation, sinus lifting and sidewinding. These gaits are explained well by [14]. In principle this makes snake robots suitable for moving and adapting to some specific unknown and challenging environments such as in rubble following landslides or building collapse.

As of yet, this is largely an unrealized potential. Many existing systems for Obstacle-Aided Locomotion (OAL) adapt to the environment in an implicit manner only, with little utilization of mechanical sensor information. In contrast, the present work is part of an effort to achieve efficient, robust and intelligent locomotor behavior by exploiting continually updated information about the geometry and mechanical properties of the robot's immediate environment.

This paper investigates how these robots may acquire such information. We limit the study to planar snake robots, i.e., ones that are intended to navigate on a smooth, two-dimensional surface, potentially with obstacles that constrain the obtainable movements, as shown in Fig. 3.1. Planar snake robots are configured such that the axes of rotation of all joints are perpendicular to the ground plane. Therefore, they are unable to lift parts of their body off this plane, and thus cannot utilize gaits such as side-winding and sinus lifting. Because of this, planar snake robots rely on either anisotropic friction between their body and the ground plane, or contact with obstacles for propulsion.

It can be argued that locomotion based on anisotropic ground friction is a form of OAL on the microscopic level. However, according to tradition we distinguish between the two, and define OAL as locomotion that takes advantage of macroscopic obstacles. This is the type of locomotion of main interest in this paper. Furthermore, general results related to planar OAL might generalize to OAL in non-planar snake robots.

On one level, efficient OAL amounts to determining how to actuate the joints of the robot so that the links push against obstacles to efficiently produce propulsive forces in the desired direction. This may involve solving estimation problems such as tactile Simultaneous Localization and Mapping (SLAM) [15] and locating obstacles of a suitable shape in a suitable location to be useful for propulsion. It typically also involves path planning to ensure that the robot encounters a sufficient number of obstacles to maintain propulsion while navigating the terrain.

Information on how and where the snake robot is in contact with its environment is useful and potentially crucial for robust AOL. In this context, it is desirable to measure contact forces as accurately as possible. To the best of our knowledge, the current paradigm to solve this task is through hardware-based approaches, i.e., through sensors mounted on or close to the exterior of the snake. However, external force sensors have the disadvantage of being exposed to the environment and are subject to wear-and-tear from the movement of the snake robot.

The authors of [11] proposed an external force estimation method that depends solely on sensors hidden within the body of the snake robot. We will refer to such approaches as *proprioceptive contact force estimation*. The term *proprioception* (also referred to as *kinaesthesia*) represents an organism's sense of self-movement and body position. In vertebrates, this sense is encoded by special groups of sensory neurons in joint and muscle tissue. Most vertebrates also have *cutaneous* (or skin) *mechanoreceptors* allowing them to sense skin touch. Previous attempts at contact force measurement have tried to mimic the function of cutaneous mechanoreceptors using an electromechanical measurement system. The method proposed in this paper attempts to achieve similar results by using a system that is modeled after the vertebrate proprioception.

The recent availability of high-quality and low-cost multi-axis force transducers based on, e.g., piezoelectric elements, strain gauges or Fiber Bragg Grating (FBG) transducers, simplifies internal constraint force measurements. In addition, high precision and low cost accelerometers are becoming accessible with the development of mobile technology and household robotics, such as robotic lawn mowers and vacuum cleaners. These technologies enable investigating whether an alternative and economically favorable soft-sensing solution may outperform previous implementations, both in terms of precision and cost. In this paper we propose a method to estimate the contact force and contact point between a planar snake robot and its environment. The method is verified experimentally for a special case where joint angles are zero and the robot remains stationary. This paper serves as a starting point for further investigation into proprioceptive contact force estimation, particularly when the robot is moving.

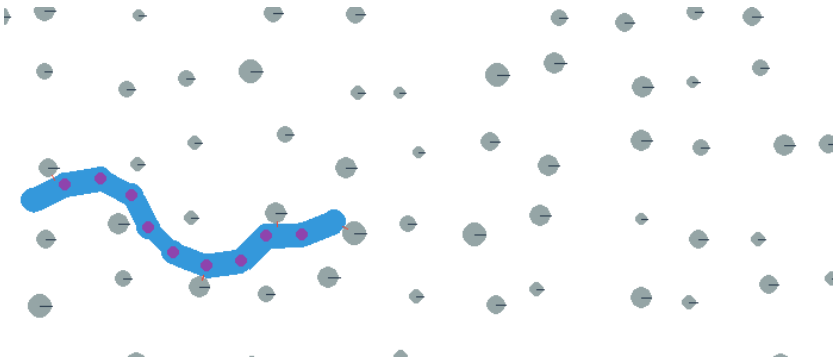


Figure 3.1: A simulated planar snake robot consisting of links (blue) and joints (purple) in a cluttered environment of fixed obstacles (gray). To autonomously understand which immediate actuation should be taken to propel oneself forward, and to autonomously plan the trajectory to make sure to be always in contact with enough obstacles to propel in the desired direction, the snake robot must be capable of sensing the surrounding environment and its properties.

3.2.2 Notation

To increase the readability of the remainder of this document, we present some of the used notation by means of the sketch in Figure 3.2 and its caption.

To complement the figure, we consider that a generic planar snake robot consists of N links connected by $N_j = N - 1$ joints whose axes are oriented in the same direction. The robot exists in a world coordinate frame (x_0, y_0). Each link of the robot has its own link local coordinate frame (x_i, y_i) where i is the link number. For the remainder of this paper, an integer superscript will denote the reference frame of the variable, and a subscript denotes the link index, e.g., $\ddot{r}^i_{i-1}$ denotes the acceleration of link $i-1$ in the link local frame of i. The local frames are oriented such that the x-axis forms a line between the axis of joint i and joint $i-1$, and the y-axis is pointing in the left transversal direction. The tail link of the robot is indexed as link 1 and the head is link N. The link angle of link i, θ_i for $i \in [1 \dots N]$, is defined as the angle between the global axis x^0 and the local axis x^i. The angle of the ith joint is denoted as ϕ_i for $i \in [1 \dots N_J]$. In the local frames, forces and torques can be schematized as in Figure 3.2. The relation between the link angles and the joint angles is finally given by

$$\phi_i = \theta_{i+1} - \theta_i. \tag{3.1}$$

3.2.3 Previous Work

Existing technologies for measuring contact forces in snake robots can be roughly divided into two categories: discrete and continuous. For example, the systems proposed by [16, 17] use discrete contact switches, and are therefore more concerned with contact force detection rather than measurement. The systems implemented by [18], [19], [20], and [21] instead measure forces on a continuous scale.

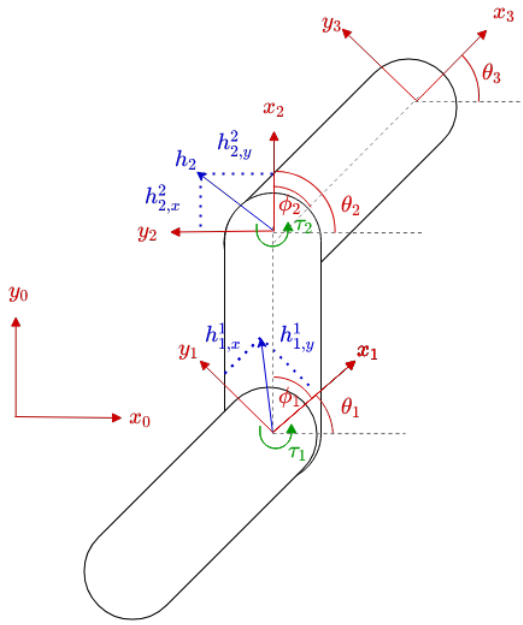


Figure 3.2: The kinematics of a simple 3-link planar snake robot as seen from above. The constraint forces h_i^i (blue) are resolved in their link-local frames (red vectors). The link angles θ_i and joint angles ϕ_i (red) relate the orientation of the link-local frames. The link torque τ_i (green) is the control input of the snake robot.

Common to all of these designs is the placement of force sensors on the outside of the snake robot, directly in between the robot body and the objects it is interacting with, or between the robot chassis and its outer shell.

In more detail, the technology developed by [21] is an optics based system. Every link of the snake robot is suspended by means of elastic springs within a cage. As the robot comes into contact with an obstacle, the link inside the cage gets displaced. This displacement is then measured using optical range sensors. Given information on the elastic coefficients and disposition of the springs, the measured displacement may then be converted into an estimate of the contact force.

Kulko, a snake robot designed by [22], implements a system which is similar in principle to the one developed by [21]. In this case each link is covered by a hollow spherical shell attached to the main body through an array of Force-Sensing Resistors (FSR). As the shell collides with obstacles, the resistance of the FSRs under the shell will change as a non-linear function of the contact force. This approach was also examined by [23] and [20]. In [18] the same principle is used to measure contact forces in a wheeled snake robot.

A capacitive contact force measurement system where the sensors can be wrapped around each link is outlined in [24]. While the primary element for

force measurement is different, the approach is similar to the FSR based systems in practice.

Using contact switches as a means of contact force detection is robust, but only provides a binary representation of the contact force (i.e., contact or no contact). In contrast, optical, capacitive or resistive force sensors enables measurement of the contact force on a continuous scale. These sensing techniques rely on material deformation to produce measurements. This translates into a trade-off between material stiffness and sensor sensitivity. Using materials that deform easily increases sensitivity, but might also make the exterior of the robot less robust as all forces to be measured must somehow be relayed from the environment to the robot chassis through the elastic element. This is undesirable, as a snake robot potentially relies on forceful contact between its exterior and the terrain to produce propulsion.

3.2.4 A Mathematical Model of the Proprioceptive Contact Force Estimation System

Towards the goal of devising the force estimator discussed in the introduction, we derive some considerations on the proprioceptive contact force sensing possibilities starting from the force balance equations of a rigid body. The equations are then utilized in the next sections to derive the proposed estimators. To do so we build on the kinematics and notation defined by Liljebäck [11].

Ideally, and as illustrated in Figure 3.2 (middle link with $i = 2$), a link in a snake robot is only affected by three forces:

1. h_i, the constraint force between link i and link $i + 1$ through joint i;
2. h_{i-1}, the constraint force between link i and link $i - 1$ through joint $i - 1$;
3. $f_{c,i}$, the sum of any external forces from the environment acting on the link (not shown in the figure).

Given this assumption, the force balance for a single link can be described as

$$m_i \ddot{r}_i = h_i - h_{i-1} + f_{c,i} \tag{3.2}$$

where m_i is the mass of link i and $\ddot{r}_i$ is the acceleration of this link's center of mass, respectively. The goal of the proprioceptive contact force measurement system is to effectively measure $f_{c,i}$ without the use of external contact force sensors. Assuming that the acceleration $\ddot{r}_i$, mass m_i and constraint forces h_i and h_{i-1} can be measured, the contact force $f_{c,i}$ can be calculated as

$$f_{c,i} = h_{i-1} - h_i + m_i \ddot{r}_i. \tag{3.3}$$

Thus, by using the proprioceptive measurement of constraint forces and acceleration, it is possible to produce an estimate of the contact forces without the use of external sensors. To reach a form for the equations in (3.3) that is more suited

for software implementation, it is necessary to derive equations for the kinematics of a planar snake robot. We now expand on (3.3) as defined in [11].

Define then the constraint force h_i^i as the force from link $i+1$ acting on link i through joint i resolved in the local frame of link i. To resolve such a constraint force vector in the link local frame of link i, the most natural choice would be to embed a force transducer located in link i measuring the constraint forces between link i and link $i+1$ directly.

All measurements must be referred to the same reference frame for (3.3) to be valid, but each constraint force h_i is originally resolved in its link local frame i. To resolve this, we introduce the rotation matrices $\mathbf{R}_{i-1}^i = \mathbf{R}_{\phi,i-1} \in \mathbb{SO}(2)$ which rotates a vector from frame $i-1$ to frame i, using the joint angle ϕ_{i-1}. By applying these rotation matrices, we can then redefine (3.3) as

$$f_{c,i}^i = \mathbf{R}_{\phi,i-1} h_{i-1}^{i-1} - h_i^i + m_i \mathbf{R}_0^i \ddot{r}_i^0, \tag{3.4}$$

which produces the contact force on link i, $f_{c,i}^i$, resolved in its own link local frame. The product $\mathbf{R}_0^i \ddot{r}_i^0$ can be interpreted as the global acceleration of the link's center of mass, rotated to the link local coordinate frame. The contact force $f_{c,i}$ is the vector sum of the friction force between the environment and the link $f_{f,i}$ and the normal force $f_{n,i}$ so that

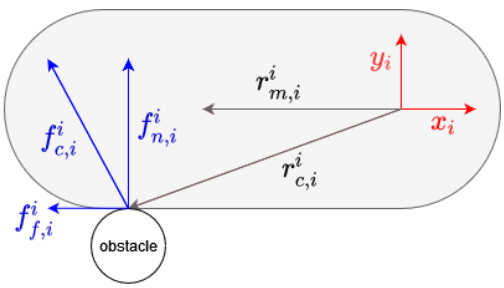


Figure 3.3: Schematic representation via a single "pill-shaped" snake robot link of how the link may be in contact with a fixed obstacle.

$$f_{c,i} = \begin{bmatrix} f_{f,i} & f_{n,i} \end{bmatrix}^T \tag{3.5}$$

as shown in Figure 3.3. The normal force $f_{n,i}$ can be obtained through vector decomposition as $f_{f,i}$ and $f_{n,i}$ are orthogonal by definition. Similar to the force balance used in (3.3), the external torque $\tau_{c,i}$ acting on link i can be found using a torque balance

$$\tau_{c,i} = \tau_{i-1} - \tau_i - I_i \ddot{\theta}_i \tag{3.6}$$

where I_i is the rotational inertia of link i around the link's center of mass. Assuming that the external torque is caused by the external force $f_{c,i}^i$, the external

torque can be written as

$$\tau_{c,i} = (r_{c,i} - r_{m,i}) \times f_{c,i} \tag{3.7}$$

where $r_{c,i}$ denotes the position of the point-of-contact between the link i and the obstacle, $r_{m,i}$ denotes the position of the center of mass of link i, and the symbol $\times$ denotes the cross product operator.

Figure 3.4: A digital render of a 5-link Boa snake robot, with a soda can for scale reference.

We now examine the special case where the robot is almost stationary, the joints are driven towards a joint angle $\phi_i \approx 0$, and there is no friction between the robot, its underlying surface or the obstacle. This configuration allows us to make the assumptions that

$$\begin{aligned} \mathbf{R}_{\phi,i} &\approx \mathbf{I}, & \ddot{r}_i^i &\approx 0, \\ \ddot{\theta}_i &\approx 0, & f_{f,i}^i &= 0. \end{aligned} \tag{3.8}$$

The assumption $\phi_i \approx 0$ is made to simplify the experimental setup and does not have any significance for the proposed estimators. In a case where $\phi_i \neq 0$, the form or statistical properties of the estimators will not change. Using the assumptions made in (3.8), on the force balance in (3.4), meaningful contact force estimates $\widehat{f}_{c,i}^i$ can be computed as

$$\widehat{f}_{c,i}^i = h_{i-1}^{i-1} - h_i^i \tag{3.9}$$

Using (3.6) and (3.7) under the assumptions given in (3.8), we can estimate the x-component $\widehat{r}_{c,i,x}^i$ of the contact point $r_{c,i}^i$ in the case where the obstacle is in contact with the flat side of link i by

$$\widehat{r}_{c,i,x}^i = \frac{\tau_{i-1} - \tau_i}{\widehat{f}_{n,i}^i} - r_{m,i,x}^i. \tag{3.10}$$

Thus, when the normal contact force $\widehat{f}_{n,i} \approx 0$, the signal $\widehat{r}^{i}_{c,i,x}$ will have a high variance and display a noisy and erratic behavior. Because of this, care must be taken when estimating the position of the contact point when the contact force is low.

3.2.5 Experiments

A series of 3 experiments were carried out to verify the efficacy of the contact estimation system outlined in Section 3.2.4. Experiments 1 and 2 are intended to verify the contact force estimator in (3.9), and experiment 3 is intended to verify the contact point estimator in (3.10). All experiments were carried out on a prototype Boa snake robot built at NTNU and configured with 5 links. A digital render of the robot can be seen in Figure 3.4.

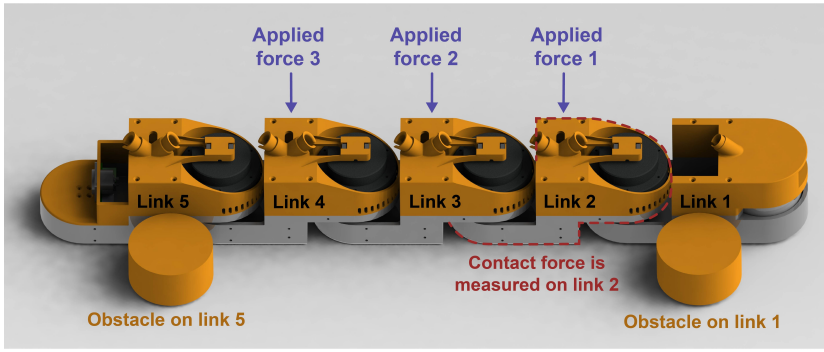


Figure 3.5: Experimental setup for Experiment 1 (top) and Experiment 2 (bottom) seen from above. The fixed obstacles are shown in grey, the force transducer in blue and the links of the robot in orange.

The setup of experiment 1 and 2 is shown in Figure 3.5 as seen from above and with the robot's forward direction to the right. Fixed obstacles were placed on the right-hand side of links 1 and 5. A force transducer (*HBM Z6FC3/20KG*) was used to apply an external normal force to link 2, pushing the robot into the obstacles. The joints of the robot were set to drive towards $\phi_i = 0$ using a PID controller. The force transducers in the joints of the snake robot are of the brand *ME-Messysteme K3D40-50N*. The applied force $f_{n,2}$, and the resulting contact force estimates $\widehat{f}_{n,1}$, $\widehat{f}_{n,2}$ and $\widehat{f}_{n,5}$ are shown in Figure 3.6. Experiment 2 is similar to Experiment 1 except that the external force is applied to link 3. The applied force $f_{n,3}$, and the resulting contact force estimates $\widehat{f}_{n,1}$, $\widehat{f}_{n,2}$ and $\widehat{f}_{n,5}$ for Experiment 2 are shown in Figure 3.7.

Experiment 3 is similar in setup to Experiment 1, but the external force is specifically applied at $r_{c,2,x} = -0.085m$. The position of the mass center is known to be $r_{m,i,x} = -0.035m$ for the links of the robot.

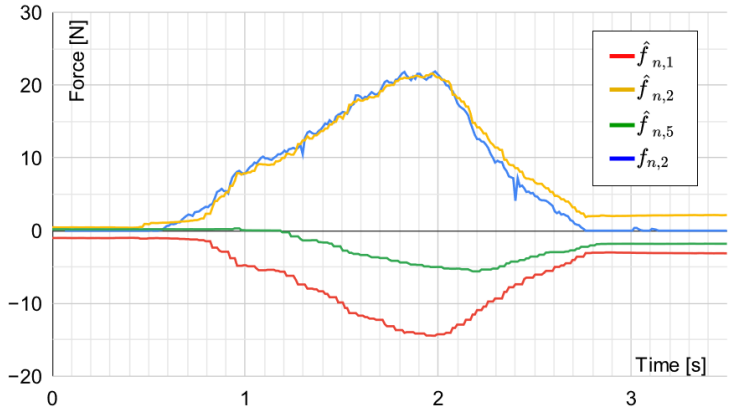


Figure 3.6: Estimated contact forces computed from the measurements collected while running Experiment 1.

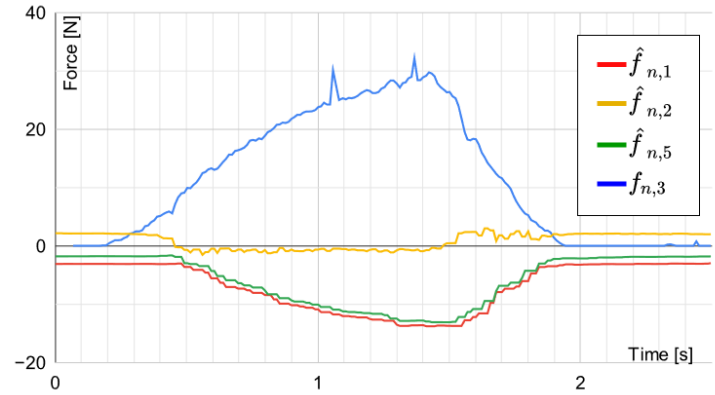


Figure 3.7: Estimated contact forces from the measurements collected while running Experiment 2.

3.2.6 Discussion

In Figure 3.6, the estimated contact force $\widehat{f}_{n,2}$ tracks the applied contact force $f_{n,2}$ with little error. Immediately after the applied force is removed at $t = 2.75$, the force estimates $\widehat{f}_{n,1}$, $\widehat{f}_{n,2}$ and $\widehat{f}_{n,5}$ show a stationary error. This is likely due to friction between the snake robot and the ground plane. When the external force is applied, then $|\widehat{f}_{n,1}| > |\widehat{f}_{n,5}|$, which is expected as the point of contact of $f_{n,2}$ is closer to link i than link 5. As the applied force $f_{n,2}$ decreases in the time span $t \in [2, 2.75]$ the estimated contact force $\widehat{f}_{n,2}$ shows an error from the applied contact force $f_{n,2}$. As the force is applied, the joint angles ϕ_i deviate slightly from the target angle of $\phi_i = 0$ as the servos attempt to drive the joint angle towards 0. When the external force is removed, the servos quickly return to $\phi_i = 0$ causing acceleration in the links. This acceleration manifests itself as an error in the contact force estimates through the acceleration term $\ddot{r}_i^i$ in (3.4). Thus, this is a weakness of the experimental setup rather than the method being

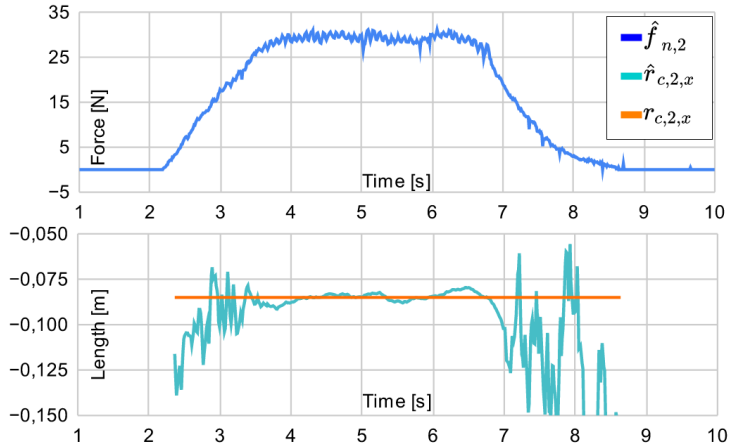


Figure 3.8: Applied contact force (upper plot), true contact point position and estimated contact point position (lower plot) for Experiment 3. The estimated contact force $\widehat{r}_{c,2,x}$ is filtered using a 10-tap moving average filter.

studied.

In Figure 3.7, the estimated contact force for link 2, $\widehat{f}_{n,2}$, is driven towards zero as the external force $f_{n,3}$ is applied to link 3. As an external contact force is applied to the robot, the servos jitter as they attempt to drive towards $\phi = 0$. It is likely that the jitter causes small movements in the body of the snake robot that counteracts the stiction between the robot and the surface.

In Figure 3.8, the contact point position estimate $\widehat{r}_{c,i,x}$ converges to its true value as the applied contact force converges towards a stationary value $f_{n,2} = 30N$. The estimator performs poorly when the applied contact force is non-constant. This is likely due to two factors: the high variance of $\widehat{r}_{c,i,x}$ when $f_{n,i}$ is sufficiently low and the high noise in τ_i and τ_{i-1} as the servos drive the joints towards $\phi = 0$ to compensate for the applied contact force. While the former is a property of the method being studied, the latter is a weakness of the experimental setup.

3.2.7 Conclusions

We have shown that using a proprioceptive contact force estimation system is a possible strategy for performing meaningful contact force detection and measurement tasks in terrestrial snake robots under semi-static conditions. With this strategy, the proprioceptive systems can be completely hidden within the mechanical structure of the robot, a feature that improves the robustness of the sensing hardware compared to other approaches where the sensors are placed on or near the outside of the robot.The ability to estimate external forces' point of attack is relevant for future tactile SLAM applications.

Physical experiments confirm that the proprioceptive contact force estimators outlined in the paper are promising alternatives to previous methods of measuring

contact forces, but refer solely to the case where the robot remains stationary. Further research should be made to develop estimators for contact force and contact point estimation in the case where the robot is in motion, and when the joint angles are non-zero, $\phi_i \neq 0$. The singularity of the contact point (3.10) as $\widehat{f}^i_{n,i} \to 0$ also requires further investigation to achieve a robust overall system.

3.3 Article II: Improved Jacobian Matrix Estimation Applied to Snake Robots

Abstract

Two manipulator Jacobian matrix estimators for constrained planar snake robots are developed and tested, which enables the implementation of Jacobian-based Obstacle Aided Locomotion (OAL) control schemes. These schemes use obstacles in the robot's vicinity to obtain propulsion. The devised estimators infer manipulator Jacobians for constrained planar snake robots in situations where the positions and number of surrounding obstacle constraints might change or are not precisely known. The first proposed estimator is an adaptation of contemporary research in soft robots and builds on convex optimization. The second estimator builds on the Unscented Kalman Filter. By simulations we evaluate and compare the two devised algorithms in terms of their statistical performance, execution times and robustness to measurement noise. We find that both algorithms lead to Jacobian matrix estimates that are similarly useful to predict end-effector movements. However, the Unscented Filter approach requires significantly lower computing resources, and is not poised by convergence issues displayed by the convex optimization based method. We foresee that the estimators may have use in other fields of research, such as soft robotics and visual servoing. The estimators my also be adapted for use in general non-planar snake robots.

3.3.1 Introduction

Snake robots are mechanisms designed to mimic biological snakes, which aspire to inherit the robustness and stability of biological snake locomotion. Like their biological counterparts, and as explained in detail in [2, 14], mechanical snakes move using an array of different propulsion techniques such as lateral undulation, sinus lifting and sidewinding. In principle this makes snake robots suitable for moving and adapting to unknown and challenging environments, such as in rubble following landslides or collapsed buildings. As of now this is largely an unrealized potential. Many experimental systems for Obstacle-Aided Locomotion (OAL) adapt to the environment in an implicit or heuristical manner only, with little utilization of mechanical sensor information. In contrast, the present work is part of an effort to achieve efficient, robust and intelligent locomotor behavior by exploiting information about the geometry and mechanical properties of the surroundings of the robot.

To do so, a generic strategy consists of calculating and then exploiting the *manipulator Jacobian* (or simply *Jacobian*) of the system. The Jacobian is a matrix which relates the robot joint velocities to its end-effector velocities through a linear transformation parameterized by the joint states (see Section 3.3.3 for more details). In many robot systems, the kinematics of the robot is known and time-invariant, which makes it possible to compute the Jacobian analytically. In the case of snake robots, however, computing the Jacobian is a much more involved task, partly because it depends on the continually-changing configuration of contact points between the robot and the environment. In snake robots, the

Jacobian matrix gives a relation between the joint speeds and the velocity of the head. This information may prove useful when designing locomotion strategies for snake robots, in the same way that Jacobian is essential in motion planning for robotic manipulators. This provokes the need for research on how to effectively estimate Jacobian matrices in constrained snake robots, ideally in real time.

The present paper addresses this problem, but specifically for the case of planar snake robots, i.e., ones that are intended to navigate on a smooth, two-dimensional surface, with obstacles that constrain the robot's movements. Planar snake robots are configured such that the axes of rotation of all joints are perpendicular to the ground plane. Therefore, they are unable to lift parts of their body off this plane, and thus cannot utilize gaits such as side-winding and sinus lifting. Because of this, planar snake robots rely on either anisotropic friction between their body and the ground plane, or its macroscopic equivalent: contact with obstacles, for propulsion. Planar snake robots have limited practical use, but their motion perfectly resembles that of a general 3D snake robot exhibiting pure lateral undulation (i.e., with no lifting action) on a flat surface containing obstacles. Thus they lend themselves to studying this particular mode of propulsion. Furthermore, results based on planar motion may create a foundation on which research generalized to non-planar scenarios can be performed, and the chosen platform thus enables basal research into OAL and related subjects.

Planar snake robots share similarities with robot manipulators in the sense that they both are constituted of primarily rotational joints and rigid links. However, they differ in the following aspects:

1. A planar snake robot is continually in contact with the surface underneath the robot. This introduces friction between the robot and the surface.

2. Most robot manipulators are grounded, in the sense that a base coordinate frame is typically fixed in the world frame. In contrast, generally no part of a planar snake robot is fixed in relation to the world frame.

3. A typical manipulator is intended to interact with its environment only through its end-effector, thus its kinematic equations have a constant structure. In contrast, a snake robot is intended to interact with its environment at any point of any link. The corresponding constraints cause structural changes in the kinematics of the robot/environment system as the robot comes into contact with new obstacles or departs from obstacles it was previously in contact with.

The last aspect is especially important as we discuss the motivation for this paper. Figure 3.9 shows how a snake robot in contact with its environment might be modeled as a kinematic chain, using pairs of translational and rotational joints fixed in the world frame to model the obstacles. The kinematics of such a model are examined in detail in [25] and [26], and will not be treated further, as the jacobian matrix estimation strategies proposed in this paper are completely independent of the model. As the snake moves through its environment, the number of obstacle

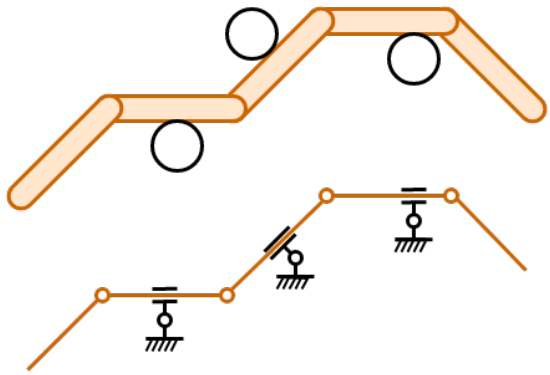


Figure 3.9: Top view of a 5 link planar snake robot (orange) in contact with 3 obstacles (black outline). The kinematic representation of the snake robot interacting with the obstacles is shown in the lower figure. Note how the obstacles can be modeled as a translational joint and a rotational joint attached to the world frame.

constraints and their positions relative to the robot will change. With incomplete prior knowledge of the position, shape and orientation of each obstacle, it is challenging to ascertain the constrained kinematics of the snake robot. Finding the Jacobian for a constrained planar snake robot is desirable from a control perspective, but due to the uncertain nature of the snake robot's kinematics, finding the Jacobian in closed form is challenging. This paper seeks to find an estimate $\widehat{\boldsymbol{J}}$ for the robot Jacobian, without the need for an exact model of the constrained kinematics of the system.

Estimating Jacobians for robot manipulators is an established and well researched field. Different methods for Jacobian estimation are diligently used in the field of visual servoing [27–29] [30], which serve as inspiration for the contributions in this paper. Similar methods have been applied for calibration of robotic stereo vision [31]. While this paper specifically addresses Jacobian estimation in planar snake robots, the endeavor of estimating Jacobians is highly relevant to other fields within robotics, and is discussed further in Section 3.3.8.

3.3.2 Notation

The notation used for the remainder of the paper is inspired from [11]. The kinematics characterizing a planar snake robot may be derived by inspecting Figure 3.10: a generic planar snake robot consisting of N links is composed of $N_j = N - 1$ joints whose axes are oriented in the same direction. The robot is assumed to be embedded in a frame of reference denoted by (x_0, y_0). Each link of the robot has its own link local coordinate frame (x_i, y_i) where i is the link number. The local frames are oriented such that the x-axis forms a line between the axis of joint i and that of joint $i - 1$, and the y-axis is pointing in the left transversal direction. The tail link of the robot is indexed as link 1 and the head as link N. As shown in the figure, the angle between the global axis x_0 and the local axis x_i, for $i \in [1 \dots N]$, is then denoted as θ_i and called the link angle of link i. The relative angles between adjacent links, i.e. the joint angels, are instead

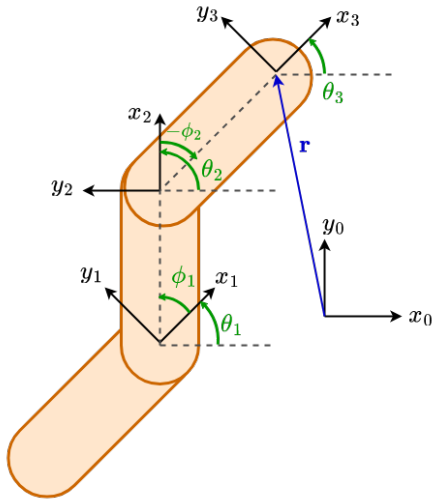


Figure 3.10: Schematic diagram for the computation of the kinematics of a simple 3-link planar snake robot as seen from above. The link angles θ_i and joint angles ϕ_i relate to the orientation of the local frames of the links.

denoted as ϕ_i for $i \in [1 \dots N_J]$. It follows that the relation between the link angles and the joint angles is given by

$$\phi_i = \theta_{i+1} - \theta_i. \tag{3.11}$$

The vector containing the joint angles $\boldsymbol{\phi}$ and vector containing the joint speeds $\dot{\boldsymbol{\phi}}$ then be defined as

$$\boldsymbol{\phi} = \left[\boldsymbol{\phi}_1,\, \boldsymbol{\phi}_2, \dots, \boldsymbol{\phi}_{N_j}\right]^\top$$
$$\dot{\boldsymbol{\phi}} = \frac{d}{dt}\boldsymbol{\phi} = \left[\dot{\boldsymbol{\phi}}_1,\, \dot{\boldsymbol{\phi}}_2, \dots, \dot{\boldsymbol{\phi}}_{N_j}\right]^\top$$

For the remainder of this paper a superscript $(\cdot)^n$ denotes the iteration step of the algorithm being studied.

3.3.3 The Manipulator Jacobian

Consider a generic robot manipulator with N rigid links and $N-1$ joints as in Figure 3.10. We denote the end-effector position as $\boldsymbol{x}$, the end-effector velocity as $\dot{\boldsymbol{x}}$, the joint angles as $\boldsymbol{q}$ and the joint velocities as $\dot{\boldsymbol{q}}$. The end-effector position is related to the joint angles by a forward kinematic function:

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{q}). \tag{3.12}$$

If a robot involves one or more rotational joints the kinematic function $\boldsymbol{f}$ is

often highly non-linear. For the purpose of developing automatic manipulation control schemes it is of interest to determine how a set of joint velocities $\dot{\boldsymbol{q}}$ will affect the velocity of the end-effector, $\dot{\boldsymbol{x}}$. This can be obtained by differentiating (3.12) wrt. time, so that

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}. \tag{3.13}$$

From an intuitive perspective, the Jacobian matrix $\boldsymbol{J}(\boldsymbol{q})$ corresponds to a parameterized linear transformation of the joint velocities $\dot{\boldsymbol{q}}$ to the end-effector velocities $\dot{\boldsymbol{x}}$. The Jacobian is also essential in mapping between joint torques and tool point forces and torques. In the case that $\boldsymbol{J}(\boldsymbol{q})$ is invertible, this quantity can be used to compute a set of joint velocities $\dot{\boldsymbol{q}}$ for any desired end-effector velocity $\dot{\boldsymbol{x}}$. In this case

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q})\dot{\boldsymbol{x}}. \tag{3.14}$$

If $\boldsymbol{J}(\boldsymbol{q})$ is not invertible, then computing $\dot{\boldsymbol{q}}$ as a function of $\dot{\boldsymbol{x}}$ may be performed using other methods relying on the existence of $\boldsymbol{J}(\boldsymbol{q})$, such as constrained optimization, or by determining the Moore-Penrose inverse $\boldsymbol{J}^{+}(\boldsymbol{q})$ for $\boldsymbol{J}(\boldsymbol{q})$.

Summarizing, the availability of $\boldsymbol{J}(\boldsymbol{q})$ is beneficial from a control perspective. For most robotics applications, the manipulator Jacobian $\boldsymbol{J}(\boldsymbol{q})$ can be found analytically as the kinematics of the robot is known. In the case of snake robots, the Jacobian relates the joint velocities to any of the snake robot's state variables. We might not only be interested in the movement of the end-effector (the head link in the case of snake robots), but also in the motion of the remaining links in the robot since this information might be useful for activities such as low-level control or path planning. Furthermore, the correspondence between joint torques and contact forces has great relevance for effective OAL, e.g for minimizing obstacle-related friction, and avoidance or resolution of jam situations (cf. [32]).

As mentioned in the introduction and shown in Figure 3.9, the Jacobian of a constrained snake robot depends on how and where the robot touches obstacles. In OAL situations, the configuration of such obstacles relative to the robot is continually changing, implying that the corresponding Jacobian is also time-varying. This introduces the problem of having to estimate it from field data as the locomotion is unfolding.

3.3.4 Optimization Based Jacobian Matrix Estimation

The first estimator proposed in this paper is inspired from Yip et. al [12, 13], where the authors present a framework for model-less control in soft robots. Soft robots share a property with planar snake robots in that the exact kinematics of the robot are difficult to ascertain. The control framework considered in [12, 13], schematized in Figure 3.12, relies on recursively estimating the Jacobian based on measurements of the soft robot's control inputs and of the resulting end-effector movements. We adapt the model-less control framework to be usable for planar

Figure 3.11: A digital render of a 5-link Boa snake robot, with a soda can for scale reference.

snake robots. The notation used in this section is consciously adapted to conform with the choices mentioned in Section 3.3.2. The joint states of the snake robot are given by the joint angles $\boldsymbol{\phi} \triangleq \boldsymbol{q}$, and the end-effector position in the reference frame is given by the position of the head of the snake robot, given in world frame $\boldsymbol{x} \triangleq \boldsymbol{r}$.

This paper explicitly considers using the position of the head of the robot $\boldsymbol{r}$ as the system output, a choice that most closely relates to the system output used in [12]. In practice one may choose different system outputs, even if this may add computational complexity. This could include the linear velocity of any of the remaining links or the angular velocities of the links. Given a current estimate of the Jacobian $\widehat{\boldsymbol{J}}^n$, the current joint speeds $\dot{\boldsymbol{\phi}}^n$ and the current head velocity $\boldsymbol{r}^n$, the modified Jacobian estimator for a snake robot is given by the optimization problem

$$\begin{aligned} \min_{\widehat{\boldsymbol{J}}^{n+1}} \quad & ||\Delta \boldsymbol{J}||_2 \\ \text{s.t.} \quad & \widehat{\boldsymbol{J}}^{n+1} = \widehat{\boldsymbol{J}}^n + \Delta \boldsymbol{J} \\ & \dot{\boldsymbol{r}}^n = \widehat{\boldsymbol{J}}^{n+1} \dot{\boldsymbol{\phi}}^n. \end{aligned} \tag{3.15}$$

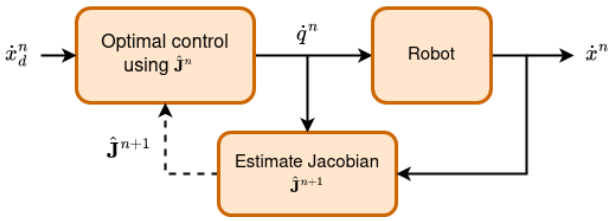


Figure 3.12: A simplified schematic of the control flow in the model-less control framework presented by [12]. At each timestep n the scheme computes a current estimate $\widehat{\boldsymbol{J}}^n$ of the Jacobian, combines it with the desired trajectory $\dot{\boldsymbol{x}}_d^n$, determines the desired joints velocities $\dot{\boldsymbol{q}}^n$, and actuates them. The system then measures the trajectory of the robot $\dot{\boldsymbol{x}}^n$, and uses it together with $\dot{\boldsymbol{q}}^n$ to compute $\widehat{\boldsymbol{J}}^{n+1}$, the estimated Jacobian, for the next time step.

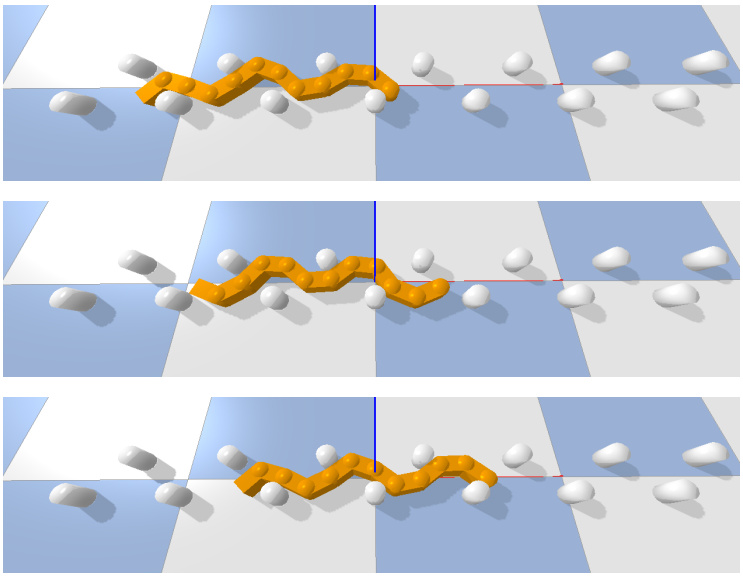


Figure 3.13: The simulated snake robot moving through its environment using a basic undulation pattern at three different time steps. Each of the colored squares on the underlying plane measures $1m \cdot 1m$.

Problem (3.15) can be interpreted as follows: find a minimal change $\Delta \boldsymbol{J}$ in the current Jacobian $\boldsymbol{J}^t$ so that the new Jacobian $\boldsymbol{J}^{t+1} = \boldsymbol{J}^t + \Delta \boldsymbol{J}$ explains the relation between the measured values $\dot{\boldsymbol{r}}^t$ and $\dot{\boldsymbol{\phi}}^t$. The Euclidean norm $\|\cdot\|_2$ is used as the metric for determining the magnitude of the change in the Jacobian. Because of this, (3.15) falls in the category of equality constrained convex optimization (CO) problems for which many efficient numerical solvers exist [33]. Some care should be taken when using this method, as (3.15) may have some undesirable properties under specific conditions. By definition (3.15) is not guaranteed to have a unique (and global) optimum and can, under some conditions, have infinitely many or no solutions.

An inherent issue with the Jacobian proposed in [12] is that if the actuators of the robot are stationary, solving the optimization problem does not lead to meaningful estimates. On inspection of the last constraint in (3.15) the estimated Jacobian $\widehat{\boldsymbol{J}}^{n+1}$ will uncontrollably diverge for sufficiently small values of $\dot{\boldsymbol{\phi}}$. In a real-life scenario the measured values $\dot{\boldsymbol{r}}^t$ and $\dot{\boldsymbol{\phi}}^t$ will be noisy to some degree. An optimization-based formulation of the estimator makes it difficult to analytically find the statistical properties of the estimates (the statistical properties of the estimator $\widehat{\boldsymbol{J}}$ are not discussed in [12]). A lack of statistical performance indices could lead to issues down-the-line if the estimates are used in e.g. control strategies, as control strategies typically require information about the statistical properties of the estimate to infer qualities such as stability and convergence.

3.3.5 Unscented Kalman Filter based Estimation

The Unscented Kalman Filter (UKF) approach is designed to perform state and parameter estimation on non-linear state-space problems by improving on the Extended Kalman Filter (EKF). The main difference between these two filters is the mechanism for propagating the error covariances: while the EKF relies on the linearization of the state-space model, the UKF relies on Unscented Transforms [34, 35]. The UKF has been shown to statistically outperform the EKF for a large subset of problems while exhibiting a similar or even reduced computational load.

While both methods rely on an assumption that the underlying is locally linearizable around its state, the UKF outperforms the EKF in scenarios where the model nevertheless contains discrete non-linearities, owing to its use of the unscented transform. This is an essential property when performing estimation on a moving snake robot as it might come in contact with new obstacles or depart obstacles it previously was in contact with. This will cause a discrete change in the system kinematics, and thus also its Jacobian.

Moreover, the UKF can be used as a model parameter estimation algorithm, the application of which is the primary interest in this paper. The general parameter estimation problem is stated by [35] as

$$\boldsymbol{y}^n = \mathcal{G}(\boldsymbol{x}^n, \boldsymbol{w}) \tag{3.16}$$

where the non-linear map $\mathcal{G}(\cdot)$ relates a system input $\boldsymbol{x}^n$ to a system output $\boldsymbol{y}^n$ parameterized by a vector $\boldsymbol{w}$ (note that [35] uses a notation for which the state variable $\boldsymbol{x}$ has a different meaning than in our Sections 3.3.2 and 3.3.3).

Estimating the Jacobian $\widehat{\boldsymbol{J}}^{n+1}$ in (3.15) can then be formulated as a parameter estimation problem in which the input is $\boldsymbol{x}^n = \dot{\boldsymbol{\phi}}^n$, the output is $\boldsymbol{y}^n = \dot{\boldsymbol{r}}^n$, the parameters $\boldsymbol{w} = \widehat{\boldsymbol{J}}^{n+1}$, and the non-linear mapping $\mathcal{G}(\cdot)$ is the matrix multiplication operation. This enables using a UKF based approach to estimate $\widehat{\boldsymbol{J}}^{n+1}$ via reformulating the parameter estimation problem into the non-linear state-space representation

$$\begin{aligned} \widehat{\boldsymbol{J}}^{n+1} &= \widehat{\boldsymbol{J}}^n + \boldsymbol{\eta}^n \\ \dot{\boldsymbol{r}}^n &= \widehat{\boldsymbol{J}}^n \dot{\boldsymbol{\phi}}^n + \boldsymbol{\nu}^n \end{aligned} \tag{3.17}$$

where both $\boldsymbol{\eta}^n$ and $\boldsymbol{\nu}^n$ are zero-mean stochastic variables, $\boldsymbol{\eta}^n$ is the process noise, and $\boldsymbol{\nu}^n$ is the measurement noise. This formulation allows users to exploit process noise covariance as a tuning parameter. This, in turn, influences the convergence properties and tracking performance of the filter. For a system with stationary parameters, $\boldsymbol{\eta}^n$ should be near zero as our confidence in the parameter estimates do not deteriorate over time.

In our case, $\widehat{\boldsymbol{J}}^n$ changes with time and requires thus a positive definite covariance for the process noise. In general, larger values for the process noise covariance will lead to a quicker response to changes in $\boldsymbol{J}^n$, but also more noise in the estimate $\widehat{\boldsymbol{J}}^n$. Conversely, small values for the process noise will give less noise in the

estimate of $\widehat{\boldsymbol{J}}^n$, but might introduce significant lag in the estimates.

We finally note the following detrimental but fundamental property that mimics the problem observed at the end of Section 3.3.4: the Jacobian estimation problem is not globally observable, since the non-linear mapping defined in (3.16) is in our case a linear mapping with respect to $\widehat{\boldsymbol{J}}^n$. The linear map is not bijective, preventing both global and local observability. However, the mapping retains the property of being *practically identifiable* [36], a property that implies that the Jacobian $\widehat{\boldsymbol{J}}^n$ can be made observable by introducing sufficiently rich input-output data pairs.

In our case, the parameters can be made observable by providing a sufficiently rich control inputs $\dot{\boldsymbol{\phi}}^n$. At the same time, due to the non-linear mapping being practically identifiable, it is difficult to infer any general guarantees on the stability, convergence or correctness of the estimated Jacobians. This may have important implications for path planning and lower-level control, which should be carried out in such a manner as to render the system practically identifiable.

Summarizing, for both the algorithms proposed in this paper the lack of persistently exciting inputs causes numerical and theoretical problems. As this paper is focused on proposing and characterizing these algorithms, how to mitigate this inherent problem is considered a future work.

3.3.6 Method

This section outlines a series of simulation experiments designed to evaluate the performance and execution times of the two algorithms described in Section 3.3.4 and 3.3.5 and their robustness to measurement noise. All experiments were performed using a snake robot simulator which was purpose-built for OAL research. The simulator is built upon the physics engine [37]. All experiments were performed on a simulated snake robot with 11 links that emulates a snake robot platform currently under development by the authors. A render of the robot is shown in Figure 3.11. Essential physical parameters of the simulated robot are given in Table 3.1. Three experiments were carried out to investigate different

Link Width	84	mm
Axis-Axis Distance	130	mm
Link Mass	500	g
Link Friction coef.	0.1	-
Joint Torque	3	Nm

Table 3.1: Summary of the most relevant physical parameters of the simulated robot.

properties of the two Jacobian estimation algorithms. All three experiments share the same basic setup: the robot is set in a starting position $\theta_N = 0$, $\boldsymbol{r} = \boldsymbol{0}$ and $\boldsymbol{\phi} = \boldsymbol{0}$ on an infinitely large plane populated by cylindrical obstacles and where $\boldsymbol{0}$ is the null vector of appropriate dimensionality. The obstacles have a radius of 50 mm, are fixed within the world frame and are placed at regular intervals in two

rows with coordinates given in meters by

$$\begin{bmatrix} 0.5k + 0.25 & 0.1 \end{bmatrix}^T \forall k \in \{-3, 10\} \\ \begin{bmatrix} 0.5k & -0.1 \end{bmatrix}^T \forall k \in \{-3, 10\}. \tag{3.18}$$

The snake robot is actuated to perform an undulation pattern that creates propulsion by a rudimentary interaction with the obstacles without the need for feedback control. This form of propulsion is only possible because the exact location and properties of the obstacles are known. The position or geometry of the obstacle are *not* known to the two jacobian estimation strategies, and that it is only given for the sake of reproducibility of the experiments.

The goal of the experiments is to examine the behavior of the jacobian estimation, not the behavior of the control strategy. The design of a more complex control strategy leveraging the estimated jacobians is considered a future work. The desired joint angles $\boldsymbol{\phi}_k^n = \begin{bmatrix} \phi_{d,1}^n, & \phi_{d,2}^n, & \ldots, & \phi_{d,N_j}^n \end{bmatrix}^T$ are computed as

$$\boldsymbol{\phi}_{d,k}^n = \frac{\pi}{3} \sin\left(4n\Delta_t - \frac{\pi}{3}k\right) \tag{3.19}$$

where Δ_t is the time step of the simulation. The other parameters for the undulation pattern were chosen to create propulsion in the snake robot, for the given set of obstacles. A visualization of the snake moving using the pattern described in (3.19) through the obstacles described in (3.18) is shown in Figure 3.13. For all three experiments the simulation is run for a duration of 10 s and a time step of $\Delta_t = \frac{1}{240}$ s. The three experiments are described as follows:

- *Experiment 1:* The Jacobian is estimated using measurements of $\dot{\boldsymbol{r}}^n$ and $\dot{\boldsymbol{\phi}}^n$ from the described simulation without further alteration.

- *Experiment 2:* Similar to Experiment 1 except that the undulation of the robot is commanded to halt such that $\dot{\boldsymbol{\phi}}^n$ rapidly approaches 0 at $t = 5$ s and remains stationary until $t = 7$ s where the undulation is resumed.

- *Experiment 3:* Is similar to Experiment 1, except that a measurement noise is applied to the measurement of $\dot{\boldsymbol{r}}^n$ such that $\tilde{\dot{\boldsymbol{r}}}^n = \dot{\boldsymbol{r}}^n + \delta_r$ where $\delta_r \sim \mathcal{N}(0, \boldsymbol{\Sigma}_r)$ and $\boldsymbol{\Sigma}_r = 0.1 \cdot \boldsymbol{I}$ is the covariance of δ_r with $\boldsymbol{I}$ being the identity matrix.

Ideally, the metric to measure the performance of the estimators would be to compare the estimated Jacobian $\hat{\boldsymbol{J}}$ to the true Jacobian $\boldsymbol{J}$. As previously discussed, the true Jacobian is challenging to obtain so other metrics of performance are used.

We apply a metric commonly used in machine learning and see if our model produces the expected output from a known input. For each time step, a prediction is produced from our estimators based on the next input $\dot{\boldsymbol{\phi}}^{n+1}$ and the current Jacobian $\hat{\boldsymbol{J}}^n$. The input $\dot{\boldsymbol{\phi}}^{n+1}$ is applied to the snake robot and produces an output $\dot{\boldsymbol{r}}^{n+1}$. The true value of $\dot{\boldsymbol{r}}^{n+1}$ is compared to the prediction from the estimators to evaluate their performance. The Mean Square Error (MSE) is used to evaluate

the error of the estimators. Both algorithms were implemented in Python, using *cvxpy* [38, 39] for convex optimization and *filterpy* [40] for the Unscented Kalman Filter and associated resources.

3.3.7 Results

All plots showing the results of Experiments 1-3 are placed in Subsection 3.3.10 due to their size and visual complexity, but should be interpreted as a part of this section.

Experiment 1

The data collected from Experiment 1 can be visualized in Figures 3.18a-3.18d. The CO-based algorithm provides decent predictions for the y-component of the head velocity vector $\dot{r}_y$, but diverges for the x-component of the head velocity vector $\dot{r}_x$ over time. The UKF-based algorithm predicts both components of $\dot{\boldsymbol{r}}$ without diverging. The Mean Square Error between the estimated $\dot{\boldsymbol{r}}$ and the true value of $\dot{\boldsymbol{r}}$ is shown in Figure 3.14.

Experiment 2

The data from Experiment 2 is visualized in Figures 3.19a-3.19d. As the robot becomes stationary shortly after $t = 5\,\text{s}$, the CO-based algorithm rapidly diverges. This is compliant with what was theorized in Section 3.3.4. As the robot continues its movement at $t = 7\,\text{s}$ the CO-based algorithm diverges further. As $\dot{\boldsymbol{\phi}} \to 0$ the UKF-based algorithm also displays stability issues. However, as the robot resumes moving at $t > 7\,\text{s}$, the UKF-based algorithm shows a higher degree of error until $t = 7.3\,\text{s}$ but rapidly converges to the true value of $\dot{\boldsymbol{r}}$ as $t > 7.3\,\text{s}$.

Experiment 3

The results from Experiment 3 are shown in Figures 3.13a-3.13d, with Figure 3.17 depicting a realization of the measurement noise about $\dot{\boldsymbol{r}}_x$. Similarly to what was seen in Experiment 1, the CO-based algorithm diverges for $\dot{\boldsymbol{r}}_x$. To compare the two approaches from a quantitative perspective, we report the MSE of both algorithms in Figure 3.15.

Execution times

The execution times of both the algorithms are shown in Figure 3.16. The CO-based algorithm has an average computation time of 16.32 ms for each time step,

Estimator	MSE
CO for $\dot{r}_x$	0.03018
CO for $\dot{r}_y$	0.00076
UKF for $\dot{r}_x$	0.00025
UKF for $\dot{r}_y$	0.00028

Figure 3.14: Mean Square Error (MSE) for the estimators in Experiment 1.

while the UKF-based algorithm has an average computation time of 1.25 ms for each time step.

3.3.8 Discussion

Experiment 1

The UKF-based algorithm outperforms the CO-based algorithm by having a significantly lower MSE for both components of $\dot{\boldsymbol{r}}$. The error of both algorithms increases when there are large changes in $\dot{\boldsymbol{r}}$ (e.g., at $t = 8$ where there is a near-instantaneous change in $\dot{\boldsymbol{r}}$). This is expected, as a large unexpected change in velocity will rapidly change the true value of the Jacobian of the system.

Experiment 2

This experiment displayed the inherent stability issues of both algorithms, as theorized in Section 3.3.4. A key difference between them is seen in their behavior immediately after the robot resumes movement at $t = 7s$. The CO-based algorithm implements a strict equality bound. After the algorithm diverges when $\dot{\boldsymbol{\phi}} \to 0$ and is restarted, the algorithm attempts to solve the constrained optimization problem to find $\hat{\boldsymbol{J}}^{n+1}$, based on an inaccurate estimate of $\hat{\boldsymbol{J}}^{n}$, leading to an inaccurate estimate of $\hat{\boldsymbol{J}}^{n+1}$. This problem is propagated into the next step of the algorithm. One simple solution to this issue would be to reset the values of the values of the Jacobian to $\hat{\boldsymbol{J}}^{n} = \mathbf{0}$ immediately after a halt. In contrast to the CO-based method, the UKF-based algorithm shows no degradation in performance after the halt compared to before the halt, except for a brief transient.

Experiment 3

The performance of the algorithms seem to degrade in a similar fashion when subjected to increasing measurement noise. The performance of the UKF-based filter can be tuned as described in [35], by adjusting the values of the measurement and process noise covariance matrices. The ratio of the elements in these two matrices controls the trade-off between the filter's ability to rapidly respond to sudden changes in state and the filter's robustness to noise.

Analysis of execution times

The higher computation time for the CO-based algorithm is likely due to the estimator being based on a numerical solver; the *cvxpy* platform uses the open source OSQP, SCS, and ECOS solvers [38]. The lower computation time of the UKF-based algorithm is likely because it is based on basic matrix computations and

Estimator	MSE
CO for $\dot{r}_x$	0,03152
CO for $\dot{r}_y$	0,00161
UKF for $\dot{r}_x$	0,00173
UKF for $\dot{r}_y$	0,00164

Figure 3.15: Mean Square Error (MSE) for the estimators in Experiment 2

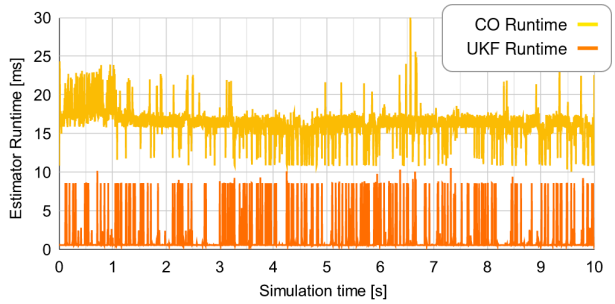


Figure 3.16: The computation time for each time step for the two estimators.

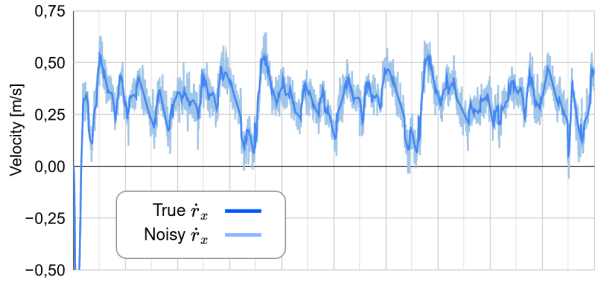


Figure 3.17: The true $\dot{\boldsymbol{r}}_x$ compared to the noisy signal used in Experiment 3

decompositions instead of numerical optimization. It may be possible to improve the average computation time for both algorithms through code optimization; such optimization has not been part of this study. Due to the fundamental differences in the two algorithms, however, we expect our qualitative comparison to survive such improvements.

Possibilities for generalization

While this paper focuses on Jacobian matrix estimation for planar snake robots, both algorithms are adaptable to a 3-dimensional scenario by increasing the dimensionality of the measurement vector $\dot{\boldsymbol{r}}^n$ from $\mathbb{R}^2$ to $\mathbb{R}^3$. Care should be taken during this process, as the number of unobservable variables will increase and the issues with convergence and observability likely will worsen as the dimensionality of the measurement vector increases.

Both algorithms presented in this paper are easy to adapt to other kinds of problems as they require little or no information about the dynamics of the system they are applied to. The proposed UKF-based algorithm may be relevant to the field of soft robotics or constrained robotics with unknown kinematics. Generally, the UKF-based algorithm may be useful for any system that can be modeled as the general parameter estimation problem in (3.16).

3.3.9 Conclusions

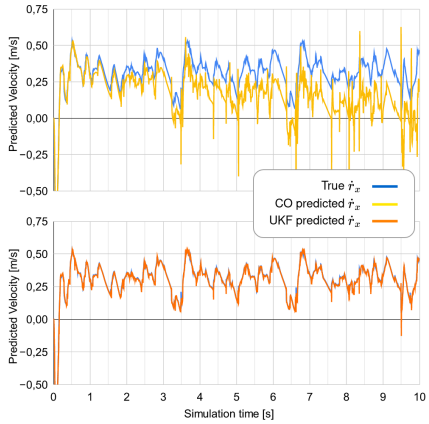
Jacobians capture the complex interactions of a planar snake robot. It models a constrained kinematic system, with its surrounding obstacles, by representing a mapping between the robot's joint velocities or torques on one hand, and its movements or interaction forces on the other hand. A constrained snake robot's Jacobian matrix may thus be used as part of an Obstacle Aided Locomotion (OAL) control schemes to allow the robot to utilize the surrounding obstacles and its many joint actuators to move efficiently through the environment. We have shown how Jacobians can be estimated from proprioceptive (internal) measurements, and proposed and tested two different strategies for obtaining such estimates.

The two methods, one based on constrained optimization concepts (CO-based) and one based on unscented Kalman filtering techniques (UKF-based), have been shown to perform quite differently. The UKF-based algorithm has a significantly lower computation time than the CO-based algorithm, while at the same time giving more accurate predictions of the end-effector velocity for a variety of simulation scenarios. Importantly, the UKF-based algorithm performs much better in scenarios where the snake halts, in which case the CO-based algorithm is plagued by divergence issues. This paper presents and analyses the algorithms from numerical perspectives, but we foresee further research into the stability and convergence properties of both algorithms as they are both practically identifiable. These properties are important as they would provide some guarantees to the correctness of the estimates, which in turn could prove important in the design of control strategies that leverage these estimates.

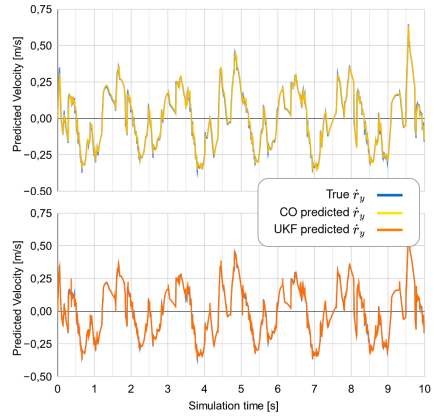
Further research should also be dedicated to devising and comparing other other types of methods for the purpose of finding which one is the most suitable for OAL. Recent advances in the modeling of snake robots using Geometric Algebra [41] may prove useful in explicitly modeling the kinematics of the robot's interaction with the obstacles, without the need of an estimation-based approach.

While this paper focuses on Jacobian matrix estimation for planar snake robots, the proposed algorithms can readily be adapted to a range of problems within robotics where the kinematics of a system are impractical or impossible to obtain analytically.

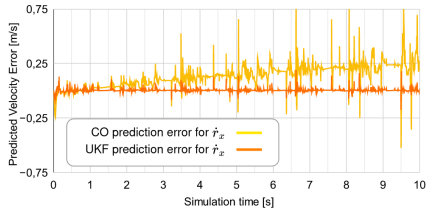
3.3.10 Supplementary Figures



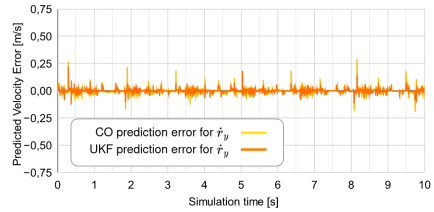
(a) Comparison of the true $\dot{\boldsymbol{r}}_x$ to the predicted values of $\dot{\boldsymbol{r}}_x$.



(b) Comparison of the true $\dot{\boldsymbol{r}}_y$ to the predicted values of $\dot{\boldsymbol{r}}_y$.

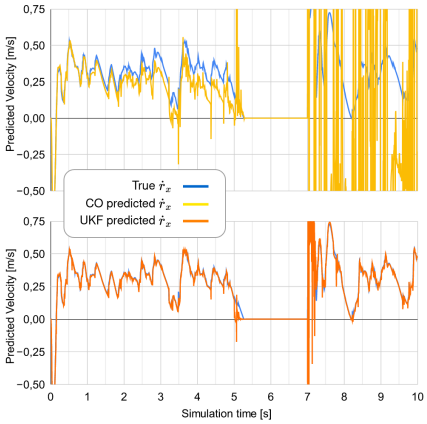


(c) The error between the true $\dot{\boldsymbol{r}}_x$ and the predicted values of $\dot{\boldsymbol{r}}_x$.

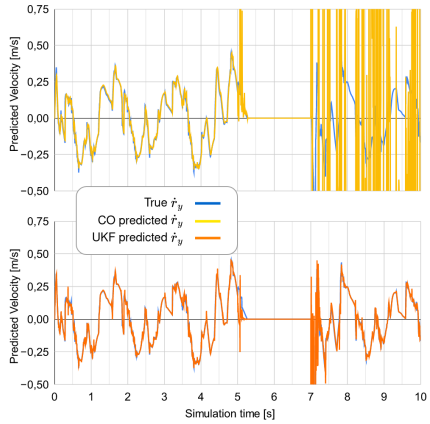


(d) The error between the true $\dot{\boldsymbol{r}}_y$ and the predicted values of $\dot{\boldsymbol{r}}_y$.

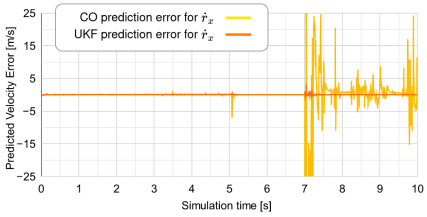
Figure 3.18: Comparison of the true $\dot{\boldsymbol{r}}$ to the predicted values of $\dot{\boldsymbol{r}}$ from the two estimators in Experiment 1. For the two topmost plots, the blue line showing the true $\dot{\boldsymbol{r}}_x$ is mostly hidden behind the orange line showing the predicted $\dot{\boldsymbol{r}}_x$. The two bottom plots show the error between the $\dot{\boldsymbol{r}}$ and the predicted values of $\dot{\boldsymbol{r}}$.



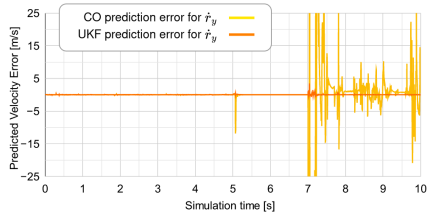
(a) Comparison of the true $\dot{\boldsymbol{r}}_x$ to the predicted values of $\dot{\boldsymbol{r}}_x$.



(b) Comparison of the true $\dot{\boldsymbol{r}}_y$ to the predicted values of $\dot{\boldsymbol{r}}_y$.

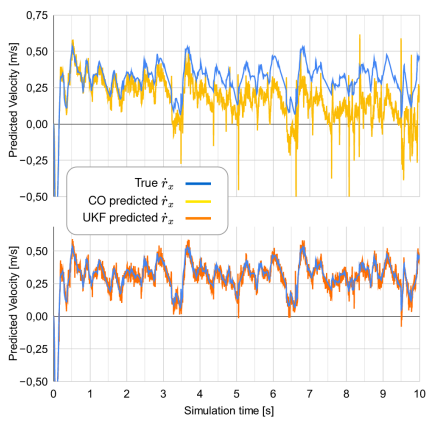


(c) The error between the true $\dot{\boldsymbol{r}}_x$ and the predicted values of $\dot{\boldsymbol{r}}_x$.

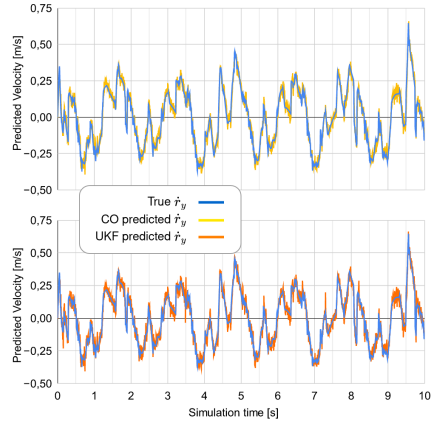


(d) The error between the true $\dot{\boldsymbol{r}}_y$ and the predicted values of $\dot{\boldsymbol{r}}_y$.

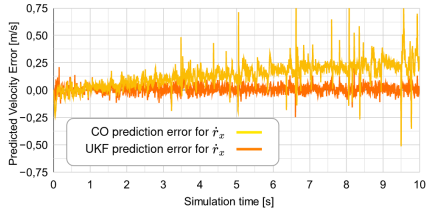
Figure 3.19: Comparison of the true $\dot{\boldsymbol{r}}$ to the predicted values of $\dot{\boldsymbol{r}}$ from the two estimators in Experiment 2. For the two topmost plots, the blue line showing the true $\dot{\boldsymbol{r}}_x$ is mostly hidden behind the orange line showing the predicted $\dot{\boldsymbol{r}}_x$. The two bottom plots show the error between the $\dot{\boldsymbol{r}}$ and the predicted values of $\dot{\boldsymbol{r}}$.



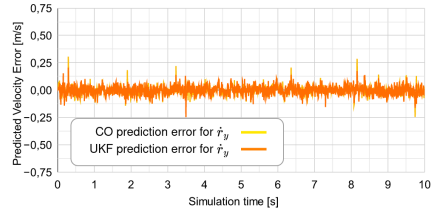
(a) Comparison of the true $\dot{\boldsymbol{r}}_x$ to the predicted values of $\dot{\boldsymbol{r}}_x$.



(b) Comparison of the true $\dot{\boldsymbol{r}}_y$ to the predicted values of $\dot{\boldsymbol{r}}_y$.



(c) The error between the true $\dot{\boldsymbol{r}}_x$ and the predicted values of $\dot{\boldsymbol{r}}_x$.



(d) The error between the true $\dot{\boldsymbol{r}}_y$ and the predicted values of $\dot{\boldsymbol{r}}_y$.

Figure 3.13: Comparison of the true $\dot{\boldsymbol{r}}$ to the predicted values of $\dot{\boldsymbol{r}}$ from the two estimators in Experiment 3. For the two topmost plots, the blue line showing the true $\dot{\boldsymbol{r}}_x$ is mostly hidden behind the orange line showing the predicted $\dot{\boldsymbol{r}}_x$. The two bottom plots show the error between the $\dot{\boldsymbol{r}}$ and the predicted values of $\dot{\boldsymbol{r}}$.

.

3.4 Article III: A Novel Model for Link Dynamics in Planar Snake Robots Using Internal Constraint Force Sensing

Abstract

We consider the problem of simplifying the typically complex task of deriving dynamical mechanical models of planar snake robots. More precisely we propose a modeling strategy that assumes the possibility of measuring the constraint forces acting between adjacent links in a snake robot, something that is now technologically possible thanks to currently available compact commercial sensors. We show how this information can be used to decouple the dynamics of each link in the snake robot, and thus to build a novel dynamic model that is simpler than the typical models in the available literature, but still powerful for predicting the movements of the robots. This implies that the proposed model may help to significantly reduce the computational complexity associated with model-based control and estimation schemes compared to other established models. Besides this, we show how the proposed model exhibits multiple properties that ease performing control, identification and state estimation tasks in general. More specifically, we show how parts of the dynamics of the model can be considered to be linear with a known non-linear exogenous disturbance which can be eliminated using feedforward control. We also show how linear Kalman Filters remain the best linear unbiased estimators for part of the state even when exogenous disturbances enter non-linearly in the system.

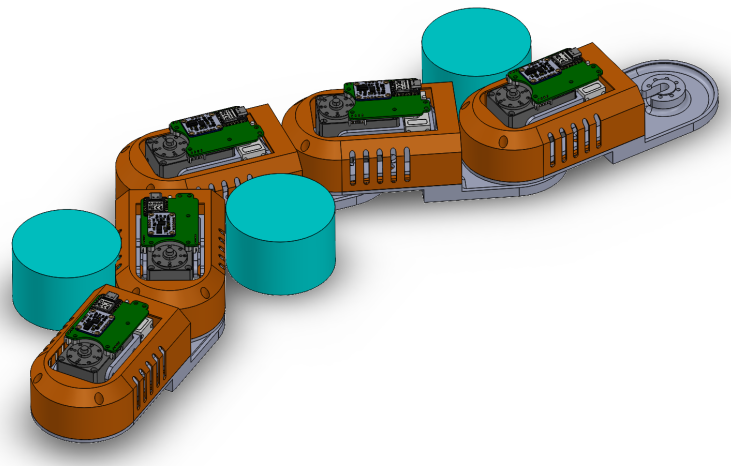


Figure 3.14: A digital render of a 5-link Boa V2 snake robot.

3.4.1 Introduction

Snake robots are mechanisms designed to mimic biological snakes, which aspire to inherit the robustness and stability properties of biological snake locomotion. Like their biological counterparts, artificial snakes move using an array of different propulsion techniques such as lateral undulation, sinus lifting and sidewinding [2]. These gaits are explained well by [14]. In principle this makes snake robots suitable for moving in and adapting to some specific unknown and challenging environments such as rubble following landslides or building collapse. As of yet, this is largely an unrealized potential. Many existing systems for Obstacle-Aided Locomotion (OAL) [5], such as in [32, 42], adapt to the environment in an implicit manner only, with little utilization of mechanical sensor information. In contrast, the present work is part of an effort to achieve efficient, robust and intelligent locomotor behavior by exploiting continually updated information about the external forces derived from constraint forces between adjacent links in the robot.

A snake robot, being a mechanical structure consisting of joints and rigid links, can be modeled as a dynamic system. Liljebäck et.al. [11] defines a complete model of the kinematics of the robot and a set of equations describing its dynamics, building on the defined kinematics. Subsequently they provide a linearization of the dynamics and investigate the controllability properties of the model. For their complete dynamic model and its derivation the reader is referred to [11].

The Boa snake robot is a novel experimental platform built for research into OAL, cf. Figure 3.14. It is the latest in a family of robots which include multi-axis force sensors in each joint of the robot, making it possible to measure the constraint forces acting between the links. Boa also includes a multi-axis Inertial Measurement Unit (IMU) in each link, making it possible to extract a wide array of sensor data from each link. The measurement system allows for exploration into alternative ways of modeling the robot. This work presents a novel dynamical model for a planar snake robot that exploits the novelties of Boa, and discusses the properties and application of the model. Specifically, the contributions of this paper include:

- A novel model that incorporates constraint force measurements from the joints of the snake robot to simplify the dynamic equations, rendering parts of the dynamics linear when modeling the internal constraint forces as known exogenous disturbances.
- A feed-forward controller to eliminate the exogenous disturbance from the rotational dynamics of the robot.
- A discretization of the novel model, which under certain conditions has the Kalman Filter as the best linear unbiased estimator for the rotational state of the model.
- A complexity analysis showing that forward computations of the novel model has a lower bound on space and computational complexity than the model in [11].

This paper is organized as follows: Section 3.4.2 reiterates some of the key notation necessary to implement the novel model. This is used in Section 3.4.3 to derive the novel model. Section 3.4.4 defines a first-order discretization of the model, while Section 3.4.5 discusses the properties of the models presented in the paper.

3.4.2 Notation

The following section reiterates some of the key notation for modeling planar snake robots, defined in [11] and Chapter 2.

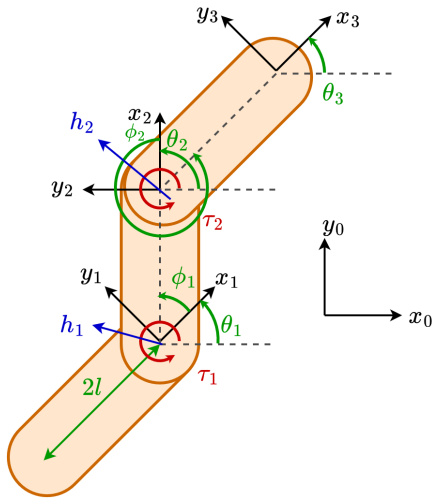


Figure 3.15: The kinematics of a simple 3-link planar snake robot as seen from above. The constraint forces h_i^i (blue) are resolved in their link-local frames (black arrows). The link angles θ_i and joint angles ϕ_i (green arcs) relate the orientations of the link-local frames. The joint torque τ_i (red) is the control input of the snake robot. The link length $2l$ is shown as a green double-headed arrow.

A visualization of the kinematics of the system under scrutiny is shown in Figure 3.15. A generic planar snake robot consists of N links connected by $N_J = N - 1$ rotational joints whose axes of rotation are all parallel. The robot exists in a world coordinate frame (x_0, y_0). Each link of the robot has its own link local coordinate frame (x_i, y_i) where i is the link number. For the remainder of this paper, an integer superscript will denote the reference frame of the variable, and a subscript denotes the link index, e.g., $\ddot{r}_{i-1}^i$ denotes the acceleration of link $i-1$ expressed in terms of the link local frame of link i.

The local frames are oriented such that the direction of the positive x_i axis coincides with the line from the axis of rotation of joint $i-1$ to that of joint i, and the y-axis points in the left transversal direction when looking in the positive x_i direction. The tail link of the robot is indexed as link 1 and the head is link N. The link angle θ_i of link i, for $i \in [1 \dots N]$, is defined as the angle between the

global axis x^0 and the local axis x^i. The angle of the ith joint is denoted as ϕ_i for $i \in [1 \dots N_J]$. In the local frames, forces and torques can be schematized as in Figure 3.15. The relation between the link angles and the joint angles is given by

$$\phi_i = \theta_{i+1} - \theta_i \tag{3.20}$$

We can then define the joint angle vector $\boldsymbol{\phi}$ and the link angle vector $\boldsymbol{\theta}$ as

$$\begin{aligned} \boldsymbol{\phi} &= \begin{bmatrix} \phi_1 & \phi_2 & \dots & \phi_{N_J} \end{bmatrix}^T \\ \boldsymbol{\theta} &= \begin{bmatrix} \theta_1 & \theta_2 & \dots & \theta_N \end{bmatrix}^T \end{aligned} \tag{3.21}$$

and relate them by the difference matrix

$$\boldsymbol{D} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & -1 \end{bmatrix} \in \mathbb{R}^{N_J \times N} \tag{3.22}$$

such that

$$\boldsymbol{\phi} = \boldsymbol{D}\boldsymbol{\theta} \tag{3.23}$$

In addition we also define the position of the snake robot's center of mass in global coordinates as $\boldsymbol{p}$.

3.4.3 A Novel Dynamical Model

The notation defined in Section 3.4.2 serves as the basis on which we derive the novel model presented here. In [11] the torque balance for a single link i is given as

$$\begin{aligned} J\ddot{\theta}_i = \tau_{i-1} - \tau_i \\ - l \sin\theta_i (h^0_{i,x} + h^0_{i-1,x}) \\ + l \cos\theta_i (h^0_{i,y} + h^0_{i-1,y}) \end{aligned} \tag{3.24}$$

where τ_i is the torque exerted by link i on link $i+1$ through joint i, $2l$ is the length of each link, and the constraint forces h are given in the world coordinate frame. The scalar J is the rotational inertia of a single link about its center of mass on the axis normal to the plane. This model assumes that torsional friction between the snake robot links and their environment is negligible, and that the external forces $\boldsymbol{f}_{e,i}$ acting on link i from the environment of the snake robot acts through the link's center of mass.

In the case that torsional friction is non-negligible and the contact forces act on an arbitrary point on the link's surface, both of these effects can be embedded in the model by introducing the external link torque for link i, denoted $\boldsymbol{\tau}_{e,i}$, such

that (3.24) would become

$$\begin{aligned} J\ddot{\theta}_i = \tau_{i-1} - \tau_i \\ - l\sin\theta_i(h^0_{i,x} + h^0_{i-1,x}) \\ + l\cos\theta_i(h^0_{i,y} + h^0_{i-1,y}) \\ + \tau_{e,i}. \end{aligned} \tag{3.25}$$

While we acknowledge the existence and effect of torsional friction and offset contact forces on the model, we choose to make the same assumptions for this model as the ones made in [11]. The effects of torsional friction can largely be mitigated through the physical design of the robot by ensuring that only a small part of the chassis of each link is in contact with the plane and by using a surface material with a low friction coefficient.

The torques arising from offset external forces $\boldsymbol{f}_{e,i}$ are known to be limited in magnitude to $l||\boldsymbol{f}_{e,i}||$ which diminishes as the length $2l$ of the links decreases. As a planar snake robot is an actuated mechanism, the control inputs τ_i are typically known. With this information it is likely possible to estimate $\tau_{e,i}$ and account for its effect on the link dynamics using an *Unknown Input Observer (UIO)*, however this remains outside the scope of this paper and is a subject for future research.

In the Boa snake robot, the joint torques τ_i, as well as the link local constraint force $h^i_{i,y}$, can be measured directly by the sensor system. With slight modifications to (3.24) we get a torque balance that uses the link local constraint forces

$$J\ddot{\theta} = \tau_{i-1} - \tau_i + l\big(h^i_{i,y} + h^i_{i-1,y}\big). \tag{3.26}$$

The scalar $h^b_{a,y}$ denotes the constraint force in the y-direction from link $a+1$ on link a through joint a resolved in the local frame of link b. While the scalar $h^i_{i,y}$ can be measured directly, the constraint force $h^i_{i-1,y}$ is not explicitly known, but can be produced by rotating the force vector $\boldsymbol{h}^{i-1}_{i-1}$ from frame $i-1$ to i by applying the rotation matrix $\boldsymbol{R}^T(\phi_{i-1})$ before decomposing the force into its x and y components. This yields

$$J\ddot{\theta}_i = \tau_i - \tau_{i-1} + l\begin{bmatrix}0 & 1\end{bmatrix}\big(\boldsymbol{h}^i_i + \boldsymbol{R}^T(\phi_{i-1})h^{i-1}_{i-1}\big), \tag{3.27}$$

which can be rewritten in matrix notation as

$$J\ddot{\boldsymbol{\theta}} = \boldsymbol{D}^T\boldsymbol{\tau} + l\boldsymbol{\eta}_y\boldsymbol{R}_h(\boldsymbol{\phi})\boldsymbol{h}, \tag{3.28}$$

where $\boldsymbol{\eta}_y$ is a coordinate selection matrix, $\boldsymbol{R}_h$ is a coordinate conversion matrix, $\boldsymbol{h}$ is the link local constraint force vector and $\boldsymbol{\tau}$ is the torque vector. Furthermore

these are defined as:

$$\boldsymbol{\eta}_y = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \dots & 0 & 1 \end{bmatrix} \in \mathbb{R}^{N \times 2N} \tag{3.29}$$

$$\boldsymbol{R}_h(\boldsymbol{\phi}) = \begin{bmatrix} \boldsymbol{I} & 0 & 0 & 0 & 0 \\ \boldsymbol{R}^T(\phi_1) & \boldsymbol{I} & 0 & 0 & 0 \\ 0 & \boldsymbol{R}^T(\phi_2) & \boldsymbol{I} & 0 & 0 \\ 0 & 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \boldsymbol{R}^T(\phi_{N(J-1)}) & \boldsymbol{I} \\ 0 & 0 & 0 & 0 & \boldsymbol{R}^T(\phi_{NJ}) \end{bmatrix} \in \mathbb{R}^{2N \times 2N_J} \tag{3.30}$$

$$\boldsymbol{h} = \begin{bmatrix} h_1^{1T} & h_2^{2T} & \dots & h_{N_J}^{N_J T} \end{bmatrix}^T \in \mathbb{R}^{2N_J} \tag{3.31}$$

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_1 & \tau_2 & \dots & \tau_{N_J} \end{bmatrix}^T \in \mathbb{R}^{N_J} \tag{3.32}$$

By defining the rotational state vector

$$\boldsymbol{q}_\theta = \begin{bmatrix} \boldsymbol{\theta} \\ \dot{\boldsymbol{\theta}} \end{bmatrix} \tag{3.33}$$

the model can be rewritten in state space form as

$$\dot{\boldsymbol{q}}_\theta = \boldsymbol{A}\boldsymbol{q}_{\boldsymbol{\theta}} + \boldsymbol{B}\boldsymbol{\tau} + \boldsymbol{C}\boldsymbol{w}_\theta(\boldsymbol{\phi}, \boldsymbol{h}), \tag{3.34}$$

where

$$\begin{aligned} \boldsymbol{A} &= \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \\ \boldsymbol{B} &= \frac{1}{J} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{D}^T \end{bmatrix} \\ \boldsymbol{C} &= \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix} \\ \boldsymbol{w}_\theta(\boldsymbol{\phi}, \boldsymbol{h}) &= \frac{l}{J} \boldsymbol{\eta}_y \boldsymbol{R}_h(\boldsymbol{\phi}) \boldsymbol{h}. \end{aligned} \tag{3.35}$$

The state space model has $\boldsymbol{q}_\theta$ as the state, $\boldsymbol{\tau}$ as the control input. The non-linear term $\boldsymbol{w}_\theta(\boldsymbol{\phi}, \boldsymbol{h})$ is further denoted $\boldsymbol{w}_\theta$ for brevity. From [11] we have that the

translational dynamics of a planar snake robot can be modeled as

$$Nm\ddot{\boldsymbol{p}} = \sum \boldsymbol{f}_{e,i} \tag{3.36}$$

The Boa snake robot comprises a sensor system intended to estimate the link local external forces $\boldsymbol{f}^i_{e,i}$. These can be transformed into global coordinates by the transformation $\boldsymbol{R}(\theta_i)$ as

$$\boldsymbol{f}_{e,i} = \boldsymbol{R}(\theta_i)\boldsymbol{f}^i_{e,i} \tag{3.37}$$

The translational dynamics of the robot can then be written in matrix notation as

$$\ddot{\boldsymbol{p}} = \frac{1}{Nm}\boldsymbol{R}_e(\theta)\boldsymbol{f}_e \tag{3.38}$$

where

$$\begin{aligned} \boldsymbol{R}_e(\theta) &= \begin{bmatrix} \boldsymbol{R}(\theta_1) & \dots & \boldsymbol{R}(\theta_N) \end{bmatrix} \\ \boldsymbol{f}_e &= \begin{bmatrix} \boldsymbol{f}^i_{e,i} \\ \vdots \\ \boldsymbol{f}^N_{e,N} \end{bmatrix}. \end{aligned} \tag{3.39}$$

By defining the translational state $\boldsymbol{q}_p$ of the snake robot as

$$\boldsymbol{q}_p = \begin{bmatrix} \boldsymbol{p} \\ \dot{\boldsymbol{p}} \end{bmatrix} \tag{3.40}$$

we can define rewrite (3.38) in state space form as

$$\dot{\boldsymbol{q}}_p = \boldsymbol{A}\boldsymbol{q}_p + \frac{1}{Nm}\boldsymbol{C}\boldsymbol{R}_e(\theta)\boldsymbol{f}_e \tag{3.41}$$

where the matrices $\boldsymbol{A}$ and $\boldsymbol{C}$ are the same as the ones defined in (3.35). By combining the translational dynamics in (3.41) and the rotational dynamics in (3.34), we get the full dynamics of a planar snake robot with the state vector

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_\theta \\ \boldsymbol{q}_p \end{bmatrix} \tag{3.42}$$

as

$$\begin{aligned} \dot{\boldsymbol{q}}_\theta &= \boldsymbol{A}\boldsymbol{q}_{\boldsymbol{\theta}} + \boldsymbol{B}\boldsymbol{\tau} + \boldsymbol{C}\boldsymbol{w}_\theta(\boldsymbol{\phi}, \boldsymbol{h}) \\ \dot{\boldsymbol{q}}_p &= \boldsymbol{A}\boldsymbol{q}_p + \frac{1}{Nm}\boldsymbol{C}\boldsymbol{R}_e(\theta)\boldsymbol{f}_e \end{aligned} \tag{3.43}$$

3.4.4 Discretization of the Model

Assume that it is possible to sample the joint angles $\boldsymbol{\phi}$, the joint torques $\boldsymbol{\tau}$ and link local constraint forces $\boldsymbol{h}$ at a regular time interval δ_t. We denote each sampling timestep $k \in \mathbb{Z}$ and every sampled value at time $k\delta_t$ with the postscript $[k]$, e.g such that $\boldsymbol{h}[k] = \boldsymbol{h}(k\delta_t)$, and can then define

$$\boldsymbol{w}[k] = \boldsymbol{w}(\boldsymbol{\phi}[k], \boldsymbol{h}[k]) \tag{3.44}$$

An approximate discretized version of the model can be produced by assuming a sufficiently small time step δ_t that a first order discretization accurately represents the systems dynamics. In the following derivation, it is assumed that

$$\dot{\boldsymbol{q}} \approx \frac{\boldsymbol{q}[k+1] - \boldsymbol{q}[k]}{\delta_t} \tag{3.45}$$

such that Euler's method can be applied. We define the discretized stochastic state space model as

$$\begin{aligned} \boldsymbol{q}_\theta[k+1] &= \boldsymbol{A}_d \boldsymbol{q}_\theta[k] + \boldsymbol{B}_d \boldsymbol{\tau}[k] + \boldsymbol{C}_d \boldsymbol{w}[k] \\ \boldsymbol{q}_p[k+1] &= \boldsymbol{A}_d \boldsymbol{q}_p[k] + \boldsymbol{C}_d \frac{1}{Nm} \boldsymbol{R}_e(\boldsymbol{\theta}[k]) + \boldsymbol{f}_e[k] \end{aligned} \tag{3.46}$$

with

$$\begin{aligned} \boldsymbol{A}_d &= \boldsymbol{I} + \delta_t \boldsymbol{A} \\ \boldsymbol{B}_d &= \delta_t \boldsymbol{B} \\ \boldsymbol{C}_d &= \delta_t \boldsymbol{C} \end{aligned} \tag{3.47}$$

where $\boldsymbol{q}_\theta[k]$ and $\boldsymbol{q}_p[k]$ are the discrete approximations of $\boldsymbol{q}_\theta$ and $\boldsymbol{q}_p$, respectively, at timestep $\boldsymbol{k}$.

3.4.5 Properties of the Novel Model

In this section we investigate the properties of the models defined in (3.43) and (3.46) with their associated definitions.

Cascaded structure of the model

The structure of the state space model in (3.43) is shown in Fig. 3.16. The system can be modeled as two cascaded systems, with $\boldsymbol{\tau}$ as the input to the rotational dynamics, and its output $\boldsymbol{\theta}$ as the input to the translational dynamics. The rotational dynamics has an implicit non-linear dependency upon its own state $\boldsymbol{q}_\theta$ through the joint angles $\boldsymbol{\phi}$. However, in the likely case that $\boldsymbol{\phi}$ can be measured explicitly, the term $\boldsymbol{w}_\theta$ can be seen as a known exogenous disturbance term. In this case the rotational dynamics can be considered linear with respect to to its own state $\boldsymbol{q}_\theta$. The incorporation of the constraint forces $\boldsymbol{h}$ into the model removes the model's dependency on the angular speeds $\dot{\boldsymbol{\phi}}$ and $\dot{\boldsymbol{\theta}}$, resulting in dynamic

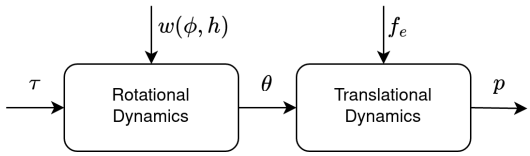


Figure 3.16: A schematic of the dynamics defined in (3.43).

equations which are not explicitly affected by centripetal forces or the Coriolis effect. Unlike the model given in [11], the rotational dynamics are independent of the external link forces $\boldsymbol{f}_e$ under the same assumptions.

Implicit enforcement of constraints

Like the established model in [11], the novel model is built upon the generalized coordinates defined in (3.42). Although the dynamics of the links are apparently separated, they are in fact connected by the identity $\boldsymbol{h}_i^i = -\boldsymbol{R}^T(\phi_{i-1})\boldsymbol{h}_{i-1}^{i-1}$, i.e. the choice of generalized coordinates implicitly enforces the kinematic constraints imposed on the model.

Elimination of exogenous disturbances from the rotational dynamics

The study of systems with exogenous disturbances such as the rotational dynamics in (3.43) is well developed, and is summarized in [43]. As the disturbances are known, it is possible to define a feed forward controller

$$\boldsymbol{\tau} = \boldsymbol{K}_w \boldsymbol{w}_\theta + \boldsymbol{\tau}^* \tag{3.48}$$

to remove the effect of the disturbances on the system, where $\boldsymbol{\tau}^*$ is the desired net torque on the joints of the robot. We choose the feed forward matrix $\boldsymbol{K}_w$ to solve

$$\boldsymbol{B}\boldsymbol{K}_{\boldsymbol{w}}\boldsymbol{w}_\theta + \boldsymbol{C}\boldsymbol{w}_\theta = 0. \tag{3.49}$$

By substituting the matrices in (3.35) the problem reduces to

$$\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{K}_{\boldsymbol{w}} + \boldsymbol{I} = 0 \tag{3.50}$$

As the product $\boldsymbol{D}^T\boldsymbol{D}$ is non-singular, it holds that the Moore-Penrose inverse of $\boldsymbol{D}^T$ which is denoted $(\boldsymbol{D}^T)^\dagger$ solves (3.50) wrt. $\boldsymbol{K}_w$. The resulting controller is

$$\boldsymbol{\tau} = \boldsymbol{\tau}^* - (\boldsymbol{D}^T)^\dagger \boldsymbol{w}_\theta. \tag{3.51}$$

We can now plug this feedforward controller into the model in (3.43) resulting in an undisturbed linear system

$$\dot{\boldsymbol{q}}_\theta = \boldsymbol{A}\boldsymbol{q}_\theta + \boldsymbol{B}\boldsymbol{\tau}^*. \tag{3.52}$$

Noise properties of the feedforward controller

In this subsection we investigate the properties of the discrete model in (3.46) with the combined feedforward controller defined in Section 3.4.5 when the exogenous disturbance $\boldsymbol{h}[k]$ is subject to noise. Assume that the joint torque and link local constraint force measurements are affected by zero-mean Gaussian noises with known covariance $\boldsymbol{\sigma}_h^2$ such that the measured link local constraint forces $\tilde{\boldsymbol{h}}[k]$ at timestep k can be defined as

$$\tilde{\boldsymbol{h}}[k] \sim \mathcal{N}(\boldsymbol{h}[k], \boldsymbol{\sigma}_h^2). \tag{3.53}$$

This in turn allows us to define the noisy exogenous disturbance $\tilde{\boldsymbol{w}}_\theta[k]$ as

$$\tilde{\boldsymbol{w}}_\theta[k] = \boldsymbol{w}_\theta(\boldsymbol{\phi}[k], \tilde{\boldsymbol{h}}[k]). \tag{3.54}$$

As the Gaussian quality of the noise does not change under linear transformations, the distribution of $\tilde{\boldsymbol{w}}_\theta[k]$ is known to be Gaussian with a covariance and expectation that can be calculated from (3.34) as

$$\tilde{\boldsymbol{w}}_\theta[k] \sim \mathcal{N}(\boldsymbol{w}_\theta[k], \boldsymbol{\sigma}_w^2) \tag{3.55}$$

$$\boldsymbol{\sigma}_w^2 = l^2 \boldsymbol{\eta}_y \boldsymbol{R}_h(\boldsymbol{\phi}) \boldsymbol{\sigma}_h^2 (\boldsymbol{\eta}_y \boldsymbol{R}_h(\boldsymbol{\phi}))^T. \tag{3.56}$$

These calculations assume that the joint angles $\boldsymbol{\phi}$ has negligible noise, which may be a reasonable assumption in most systems, as high resolution rotary encoders are known to purvey little-to-no measurement noise. By inserting the controller from (3.48) using the noisy disturbance $\tilde{\boldsymbol{w}}_\theta[k]$ into the rotational dynamics from (3.46), we get the system

$$\begin{aligned} \boldsymbol{q}_\theta[k+1] = \; & \boldsymbol{A}_d \boldsymbol{q}_\theta[k] \\ & + \boldsymbol{B}_d(\boldsymbol{\tau}^* - (\boldsymbol{D}^T)^\dagger)\tilde{\boldsymbol{w}}_\theta[k] \\ & + \boldsymbol{C}_d \boldsymbol{w}_\theta[k], \end{aligned} \tag{3.57}$$

which reduces to a linear system with Gaussian zero mean noise

$$\boldsymbol{q}_\theta[k+1] = \boldsymbol{A}\boldsymbol{q}_\theta[k] + \boldsymbol{B}_d\boldsymbol{\tau}^*[k] + \boldsymbol{\xi}[k] \tag{3.58}$$

with

$$\boldsymbol{\xi}[k] \sim \mathcal{N}(0, \boldsymbol{\sigma}_w^2). \tag{3.59}$$

Equation (3.58) represents a linear process model with zero-mean Gaussian noise as defined in [44], which is used to form the basis for a *Kalman Filter*.

Optimality of the Kalman Filter as the rotational state estimator

We now introduce an observation model $\boldsymbol{z}_\theta[k]$ that makes it possible to measure the rotational state $\boldsymbol{q}_\theta[k]$ as a linear combination of the states through the measurement function

$$\boldsymbol{z}_\theta[k] = \boldsymbol{H}\boldsymbol{q}_\theta[k] + \boldsymbol{v}[k], \tag{3.60}$$

where

$$\boldsymbol{v}[k] \sim \mathcal{N}(0, \boldsymbol{\sigma}_v^2). \tag{3.61}$$

One such implementation of $\boldsymbol{H}$ and $\boldsymbol{v}[k]$ is found in the Boa snake robot, where it is possible to measure the joint angles $\boldsymbol{\phi}$ through rotary encoders in the joints of the robot, and $\boldsymbol{q}_\theta$ using a gyroscope and magnetometer embedded in each link. The resulting measurement model then becomes

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{D} & \boldsymbol{0} \\ \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \tag{3.62}$$

For the discrete model in (3.58) with a measurement model structure as in (3.60) resulting in an observable system, the Kalman Filter is the *Best Linear Unbiased Estimator (BLUE)* for the state $\boldsymbol{q}_\theta[k]$ [44]. The implication of this is that, even if the model (3.46) is not linear wrt. $\boldsymbol{h}[k]$ and $\boldsymbol{\phi}[k]$, the Kalman Filter is still the BLUE estimator for the state $\boldsymbol{q}_\theta[k]$ if the constraint force measurements $\tilde{\boldsymbol{h}}[k]$ are unbiased and Gaussian. The unbiasedness and optimality of the estimator is dependent on that the measurements of the joint angle $\boldsymbol{\phi}$ are known and near noiseless.

Computational complexity

The following subsection uses big-$\mathcal{O}$ notation as defined in [45]. A property of the novel model is its ability to reduce computational complexity in calculations. Consider the state space model defined in (3.43). The band sparse structure of the matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{\eta}_y\boldsymbol{R}_h(\boldsymbol{\phi})$ arises from the fact that the constraint force measurements decouples the dynamics of the links, making the dynamics of a single link independent of the dynamics of the adjacent links in the robot. Band sparse matrices have a number of non-zero elements limited by $\mathcal{O}(N)$ as opposed to the matrices used in the model in [11] where most matrices are dense and the number of non-zero elements is limited by $\mathcal{O}(N^2)$. The matrix $\boldsymbol{R}_e$ has a number of non-zero elements limited by $\mathcal{O}(N)$ even if it is not band-sparse.

For the model in (3.43), the computation of $\dot{\boldsymbol{q}}$ is upper bounded in computational complexity by the matrix multiplication operation which is normally bounded by $\mathcal{O}(N^3)$. As all matrices included in this computation are bandsparse, the computational complexity of $\dot{\boldsymbol{q}}$ can be reduced to $\mathcal{O}(N^2)$ [46]. Similarly, the space complexity of the model is upper bounded by $\mathcal{O}(N)$. The savings in com-

putation and space complexity using the novel model can be useful when running control or state estimation algorithms on systems limited in storage and computational power such as in embedded microcontrollers.

3.4.6 Conclusions

We derived and proposed a novel model for planar snake robots that has linear rotational dynamics with known exogenous disturbances, and shown how these disturbances may be eliminated using rudimentary feed forward control actions. We have also discussed how a stochastic discretization of the model can be used to prove that, under certain conditions, linear Kalman Filtering is the minimum variance linear unbiased estimation technique for the rotational state of the robot. Finally, we have shown how the novel model leads to lower computational and space complexity requirements than using other established models.

CHAPTER

FOUR

FORM CLOSURE FOR ROBUST LOCOMOTION IN SNAKE ROBOTS

4.1 Introduction

The following chapter is dedicated to the theory of form closure and its application in snake robots. We start by examining the underlying problem that led to form closure being a relevant topic in snake robotics.

The control of snake robots is a complex task. A snake robot is under-actuated in that its joints may be actuated directly, but its position and orientation in its environment constitute unactuated degrees of freedom. The robots' interaction with obstacles in its environment further complicates its kinematics by introducing non-holonomic contact constraints. When the robot is actuated, it is important to determine whether the consequent movement will cause the robot to depart from a contact point due to its unactuated dynamics.

Thus arose the need to identify a subset of the robots' configurations where its unactuated dynamics are constrained by obstacles in its environment. By restraining the motion of the robot to this subset, its kinematics can be simplified, in order to make OAL control strategies more tractable.

Form closure is a concept developed from the field of robotic gripping, where it is used as a geometric condition on a robots grasp of an object, to ensure that the object is geometrically locked between the "fingers" so that it cannot slip or fall out of the gripper [47]. Form closure conceptually applies to snake robots in the same manner as a condition to ensure that a snake robot cannot slip away from the obstacles in its environment.

The articles enclosed in this chapter elaborate further on this concept. Article IV adapts the theory of form closure to snake robots, while Article V builds on

these findings to identify regions of the robots' configuration space which has the aforementioned properties. Prior to the articles, Section 4.2 constitutes a tutorial that approaches the subject of form closure from a more pedagogical perspective to make the subject matter available for a broader readership.

The tutorial will naturally have some overlap with Article IV and Article V as they cover many of the same topics but does not include any novel research

Finally, in Section 4.5, we outline a method for analytic computation of form closed configurations. In previous literature this has typically been acheived by numeric optimization.

4.2 A Tutorial on Form Closure

This chapter serves as a general introduction to the theory of form closure before proceeding to its application to snake robots. Form closure, is the problem of finding configurations where a rigid body is fixed in place by its contact with its environment. The literature on form closure can seem scattered, and the topic is oftentimes only covered briefly in books and other teaching materials. It is covered in great detail in academic research articles, however they can seem daunting without sufficient background knowledge.

The section is written in the form of a tutorial, where the focus remains on promoting intuition and giving ample explanations and examples along the way, while maintaining some level of mathematical rigor. In the tutorial we will generally not include proofs derived from other academic literature, and rather refer to the original text, to keep the tutorial slim and concise. The tutorial is aimed at readers with an undergraduate level knowledge of robotics and its adjoined topics, and the language style is more colloquial than what might be expected of an academic manuscript

In this tutorial we will investigate form closure, starting at a very conceptual level, as a motivation for further reading. We will then review some important preliminary theory that is essential to understanding the remainder of the tutorial. Finally we will approach form closure from a more mathematically formal perspective, and discuss how we can make numerically feasible computations on form closure.

The following material is primarily based on the works of [47–54]. Other works are referred to in the text when necessary.

4.2.1 Introduction

Form closure has been a subject of intensive study for the past decades. The general concepts date back to the initial studies of Releoux in the late 1800's [55] on the immobilization of rigid bodies. The term form closure was coined in the early 90's [47] as the development and study of robotic grasping and manipulation became an active field of study and reached its maturity. It has proved to be important in the theory on prehensile robots, in computing suitable grasps for picking up and manipulating objects of different geometries.

To begin with, the core concepts of form closure might best be explained by imagining a little game: We are sitting in front of a level cork-board. I fetch a piece of cardboard, and cut any shape I'd like from the cardboard and place it on the board. You receive a box of thumb tacks. Your goal is to stick the thumb tacks into the board, so that the piece of cardboard is completely immobile. The catch is that you are not allowed to pierce the cardboard, you can only place the tacks so that they touch the cardboard on the sides. You win if there is no way i can twist, turn or slide the piece of cardboard across the cork-board. Sounds easy enough? Although the concept is alluringly simple, it can quickly become complex when we add some more rules to our game. Here is some food for thought:

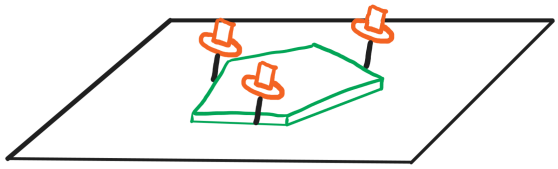


Figure 4.1: A piece of cardboard (green) on a level cork-board. If the cardboard piece is immobilized by the tacks, it is said to be under form closure.

- What if we limited the number of tacks you could use to 4 tacks. Will it still be possible to find a placement of the tacks so that you always win the game?
- What if we limited the number of tacks you could use to 3 or even 2 tacks. Could you still always win the game, regardless of the shape i cut?
- Is there any cardboard shape i can cut which makes it impossible for you to win the game? i.e. is there a shape that cannot be immobilized?
- What if I can only cut cardboard shapes with straight sides (i.e. a polygon). Does this change the difficulty of the game for either of the two players?

As a little spoiler, we can reveal that when the cardboard piece is immobilized by the tacks, it is said to be under form closure by the tacks. As we will delve into the subject of form closure, we will also answer the hypotheticals above. Before we start our journey we need to visit some fundamental theory that will be useful in our study on form closure.

4.2.2 Preliminaries

In this subsection we review some of the mathematical foundation necessary to understand the rest of this tutorial. These are all well-studied topics, and there is a vast and comprehensive literature covering all of them. Thus, we will only cover each topic superficially

The Homogeneous Transformation and C-space

Our journey starts with a rigid body. The rigid body inhabits a d-dimensional euclidean space $\mathbb{R}^d$ called the *workspace* that we will denote as $\mathcal{W}$ where $\mathcal{W} = \mathbb{R}^d$. As form closure theory is intended to apply to real-world physical objects, we can safely limit our studies to cases where $\mathcal{W} = \mathbb{R}^2$ or $\mathcal{W} = \mathbb{R}^3$. We define the initial configuration of the body as a region $\mathcal{B}_0 \subset \mathcal{W}$ that the body occupies.

We can define a *transformation* $f : \mathcal{W} \mapsto \mathcal{W}$ that maps a point in the workspace to another. We can apply the transformation to our body $\mathcal{B}_0$ by computing the *image* $f(\mathcal{B}_0) \subset \mathcal{W}$, that gives us a new region that the body now occupies. In general a transformation might stretch and deform the body in the process. We, however, are concerned with rigid bodies that, by their very definition, cannot be stretched or deformed.

The set of all transformations that maintains the shape of the body are called the *homogeneous transformations*. When the body is subject to a homogeneous transformation, it is simply rotated and translated in space without deforming or stretching. Because of this property, these transformations are often referred to in the literature on robots as *rigid transformations*. For the remainder of this tutorial we will denote these by $\mathcal{H} : \mathcal{W} \mapsto \mathcal{W}$.

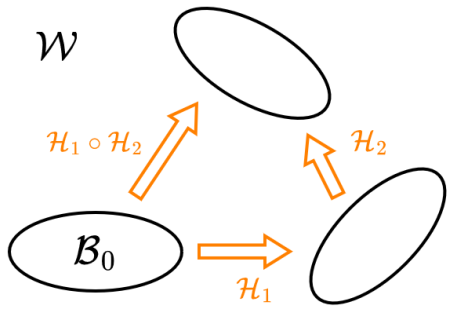


Figure 4.2: An illustration of how a body $\mathcal{B}_0 \subset \mathbb{R}^2$ might be transformed by two homogeneous transformation. Note how the two transformations compose. All the transformations are also invertible.

The set of all homogeneous transformations on the workspace is named the *Special Euclidean Group* or simply $SE(d)$, allowing us to define that $\mathcal{H} \in SE(d)$. The word "group" is not used arbitrarily, as the homogeneous transformations indeed form a mathematical group under the composition operator denoted by the symbol "$\circ$". This imbues $SE(d)$ with some interesting properties; the most important one being that when we compose two homogeneous transformations, we always get a third homogeneous transformation. Formally:

$$\mathcal{H}_1 \circ \mathcal{H}_2 \in SE(d) \quad | \quad \mathcal{H}_1, \mathcal{H}_2 \in SE(d). \tag{4.1}$$

The homogeneous transformations have an identity element called the *identity transformation* which is often abbreviated as id. This transformation intuitively is the transformation that does not change the configuration of the rigid body at all. Another important property is that every transformation $\mathcal{H}$ has an inverse $\mathcal{H}^{-1}$ such that

$$\mathcal{H} \circ \mathcal{H}^{-1} = \text{id}. \tag{4.2}$$

Before we proceed we want to take a look at two other groups, namely the *Special Orthogonal Group* $SO(d)$ and the *Translational Group* $T(d)$. The first of the two should look familiar, as it is the group containing all rotations. The latter is the group of all transformations that simply translate the body to a new position in space. The homogeneous transformations can be described as a composition of

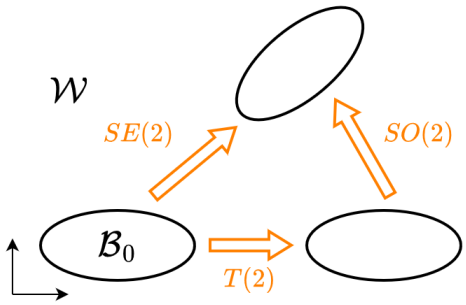


Figure 4.3: An illustration of how a transformation in $SE(2)$ might be composed from a translation in $T(2)$ and a rotation in $SO(2)$.

these two: First a rotation, then a translation. Formally, we can write this as:

$$\mathcal{H} = \mathbf{R} \circ \boldsymbol{t} \quad | \quad \mathbf{R} \in SO(d), \boldsymbol{t} \in T(d). \tag{4.3}$$

This allows us to write the homogeneous transformation in a very elegant way, in the form of a linear transformation. We can define the homogeneous transform in its canonical form as

$$\mathcal{H}(\boldsymbol{w}) = \mathbf{R}(\boldsymbol{\psi})\boldsymbol{w} + \boldsymbol{r} \tag{4.4}$$

where $\boldsymbol{w} \in \mathcal{W}$, $\mathbf{R}$ is a rotation matrix and $\boldsymbol{r}$ is a translation vector. The vector $\boldsymbol{\psi}$ contains the parameters for the rotation matrix $\mathbf{R}$. We see now that every homogeneous transformation can be built from a rotation matrix and a translation vector. As such, we can parameterize every homogeneous transformation by a generalized coordinate vector $\boldsymbol{q}$ that we define as

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{\psi} \\ \boldsymbol{w} \end{bmatrix}. \tag{4.5}$$

We call the space $\mathcal{Q}$ that $\boldsymbol{q}$ inhabits, i.e. that $\boldsymbol{q} \in \mathcal{Q}$, the *configuration space* or *c-space* for short. The c-space is also a euclidean space, in that $\mathcal{Q} = \mathbb{R}^k$ where $k = 3$ for a 2-dimensional workspace and $k = 6$ for a 3-dimensional workspace. This makes sense as $\boldsymbol{q}$ consists of one angle and two spatial coordinates in the plane, but when we expand the problem to 3-dimensional space, we require three angular parameters to represent the orientation of the body and three spatial coordinates.

Intuitively, every point $\boldsymbol{q} \in \mathcal{Q}$ represents a single unique transformation $\mathcal{H}$ and vice versa. The c-space and homogeneous transformations are in fact *diffeomorphic*. A diffeomorphism between two spaces, which we denote by the symbol $\cong$, means that there exists a one-to-one relation between the two spaces, that is not only continuous, but also differentiable. That two spaces are diffeomorphic implies that they share many properties up to the point where they can almost

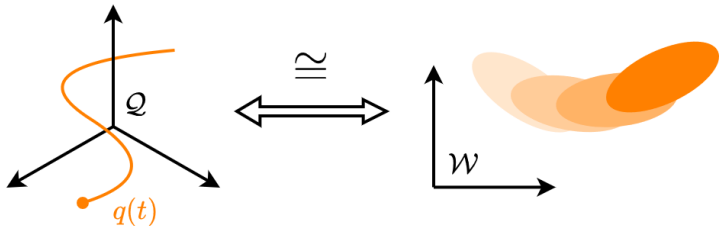


Figure 4.4: An illustration of how a continuous c-space path $\boldsymbol{q}(t) \subset \mathbb{R}^3$ might represent the continuous motion of a rigid body in $\mathbb{R}^2$.

be considered identical. Because of this nice property, a proof that applies to one space, oftentimes applies to all spaces it is diffeomorphic to as well. In our case, the configuration space and the homogeneous transformations are diffeomorphic so that we can write

$$\mathcal{Q} \cong SE(d). \tag{4.6}$$

This implies that a continuous path $\boldsymbol{q}(t) : \mathcal{T} \mapsto \mathcal{Q}$ through c-space represents a continuous rigid motion of a body in our workspace $\mathcal{W}$. Intuitively, any physically realizable rigid motion can be represented as a continuous path through c-space. A discontinuity in the path $\boldsymbol{q}$ would imply a instantaneous jump from one location to another, that does not make sense for a physical body.

Twist and twist-space

As we have already established, we can define continuous motions of rigid bodies as continuous paths in c-space. This raises a question: Now that we have defined all possible continuous motions on the rigid body, is it possible to find its derivative, i.e. something that represents the instantaneous motion of the body? This type of instantaneous motion is called a *twist* and is denote by ξ. We define a twist as:

$$\xi(\boldsymbol{w}) = [\dot{\boldsymbol{\psi}}]_\times \boldsymbol{w} + \dot{\boldsymbol{r}} \tag{4.7}$$

where the bracket notation $[\cdot]_\times$ denotes the skew symmetric matrix of a given vector. In the same way that the homogeneous transformations were parameterized by an angle $\boldsymbol{\phi}$ and a translation $\boldsymbol{r}$, the twist is parameterized by an angular velocity $\dot{\boldsymbol{\phi}}$ and a linear velocity $\dot{\boldsymbol{r}}$. The twists inhabit the space $\mathfrak{se}(d)$ i.e. that $\xi \in \mathfrak{se}(d)$. Unlike the homogeneous transformation, the twists ξ takes a point in $\mathcal{W}$ as an input and returns the velocity of the point instead of a new position.

Say one applies a twist ξ to a body $\mathcal{B}_0 \subset \mathcal{W}$, where a point in the body is denoted $\boldsymbol{b}_0 \in \mathcal{B}_0$. The resulting velocity vector $\xi(\boldsymbol{b}_0)$ is the linear velocity of the point in the body when the twist is applied. The twists can also be described

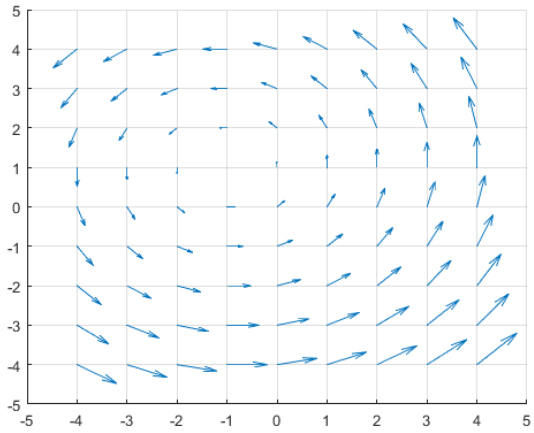


Figure 4.5: A quiver plot of the twist parameterized by $\dot{\psi} = 1$ and $\dot{\boldsymbol{r}} = [1, 1]$ in the space $\mathbb{R}^2$. All twists where $\dot{\psi} \neq 0$ form this kind of vortex-like radially symmetric vector field.

using a generalized coordinate

$$\dot{\boldsymbol{q}} = \begin{bmatrix} \dot{\psi} \\ \dot{\boldsymbol{r}} \end{bmatrix} \tag{4.8}$$

that inhabits a space $\dot{\mathcal{Q}}$, i.e. that $\dot{\boldsymbol{q}} \in \dot{\mathcal{Q}}$, that we call *twist-space*. The twist-space is diffeomorphic to $\mathfrak{se}(d)$. In other words:

$$\dot{\mathcal{Q}} \cong \mathfrak{se}(d). \tag{4.9}$$

This implies that any twist can be represented by a point in $\dot{\mathcal{Q}}$ in the same way that any homogeneous transformation may be represented as a point in $\mathcal{Q}$. A quiver-plot of a twist in $\mathcal{W} = \mathbb{R}^2$ is shown in Figure 4.5 and shows how a twist can be visualized as a flower-like radially symmetrical vector field.

In the same way as any path in space might be represented as the integral of velocities, any homogeneous transformation might be represented as the integral of twists starting from an initial configuration $\boldsymbol{q}_0$ such that

$$\boldsymbol{q}(t) = \int \dot{\boldsymbol{q}}(t)\mathrm{dt} + \boldsymbol{q}_0. \tag{4.10}$$

Hyperplanes, Half-spaces and Polyhedral Cones

In this subsection we will look at polyhedral cones; an important topic in the definition of form closure. Before defining what a polyhedral cone is, we will first tackle hyperplanes and half-spaces.

We define the boundary of a region by the operator δ. Formally a hyperplane is a euclidean subspace $\delta\mathcal{N}$ whose dimensionality is exactly one less than the euclidean space it resides in. As an example, a hyperplane in $\mathcal{X}$ where $\mathcal{X} = \mathbb{R}^3$ would be a plane. In $\mathbb{R}^2$ it would be a line, and in $\mathbb{R}^4$ it would be a volume. A hyperplane must contain the origin and stretches infinitely far into the horizon of its ambient space in all directions. They are typically defined by a unit normal vector $\hat{\boldsymbol{n}} \in \mathcal{X}$ such that

$$\delta\mathcal{N} = \{\boldsymbol{x} \in \mathcal{X} \quad | \quad \hat{\boldsymbol{n}} \cdot \boldsymbol{x} = 0\} \tag{4.11}$$

where the operator $(\cdot)$ denotes the dot product. A half-space is created by dividing the ambient space $\mathcal{X}$ in half with a hyperplane. The resulting half-space is called $\mathcal{N}$ and has $\delta\mathcal{N}$ as its boundary.

The half space consists of any point residing on the same side of the hyperplane as the normal vector. Formally we can define a half-space $\mathcal{N}$ as

$$\mathcal{N} = \{\boldsymbol{x} \in \mathcal{X} \quad | \quad \hat{\boldsymbol{n}} \cdot \boldsymbol{x} \geq 0\}. \tag{4.12}$$

The half-space $\mathcal{N}$ parameterized by a given a unit vector $\hat{\boldsymbol{n}}$ can be written in short hand as

$$\mathcal{N} = \text{half}(\hat{\boldsymbol{n}}). \tag{4.13}$$

With some foundational knowledge of hyperplanes and half-spaces we can tackle the subject of polyhedral cones. The general structure of a polyhedral cone is more or less given in its name, and might best be understood through its definition. Assume we have a set of half-spaces where the i'th half-space is defined by $\mathcal{N}_i = \text{half}(\hat{\boldsymbol{n}}_i)$. A polyhedral cone $\mathcal{V}$ is defined as the intersection between the half-spaces. Formally:

$$\mathcal{V} = \bigcap_i \mathcal{N}_i = \bigcap_i \text{half}(\hat{\boldsymbol{n}}_i). \tag{4.14}$$

This definition can quite aptly be described by a homogeneous linear inequality, by interpreting $\hat{\boldsymbol{n}}_i$ as a row vector, and $\boldsymbol{x}$ as a column vector and defining

$$\mathcal{V} = \{\boldsymbol{x} \in \mathcal{X} \quad | \quad \boldsymbol{N}\boldsymbol{x} \geq 0\} \quad | \quad \boldsymbol{N} = \begin{bmatrix} \hat{\boldsymbol{n}}_1 \\ \vdots \\ \hat{\boldsymbol{n}}_i \end{bmatrix}. \tag{4.15}$$

Depending on the structure and rank of $\boldsymbol{N}$, a polyhedral cone might take one of many different shapes, such as a point, a ray, a line, a hyperplane or a closed region of its ambient space, as shown in Figure 4.6. A polyhedral cone is convex by definition, and is also unbounded except for the case when the cone is simply a point. Any polyhedral cone also always contains the origin $\mathbf{0}$ so that $\mathbf{0} \in \mathcal{V}$.

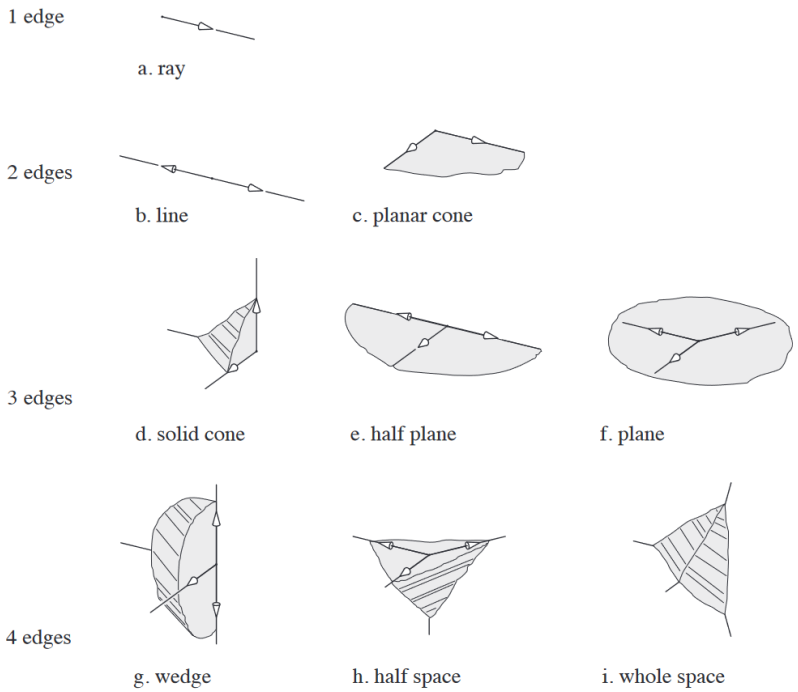


Figure 4.6: Varieties of polyhedral cones in $\mathbb{R}^3$, courtesy of [56].

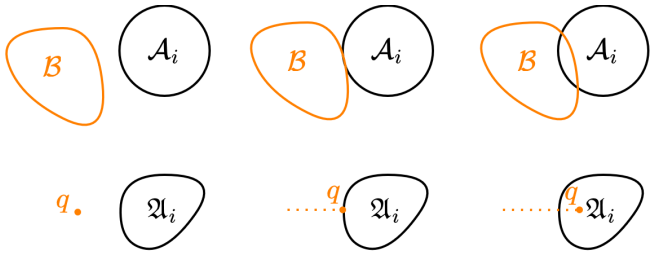


Figure 4.7: An object $\mathcal{B}$ in three different configurations, and their equivalent representations in c-space. In the left figure, the object and the obstacle do not interact. In the center figure they overlap, but only at the border, meaning that the two bodies are colliding. The right figure shows a penetrating configuration in that the two bodies overlap.

4.2.3 A Formal Introduction to Form Closure

Form closure studies the mobility of an object occupying the region $\mathcal{B}_0$, in contact with fixed *obstacles* that are open, bounded regions denoted $\mathcal{A}_i \subset \mathcal{W}$. Modeling the obstacles as open regions may not make physical sense, but is common practice in literature on path planning. By doing this, a body sharing a boundary with an obstacle (i.e. they physically touch) do not overlap. In much of the existing literature, the obstacles are referred to as *finger bodies*. The name derives from form closure's origins in analyzing grasps of robot fingers. As the term is quite

limiting in scope, we choose to opt for the more general term obstacles instead.

The obstacles might represent any rigid bodies fixed to the robots' environment. In its most general form, the goal of form closure is to identify combinations of object geometries $\mathcal{B}$ and obstacle geometries $\mathcal{A}_i$ that make the object $\mathcal{B}$ completely immobile.

Before formally defining form closure, we need some stepping stones to help us understand the definition. First, we need a way of defining which configurations do, and which configurations do not cause a collision between $\mathcal{B}$ and $\mathcal{A}_i$.

Consider the configuration $\boldsymbol{q}$ in c-space and the homogeneous transformation $\mathcal{H}$ parameterized by $\boldsymbol{q}$. The configuration $\boldsymbol{q}$ is considered a *penetrating configuration* of the body $\mathcal{B}$ if the region $\mathcal{B} = \mathcal{H}(\mathcal{B}_0)$ intersects with $\mathcal{A}_i$. I.e that

$$\mathcal{B} \cap \mathcal{A}_i \neq \emptyset. \tag{4.16}$$

A penetrating configuration does not make sense from a physical perspective as it implies that two solids share the same space. We do, however, allow two objects to overlap only at their boundaries. In this case we call it a *contacting configuration*. The concepts of collision and penetration is neatly defined in $\mathcal{W}$, and the question immediately arises whether we can translate these concepts to c-space.

A *c-space obstacle* that we denote $\mathfrak{A}_i \subset \mathcal{Q}$ is an open subset of c-space, that represents the interaction between $\mathcal{A}_i$ and $\mathcal{B}$. This subset is constructed from all configurations $\boldsymbol{q}$ that would cause a penetration of the obstacle $\mathcal{A}_i$. One can think of each c-space obstacle as a"blob" in c-space that the configuration is not allowed to enter, as this would cause a penetration in the physical world.

Unlike in the physical world, c-space obstacles are allowed to overlap. A configuration $\boldsymbol{q}$ that is inside two c-space obstacles at once simply represents a real-life configuration where the body would penetrate two obstacles simultaneously. Similarly to the real world, a configuration $\boldsymbol{q}$ on the border of a c-space obstacle represents a contact. The *free space* $\mathcal{F} \subseteq \mathcal{W}$ is then all configurations that are not inside the c-space obstacles such that

$$\mathcal{F} = \mathcal{Q} \setminus \bigcup_i \mathfrak{A}_i. \tag{4.17}$$

Intuitively we can find the space of configurations inside the c-space obstacles by the complement of $\mathcal{F}$ that we denote $\mathcal{F}'$. Now that we know what regions in c-space we can and can't move in, we can also define what motions are possible in c-space. We define a path $\boldsymbol{q}(t) : \mathcal{T} \mapsto \mathcal{Q}$ as continuous path through c-space, where $\mathcal{T}$ is the timeframe in which we study the robot. Intuitively, this can be considered as a sequence of configurations $\boldsymbol{q}$ that forms a continuous motion of

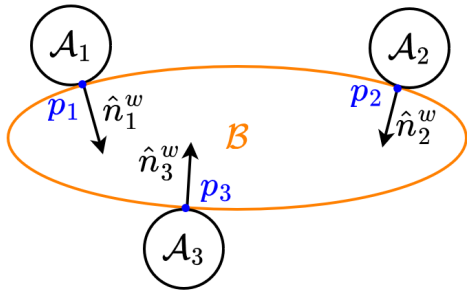


Figure 4.8: An object $\mathcal{B}$ in contact with three obstacles. The contact normal vectors are marked as black arrows and the contact points are marked as blue points.

the body $\mathcal{B}$ in the plane. Similarly, we define a *free path* as

$$\alpha(t) : \mathcal{T} \mapsto \mathcal{F} \tag{4.18}$$

that represents any path that is only in the free space, meaning that no configuration along the path will cause a penetration. With the definition of a free path solidly defined we arrive at the beautifully simple definition of form closure:

An body is *under form closure* in the configuration $\boldsymbol{q}_0$ if there exists no free path $\alpha(t)$ where $\alpha(0) = \boldsymbol{q}_0$ except for the trivial $\alpha(t) = \boldsymbol{q}_0 \quad \forall t \in \mathcal{T}$.

Lets break that down. We start at a configuration $\boldsymbol{q}_0$ in c-space. Our goal is to find a continuous path through c-space so that we never pass through the interior of the "blobs" defined by the c-space obstacles. If the only path we can find is the one where we remain at our initial configuration $\boldsymbol{q}_0$, then we are under form closure from the obstacles. One can visualize this as a point $\boldsymbol{q}_0$ that is totally enclosed by the c-space obstacles on all sides. If you are more familiar with the language of topology, one can also describe the form closed configurations as a property on $\mathcal{Q}$ as a topological space:

A rigid body is under under form closure in a pose $\boldsymbol{q}_0$ if, and only if, $\boldsymbol{q}_0$ is an isolated point of the free region $\mathcal{F}$ in the space $\mathcal{Q}$.

4.2.4 Computing Form Closure

Before we start this section, a little disclaimer is needed. The form closure problem is a complex one, as we are looking at finding continuous paths in space with obstacles of unknown shape and complexity. There exists no general solution to the form closure problem, however many methods have been developed to identify large and very useful subsets of configurations and obstacle geometries that do cause form closure. We will start by introducing the simplest form called 1^{st} order form closure and study its properties and shortcomings before we proceed to the other methods.

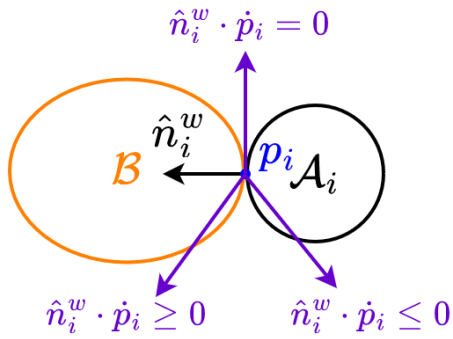


Figure 4.9: An object $\mathcal{B}$ in contact with an obstacle $\mathcal{A}_i$. The purple arrows show three different instantaneous motions of the point $\boldsymbol{p}_i$.

1st Order Form Closure

First order form closure builds on the idea of studying the twist space rather than the c-space directly. If we can model the interactions between the body and the obstacles in twist-space, and show that there exists no possible twist that would not cause a penetration, then surely the body must be under form closure.

1st order form closure is based on the knowledge of two pieces of information: The position of the contact points $\boldsymbol{p}_i \in \mathcal{W}$ and the contact normal vectors that we shall denote $\hat{\boldsymbol{n}}_i^{\mathcal{W}}$ where $\hat{\boldsymbol{n}}_i^{\mathcal{W}} \in \mathcal{W}$. The contact normal vectors are unit vectors such that for each contact point $\boldsymbol{p}_i$, the contact normal vectors originate in $\boldsymbol{p}_i$ and are normal to the bodies surface and point away from the body towards the obstacles. See the figure for a more intuitive explanation.

In a sense, each contact $(\hat{\boldsymbol{n}}_i^{\mathcal{W}}, \boldsymbol{p}_i)$ is defined by the contact point and the normal vector. We want to study the properties of these contact points when the body is under the influence of a twist ξ. Recall that a twist maps from a position to an instantaneous velocity. This means that it is possible to calculate the instantaneous velocity $\dot{\boldsymbol{p}}_i$ of each point $\boldsymbol{p}_i$ by applying (4.7) so that

$$\dot{\boldsymbol{p}}_i = \xi(\boldsymbol{p}_i). \tag{4.19}$$

If $\dot{\boldsymbol{p}}_i$ is oriented in the about the same general direction as $\hat{\boldsymbol{n}}_i^{\mathcal{W}}$ we see that the twist would indeed move the body away from obstacle $\mathcal{A}_i$. If $\dot{\boldsymbol{p}}_i$ was normal to $\hat{\boldsymbol{n}}_i^{\mathcal{W}}$ then the body would slide along the obstacle. If $\dot{\boldsymbol{p}}_i$ is oriented in the opposite general direction as $\hat{\boldsymbol{n}}_i^{\mathcal{W}}$ we see that the twist would move the body into the obstacle $\mathcal{A}_i$ causing a penetrating configuration. We can formalize this notion by saying that all twists that do not cause a penetration with obstacle $\mathcal{A}_i$ are constrained by

$$\xi(\boldsymbol{p}_i) \cdot \hat{\boldsymbol{n}}_i^{\mathcal{W}} \geq 0, \tag{4.20}$$

or more intuitively

$$\dot{\boldsymbol{p}}_i \cdot \hat{\boldsymbol{n}}_i^{\mathcal{W}} \geq 0. \tag{4.21}$$

This may look familiar. It is indeed the definition of a half-space in $\mathcal{W}$ similar to the definition in (4.12). The contact point creates a half-space of instantaneous velocities that would not cause a penetrating condition. Previously we saw how homogeneous transformations can be modeled in c-space. Can we do the same for this problem in twist-space? We start by applying (4.7) to (4.20) and expanding to

$$([\dot{\psi}]_\times \boldsymbol{p}_i + \dot{\boldsymbol{r}}) \cdot \hat{\boldsymbol{n}}_i^{\mathcal{W}} \geq 0. \tag{4.22}$$

With some algebraic manipulation, we find that

$$\dot{\boldsymbol{q}} \cdot \hat{\boldsymbol{n}}_i \geq 0 \quad | \quad \hat{\boldsymbol{n}}_i = \begin{bmatrix} [\boldsymbol{p}_i]_\times \hat{\boldsymbol{n}}_i^{\mathcal{W}} \\ \hat{\boldsymbol{n}}_i^{\mathcal{W}} \end{bmatrix}. \tag{4.23}$$

Again, the same pattern emerges. The equation (4.23) again forms a half-space, but this time it is not in the plane but rather in the twist space $\dot{\mathcal{Q}}$ parameterized by the point $\dot{\boldsymbol{q}}$. We see that if we pick a twist ξ that is parameterized by $\dot{\boldsymbol{q}}$ then if the relation (4.23) holds, then the twist will not cause a collision with obstacle $\mathcal{A}_i$.

The half-space may be easy to visualize for the twist space $\dot{\mathcal{Q}} = \mathbb{R}^3$ when we are working in the plane, but becomes difficult to wrap ones head around when raising the problem to 3D-space where the twist space is $\dot{\mathcal{Q}} = \mathbb{R}^6$. While we can get a somewhat intuitive understanding of half-spaces in $\mathbb{R}^3$ we will have to trust the math when working our way into higher dimensions. We can define the half-space $\mathcal{V}_i^1$ where $\mathcal{V}_i^1 \subset \dot{\mathcal{Q}}$ as

$$\mathcal{V}_i^1 = \{\dot{\boldsymbol{q}} \quad | \quad \hat{\boldsymbol{n}}_i \cdot \dot{\boldsymbol{q}} \geq 0\} \subset \mathbb{R}^k. \tag{4.24}$$

If we have more than one obstacle, then we can take the intersection of the resultant half-spaces to create an ever shrinking space of permissible twists. The space of permissible twists $\mathcal{V}^1 \subset \mathbb{R}^k$ is thus defined as

$$\mathcal{V}^1 = \bigcap_i \mathcal{V}_i^1 = \bigcap_i \text{half}(\hat{\boldsymbol{n}}_i) \tag{4.25}$$

which again might look familiar. It is the definition of a polyhedral cone from (4.14). We see that with the contacts $(\hat{\boldsymbol{n}}_i^{\mathcal{W}}, \boldsymbol{p}_i)$ the permissible twists can be modeled as a polyhedral cone in $\dot{\mathcal{Q}}$. This allows us to give a very succinct definition

of 1st order form closure: The body is in 1st form closure if and only if

$$\mathcal{V}^1 = \{0\}. \tag{4.26}$$

i.e. that the only permissible twist is the zero-twist in which the body remains stationary. From this definition we can define a key property of form closure: In general you need a minimum of 4 contact points to achieve 1st order form-closure in $\mathcal{W} = \mathbb{R}^2$ and a minimum of 7 to achieve the same with $\mathcal{W} = \mathbb{R}^3$. This results form the *Releaux-Somov proposition* [49] that minimum of $d + 1$ restraints are required for form closure in $\mathbb{R}^d$.

We see that the 1st order form closure problem reduces to the problem of computing whether or not a polyhedral cone contains only its origin; a problem that is numerically feasible to compute. In fact, the structure of the 1st order form closure problem is very similar to that of Linear Programming.

Numerical Computation of 1st. Order Form Closure

Lakshminarayana [49] introduced a numerical method for computing form closure for a given configuration and body-obstacle topology: A body is under form closure if its *constraint matrix* $\boldsymbol{GN}$ is of full row rank and there exists an $\boldsymbol{x} > 0$, $x \in \mathcal{Q}$ such that $\boldsymbol{GNx} = 0$ where

$$\boldsymbol{N} = \text{diag}(\hat{\boldsymbol{n}}_1 \dots \hat{\boldsymbol{n}}_i), \tag{4.27}$$

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{I} & \dots & \boldsymbol{I} \\ [\boldsymbol{p}_1]_\times & \dots & [\boldsymbol{p}_i]_\times \end{bmatrix}, \tag{4.28}$$

which is known to form a numerical optimization problem of the form

$$\max_{\boldsymbol{f} \in \mathcal{Q}} \quad \boldsymbol{f}^T \boldsymbol{x} \tag{4.29}$$

$$\text{s.t.} \quad \boldsymbol{N}^T \boldsymbol{G}^T \boldsymbol{x} \geq 0. \tag{4.30}$$

The existence of feasible solution is a sufficient condition for determining that a system is under form-closure. The problem amounts to standard Linear Programming (LP) problem, that is well-researched and for which many efficient solvers have been developed.

The Shortcomings of 1st Order Form Closure

It should be clear by now that 1st order form closure is an interesting property, as it is simple, tangible, and requires only a bare minimum of information about the body and obstacles (the contact points and normals). It is also possible to compute 1st order form closure in $\mathbb{R}^n$ using Linear Programming. It does however have some significant short-comings that we shall discuss further. We base our reasoning on Figure 4.10.

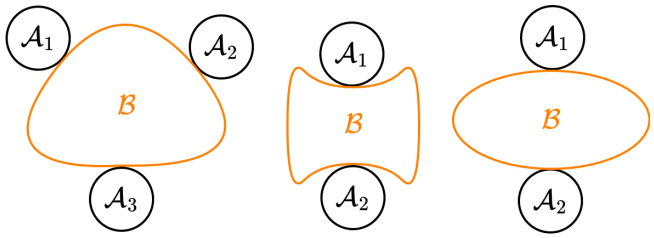


Figure 4.10: The left object is immobilized using only 3 obstacles. The middle object is immobilized using only 2 obstacles, while the right figure is not. However, the latter two are equivalent from a 1st order perspective.

First, we show a condition that immobilizes a body using 3 obstacles, contrary to the notion that 1^{st} order form closure would require $n \geq 4$ obstacles for immobilization. Interestingly, if one did the aforementioned calculations, one would find that this configuration does *not* have 1^{st} order form closure.

Secondly, we show two different bodies with the same obstacle topology, both of which are equivalent from a 1^{st} order form-closure perspective (i.e. that the contact points and normals are the same). Indeed, if one did the calculations, neither of the two configurations would have 1^{st} order form closure. However, one body is obviously immobilized by the obstacles, but the other is not. How can this be?

Both of these short-comings stem from the same problem: 1^{st} order form closure is a conservative metric. This means that there can exist configurations that are form closed, but that are not 1^{st} order form closed. Indeed, the systems with 1^{st} order form closure form only a subset of systems that can have form closure.

If we again look at our figures, the reason for this becomes clear. The curvature of the obstacles and the body around the contact points must be important properties of form closure, but this is not encoded into the 1^{st} order form closure problem. A different technique is needed to quantify systems with form closure that are not "caught" by our 1^{st} order analysis.

2nd. Order Form Closure

Before we proceed further, we will give a little warning on 2^{nd} order form closure:

While our 1^{st} order analysis is quite simple, the calculation of a 2^{nd} order analysis is significantly more complex. Unlike the 1^{st} order analysis, the 2^{nd} order analysis is far less studied and understood. The derivation and computation of 2^{nd} order form closure spans a full 16-page article on mechanical theory [51], and a comprehensive study would not fit neither the page count nor the scope of this tutorial. Because of this, we will limit our study on 2^{nd} order form closure to two aspects:

- An intuitive understanding of how it works, and how it amends some of the issues of 1^{st} order form closure.

- Its properties and application.

To calculate 2^{nd} order form closure we need 3 pieces of information. Similar to our 1^{st} order analysis we need the contact points $\boldsymbol{p}_i$ and their associated contact normals $\hat{\boldsymbol{n}}_i^{\mathcal{Q}}$. In addition, the analysis requires the *curvature form*, which is denoted κ_i. The curvature form is a property of a contact point, and encodes the curvature of both the surface body and the surface of the obstacle in the immediate vicinity of the contact point.

Similarly to how a 1^{st} order analysis makes a 1^{st} order approximation of the geometry of the contact point, a 2^{nd} order analysis approximates the problem to the 2^{nd} order. In the 1^{st} order analysis we required a calculation of the twist parameterized by $\dot{\boldsymbol{q}}$. In the 1^{st} order analysis we require also the derivative of the twist $\ddot{\boldsymbol{q}}$ that can be thought of as an interpretation of acceleration in the world of homogeneous transformations and twist.

The resulting problem is similar to the one we solved for 1^{st} order form closure where a polyhedral cone had to be reduced to only its origin. In our 2^{nd} order analysis, we attempt to do the same, only that we now allow our free paths to be curved and the spaces representing our obstacle interactions in c-space to be curved. In Figure 4.10, the center and left figures both have 2^{nd} order form closure while the right figure does not.

2^{nd} order form closure has the useful property that a minimum of only 2 contact points are required. This applies to both bodies in $\mathcal{W} = \mathbb{R}^2$ and $\mathcal{W} = \mathbb{R}^3$ which is a significant reduction from 1^{st} order analysis where respectively 4 and 7 contact points were required. This allows us to quantify a whole range of systems that may have form closure, that are not necessarily form closed to the 1^{st} order.

The Shortcomings of 2nd. Order Form Closure

Apart from being difficult to compute, and the requirement additional information on the body and obstacle geometry, 2^{nd} order form closure has another significant short-coming which we shall address here.

A known and important property of 1^{st} order form closure is that for any finite force, torque or combination thereof (i.e. a wrench) applied to the body will result in finite reaction forces from the obstacles to keep the body stationary. This is not necessarily true for 2^{nd} order form closure where a finite wrench might produce infinite reaction forces. One such configuration is shown in the middle part of 4.10 where any attempted finite horizontal displacement would have to create an infinite reaction force from the obstacles to keep the body stationary.

This has implications for real-life implementations as the robot utilizing form closure as part of a control system would have to account for the possibility of infinite or near-infinite forces in the system, that could cause trouble for down-the-line control strategies. This concept is also explored in the field of statics

and architectural engineering where such systems are called *infinitesimally mobile*, giving the notion that the system can only be subject to infinitesimally small perturbations without creating infinite reaction forces.

Higher order form closure

In a similar fashion to 1st order analysis, there exists systems that are form closed that do not have 2nd order form closure. In fact, this notion is possible to extend into absurdity as we can find larger and larger subsets of form closed systems by increasing the order of our analysis.

To the author's knowledge, no formal derivation of 3rd or higher order form closure exists. Neither does there exist a computational method to calculate it. The reasoning for this might be the few added benefits of a higher order analysis. Intuitively, it is not possible to immobilize a body with fewer than 2 contact points, and as such a higher order analysis mostly introduces computational complexity with no real improvement for real-life applications.

The cases where a 2nd order analysis is insufficient (i.e. where a 3rd or high order analysis are needed) are few and especially contrived. Because of this we deem them of little importance for further study in this tutorial.

An Analytic Test for 1st. Order Form Closure in the Plane

In this subsection we derive an analytical test for form closure in the plane. Recall that calculation of 1st order form closure requires determining whether a polyhedral cone $\mathcal{V}^1$ contains only its origin. The polyhedral cone has only one vertex, namely at the origin, and the edges of polyhedral cone are rays emanating from the origin and stretching towards the horizon of the ambient space.

Thus we can make a quick and intuitive claim: A polyhedral cone with no edges can only contain the origin. I.e. that $\mathcal{V}^1 = \{0\}$. This implies that we can assert whether a body is under form closure by checking if a polyhedral set has no edges (which in our case are rays). If the polyhedral cone is defined as the intersection of multiple half-spaces $\mathcal{N}_i$, the edges can only exist in the intersection between the boundary of two half-spaces $\delta\mathcal{N}_i \cap \delta\mathcal{N}_j$ where the two planes are non-parallel. We call this the *edge candidate set* $\mathcal{E} \subset \mathcal{Q}$ that we define as

$$\mathcal{E} = \bigcup_{i,j} (\delta\mathcal{N}_i \cap \delta\mathcal{N}_j) \quad | \quad \delta\mathcal{N}_i \nparallel \delta\mathcal{N}_j \tag{4.31}$$

that forms a set of rays emanating from the origin. Each intersection contributes two rays, each going in opposite directions. Checking whether a point is in $\mathcal{V}^1$ is the fairly simple task of checking the inequality in (4.12). If we check every point in $\mathcal{E}$ and none of them are inside $\mathcal{V}^1$, then there exists no possible set where the edges of the polyhedral cone can exist, that again implies 1st order form closure. Formally the relation can be written as

$$\mathcal{E} \cap \mathcal{V}^1 = \{0\} \implies \mathcal{V}^1 = \{0\}. \tag{4.32}$$

If a single point on a ray is inside $\mathcal{V}^1$, then the entire ray must be inside $\mathcal{V}^1$. We can reduce the complexity of the problem even further by specifying one point on each ray and checking if that single point is inside $\mathcal{V}^1$ rather than checking all points on the ray. To this end, we select every point on every ray in $\mathcal{E}$ that is a unit distance from the origin, that can be computed as $\mathcal{S}^2 \cap \mathcal{E}$ where $\mathcal{S}^2$ is the unit sphere. We find that

$$\mathcal{S}^2 \cap \mathcal{E} \cap \mathcal{V}^1 = \emptyset \implies \mathcal{E} \cap \mathcal{V}^1 = \{0\} \implies \mathcal{V}^1 = \{0\}. \tag{4.33}$$

The set $\mathcal{S}^2 \cap \mathcal{E}$ is always finite and can be computed using simple linear arithmetic, so the act of checking $\mathcal{S}^2 \cap \mathcal{E} \cap \mathcal{V}^1$ can be done using a finite number of computations.

While this computation may seem theoretical in nature, it has a very logical and succinct representation in the real world: To understand this representation we first have to examine the properties of a twist. Thus far, we have parameterized a twist by a linear velocity $\dot{\boldsymbol{r}}$ and an angular velocity $\dot{\boldsymbol{\psi}}$, fused together to form the c-space coordinate $\dot{\boldsymbol{q}}$. A special case of Chasles' theorem states that any twist in the plane can be parameterized by a special point called a *pole* and an angular velocity about the pole. We call this a *pole parametrization* and denote it by the pair $(\dot{\psi}, \boldsymbol{s})$ where $\boldsymbol{s} \in \mathcal{W}$ is the pole. By looking at Figure 4.5 one can clearly see the pole as the point where the field vanishes, i.e. where $\xi = 0$. The angular velocity $\dot{\psi} \in \mathbb{R}$ defines the handedness and the magnitude of the field surrounding $\boldsymbol{s}$. For the pole parametrization to represent all possible twists, we need to include the scenarios where $\boldsymbol{s}$ is located infinitely far away from the origin in some direction, as this is needed to represent pure linear velocities.

We can model the contact between the snake and an obstacle using the pole parametrization as

$$\dot{\psi}([\hat{\boldsymbol{n}}_i^{\mathcal{W}}]_\times(\boldsymbol{p}_i - \boldsymbol{s})) \geq 0 \tag{4.34}$$

where $\boldsymbol{p}_i - \boldsymbol{s}$ is the relative position of the obstacle to the pole. If we find a single

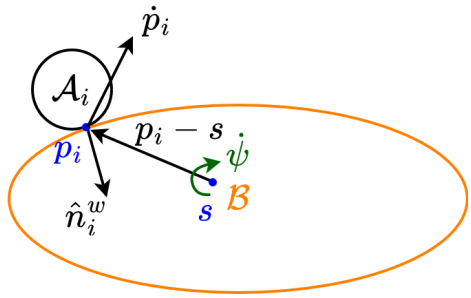


Figure 4.11: A visualization of all the elements that constitute a contact under a pole parameterized twist.

positive $\dot{\psi}$ that satisfies (4.34) then the inequality holds for all positive values of $\dot{\psi}$.

The same applies in the converse case that a single negative solution is sufficient for the inequality to hold for all negative values. This intuitively makes sense as form closure really doesn't care about the "magnitude" of a twist, but rather its "direction". We see that this encompasses the notion of looking only at twists in $\mathcal{S}^2 \subset \mathcal{Q}$. We define the *unit twists* as all twists where $|\dot{\psi}| = 1$.

Recall that 1st order form closure requires that there exists no twist that won't cause a penetration with at least one obstacle. Using this knowledge we can narrow the definition of form closure to the following: A body is under form closure if there exists no *unit* twist that won't cause a penetration with at least one obstacle. This is a subtle difference, but it significantly reduces the space of twists we have to "try" in order to verify form closure. It suffices to check every possible pole $\boldsymbol{s}$ with $\dot{\psi} = \pm 1$. We now define two geometric properties of the contact points:

- The *contact orthogonal line* $\mathcal{L}_i \subset \mathcal{W}$ of a contact point i is a line passing through the contact point $\boldsymbol{p}_i$ and that is spanned by the contact normal vector $\hat{\boldsymbol{n}}_i^{\mathcal{W}}$.
- Two lines $\mathcal{L}_i$, $\mathcal{L}_j$ are *concurrent* if they coincide in a single concurrent point $\boldsymbol{\sigma}_{i,j} \in \mathcal{W}$.

Every contact point has a contact orthogonal line $\mathcal{L}_i$, and as these lines intersect they form concurrent points. We create a set of all concurrent points of the lines $\mathcal{L}_i$

$$\mathcal{S}_\sigma = \left\{ \boldsymbol{\sigma}_{i,j} \quad | \quad \mathcal{L}_i \nparallel \mathcal{L}'_j \right\}. \tag{4.35}$$

In the case that the lines are parallel, they don't intersect and consequently don't form a concurrent point. The number of concurrent points is upper bounded by

$$\text{card}(\mathcal{S}_\sigma) \leq \frac{{n_c}^2 - n_c}{2} \, . \tag{4.36}$$

where n_c is the number of contact points. With these intermediate results in mind we can analyze the form closure problem in the plane. The concurrent points have a very interesting property when we use them as poles in twists. It is trivial to show that for a twist with a pole placed at one of the concurrent points $\boldsymbol{s} \in \mathcal{S}_\sigma$, then $\dot{\boldsymbol{p}}_i \cdot \hat{\boldsymbol{n}}_i^{\mathcal{Q}} = 0$ for at least 2 of the contact points. Such a twist is shown in Figure 4.12. Physically we can think of this as a twist that causes the body to roll or slide along two of the contact points. This again ties nicely into our theoretical foundations, as all twists that share the same pole lies along one of the lines $\delta\mathcal{N}_i \cap \delta\mathcal{N}_j$. One consequence of such a twist is that two of the contact points contribute nothing in immobilizing the body under the given twist. Thus we need at least two more contact points, one preventing a clockwise twist and one preventing a counterclockwise twist.

We see that we have proved the previous statement that we need at least 4 contact points for 1st order form closure. Interestingly, if we move the pole

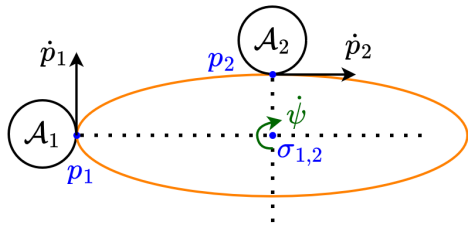


Figure 4.12: A body with a twist applied around a pole in the concurrent point of the two contact normals.

away from the concurrent point, we will see that the same number or more of the contact points contribute to keeping the body immobilized. The concurrent points form the poles that parameterize the twists that are most likely to cause a non-penetrating motion. If we can prove that any twist with a pole at a concurrent point causes a penetration, then this applies to all other twists as well, as they are impeded by the same amount or more contact points. This again implies form closure of the body.

This implies that form closure can be verified using very few computations. The most succinct way would be to define a set

$$\mathcal{S}_c = \left\{ \dot{\boldsymbol{p}}_1 \cdot \hat{\boldsymbol{n}}_1^{\mathcal{W}} \dots \dot{\boldsymbol{p}}_i \cdot \hat{\boldsymbol{n}}_i^{\mathcal{W}} \right\}, \tag{4.37}$$

that would naturally contain two elements that are zero when the pole is placed at a concurrent point. If, for every pole at a concurrent point $\boldsymbol{s} \in \mathcal{S}_\sigma$, the set $\mathcal{S}_c$ contains at least two elements of opposing signs, then the body is under form closure. The two elements of opposing signs intuitively represent the two extra contact points needed to prevent a clockwise twist or a counterclockwise twist. Intuitively a configuration with no concurrent points cannot have form closure as all contact points are parallel, in which case we can always find a translation that does not penetrate any obstacle.

We denote the number of contact points n_c. A simple and efficient test for form closure then can be structured as follows:

1. If $n_c < 4$ then we have no 1^{st} order form closure;
2. Calculate $\mathcal{S}_\sigma$. If $\mathcal{S}_\sigma = \emptyset$ then we have no 1^{st} order form closure;
3. If and only if, for every pole at a concurrent point $\boldsymbol{s} \in \mathcal{S}_\sigma$, the set $\mathcal{S}_c$ contains at least two elements of opposing signs, then we do have 1^{st} order form closure

4.3 Article IV: Snakes On a Plane: Form Closure and Constrainedness in Planar Snake Robots

Abstract

Several tasks related to snake robots require coordinated exploitation of the robot's multiple points of contact with its environment; the robot "grasps" the environment in order to constrain its own movements in a predictable manner. In this paper the theory of form closure is applied to planar snake robots to identify necessary and sufficient conditions for this form of constrainedness. The potential areas of application for this concept include path planning, locomotion and anchoring. We show how a minimum of 2 contact points are required for form closure in smooth snakes, and that more than 2 contact points are required for form closure in articulated snake robots. Form closure in articulated snake robots is possible with 3 contact points under some restraining conditions. A computationally efficient way of determining 1st order form closure is presented.

This paper is not yet published and is therefore not included.

4.4 Article V: Form Closure For Fully Actuated and Robust Obstacle-Aided Locomotion in Snake Robots

Abstract

In this paper we adapt the theory of form closure to identify the form closed region: a subset of a snake robots' configuration space where the robot can be modeled as a fully actuated system by leveraging constraints imposed by obstacles in its environment. We show that the identification of form closed configurations is numerically feasible, and introduce the relaxed condition of form boundedness to achieve robustness in the presence of model uncertainties. Finally, an example application of the form closed region is shown, where a snake robot produces predictable constrained motion in a cluttered environment using lateral undulation in the form closed region.

4.4.1 Introduction

Snake robots aspire to inherit the unique abilities of their biological counterparts. Be it to maneuver in rugged and complex terrain that is inaccessible for legged or wheeled robots, move in narrow and enclosed spaces and possibly climb complex structures [3]. Currently, this is a largely unrealized potential. Like biological snakes, these robots move using an array of different propulsion techniques [2]. This paper focuses specifically on one genre of snake robot locomotion called *Obstacle-Aided Locomotion* (OAL) [5]. The overarching goal of OAL is for the snake robot to produce propulsion by pushing its body against obstacles in a cluttered environment such as the one shown in Figure 4.20.

The endeavor of controlling a snake robot is complex. These systems are underactuated in its very nature, as its joints may be actuated directly but its position and orientation in its ambient space constitute unactuated degrees of freedom. The snake robots' interaction with obstacles in its environment further complicates the robots' kinematics by introducing non-holonomic and discontinuous contact constraints.

In this paper we attempt to identify a computationally feasible region of the robots' configuration space in which the kinematics of the robot can be simplified, in order to make OAL control strategies more tractable. We introduce the concept of form closure to identify the *form closed region*: a subset of a robots' configuration space where it can be modeled as a fully actuated kinematic system, by leveraging the constraints imposed by obstacles in its environment. A robot that is limited in motion to the form closed region exhibits several properties that are beneficial in motion planning and OAL, which are covered in this paper.

Form closure has found diligent use in the field of robotic grasping, and is widely used as a method for calculating suitable grasps for manipulating objects with different geometries. A previous work adapts the theory of form closure to planar snake robots (c.f Section 4.3), and shows how calculations on form closure

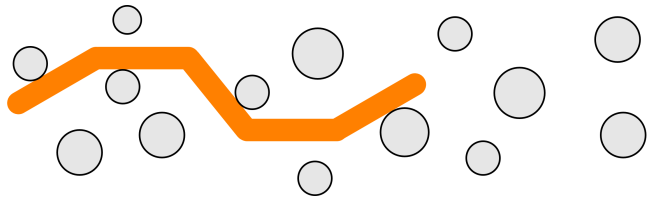


Figure 4.20: An articulated snake robot (orange) seen from above, in a planar environment cluttered by obstacles. How may the snake robot actuate to produce meaningful locomotion leveraging the obstacles in its environment?

are numerically, and in some cases even analytically, feasible. The previous work should be considered a direct prerequisite to the current paper.

The structure of this paper is as follows: Section 4.4.2 gives a brief review of the state-of-the art in snake robot modeling and locomotion. The background theory on snake robot geometry and kinematics, and a formal treatment of form closure can be found in Section 4.4.3. In Section 4.4.4, we show an application of form closure in a planar case. Here it is shown that form closure can be used to guarantee constrained locomotion in cluttered environment using lateral undulation. Section 4.4.5 discusses the findings of this paper, identifies potential weaknesses and outlines future work. The main contributions of this paper are covered in Section 4.4.3 through Section 4.4.5.

4.4.2 Recent Works

The history and application of different locomotion strategies have been thoroughly documented in a recent review on snake robots [3]. Because of this, we limit this section to works that have been published since the review, and publications directly pertaining to OAL or similar concepts.

The term OAL was first introduced in [5] and further explored in the subsequent works [11, 26, 32]. Hybrid modeling of snake robots was first visited in [26], but has since been elaborated upon by [25] which adapts the theory of Hybrid Position-Force Control. In [58], a piece-wise helical motion is used to produce OAL in a cluttered planar environment. More recently, the findings in [63] show a strategy for locomotion applying theory from geometric mechanics in a scenario similar to the one studied in this paper. The work in [64] relates closely to the fundamental ideas of OAL, where traveling waves are used to propel a snake robot through virtual "hoops". A method for perception-driven path planning for OAL was proposed in [65].

Recent works pertaining to snake locomotion, but only indirectly to OAL, include a study that shows how concertina-like locomotion can be used for mobile manipulation [66], a study on path following using anisotropic friction on planar surfaces [67] and a study on helical rolling in straight pipes [68]. A similar approach for locomotion in pipes using trapezoidal-like waves can be found in [69]. Adaptive control for under-actuated snake robots has been explored in [70].

Form closure was used to calculate grasps when using the body of a snake robot as a gripping mechanism in [71]. However, to the best of our knowledge, the present study is the first to demonstrate the use of form closure and related concepts for locomotion purposes.

4.4.3 Theory

The following section contains a kinematic and geometric model of a planar snake robot and a formal introduction to form closure

A geometric model of a planar snake robot

Consider a planar snake robot that inhabits a planar workspace $\mathcal{W} = \mathbb{R}^2$ that we study in a compact timeframe $\mathcal{T} \subset \mathbb{R}$ with t representing a point of time in $\mathcal{T}$. The robot is comprised of N links connected by $N-1$ joints, indexed from tail to head, as shown in Figure 4.21. We model these as an open kinematic chain of N links where the joints are placed where two consecutive links meet. The joint angles are given by

$$\boldsymbol{\phi} = (\phi_1 \dots \phi_{N-1}) \in \mathbb{R}^{N-1}$$

where ϕ_i is the relative angle between the two links interconnected by joint i. The distance between two consecutive joints is $2l$. As the robot is not tethered to its environment, its *pose* in relation to its environment is given by a vector $\boldsymbol{q}_N \in SE(2)$. While the pose of the robot may be referenced to any part of its body, we chose to reference the pose to its head such that

$$\boldsymbol{q}_N = (\boldsymbol{x}_N, \boldsymbol{y}_N, \boldsymbol{\theta}_N), \tag{4.59}$$

where $(\boldsymbol{x}_N, \boldsymbol{y}_N) \in \mathcal{W}$ is the position of the robots' head in the workspace and θ is the orientation of the its head relative to the world frame. The complete configuration of the robot is given by the generalized coordinate $\boldsymbol{q}$ which is defined as

$$\boldsymbol{q} = (\boldsymbol{\phi}, \boldsymbol{q}_N) \in \mathcal{Q}, \tag{4.60}$$

where $\mathcal{Q} = \mathbb{R}^{N-1} \times SE(2)$ is the configuration space of the robot. The pose $\boldsymbol{q}_N$ inhabits a subspace of the configuration space $\mathcal{Q}_N \subset \mathcal{Q}$ such that $\mathcal{Q}_N = SE(2)$. The configuration space is divided into two orthogonal subspaces: $\mathcal{Q}_N$ which contains all of the unactuated variables, and $\mathcal{Q} \backslash \mathcal{Q}_N$ that contains the actuated variables.

The body of the robot occupies an open, bounded and simply connected region $\mathcal{S}(\boldsymbol{q}) : \mathcal{Q} \mapsto \mathcal{W}$. The choice of $\mathcal{S}$ is arbitrary, but for the remainder of this paper we define $\mathcal{S}$ as the set of all points whose distance to the spine is less than w. The distance $2w$ is consequently also the width of the robots' body.

As the coordinate $\boldsymbol{q}$ provides a full parametrization of the robot, any connected trajectory $\boldsymbol{q}(t) : \mathcal{T} \mapsto \boldsymbol{Q}$ corresponds to a connected physical motion of the snake

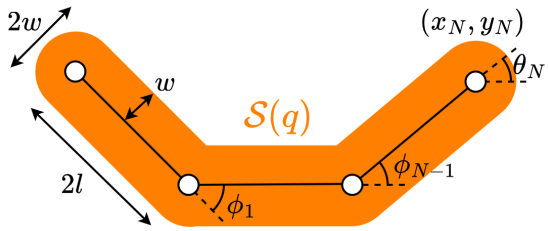


Figure 4.21: A kinematic and geometric model of an articulated planar snake robot with $N = 3$ links.

robot in the workspace $\mathcal{W}$. If we add the additional constraint that $\boldsymbol{q}(t)$ is twice differentiable with respect to time, then the trajectory represents a physically realizable motion with finite generalized forces. The workspace $\mathcal{W}$ is cluttered by a series of obstacles which occupy a closed, bounded and possibly disconnected set $\mathcal{O} \subset \mathcal{W}$. We now define

Definition 6 *A configuration $\boldsymbol{q} \in \mathcal{Q}$ is penetrating if the body of the robot and the obstacles overlap such that*

$$\mathcal{O} \cap \mathcal{S}(\boldsymbol{q}) \neq \emptyset.$$

Intuitively, any penetrating configuration is not physically realizable, and constrains the motion of the robot to a physically realizable region. We define this as:

Definition 7 *The free region $\mathcal{F} \subset \mathcal{Q}$ is the region of all non-penetrating configurations such that*

$$\mathcal{F} = \{\boldsymbol{q} \in \mathcal{Q} \quad | \quad \mathcal{O} \cap \mathcal{S}(\boldsymbol{q}) = \emptyset\}.$$

Definition 8 *A configuration in the free region $\boldsymbol{q} \in \mathcal{F}$ is contacting if it is on the boundary of the free region such that $\boldsymbol{q} \in \delta\mathcal{F}$,*

where δ is an operator denoting the boundary of a region. The complement to the free region, $\mathcal{F}' = \mathcal{Q} \backslash \mathcal{F}$, consequently denotes the region of physically infeasible configurations.

We denote the projection of the free region $\mathcal{F}$ into the subspace $\mathcal{Q}_{\mathcal{N}}$ as $\mathcal{F}_N(\boldsymbol{\phi})$. The region $\mathcal{F}_N(\boldsymbol{\phi})$ represents any free motions of the head pose given a set of joint angles $\boldsymbol{\phi}$.

Form closure for snake robots

A rigid motion of the robot is defined as one during which all joint angles are kept constant, i.e. $\dot{\boldsymbol{\phi}} = \mathbf{0}$. The overarching goal of form closure analysis in snake robots is to determine if all continuous rigid motions of the robot from a starting

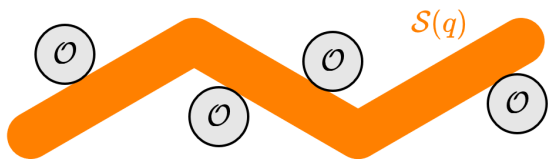


Figure 4.22: A simple articulated snake robot $\mathcal{S}(q)$ with $N = 3$ links in a workspace $\mathcal{W} = \mathbb{R}^2$ which is under form closure from the obstacles in its environment.

configuration would cause its body to penetrate the obstacles in its environment. From a physical perspective, this implies that snake robot under form closure would be completely immobilized by the obstacles if it were to keep its joints rigid. A simple example of a form closed robot is shown in Figure 4.22.

All possible rigid motions of the robot are encoded in the model by the pose $\boldsymbol{q}_N$; if all neighboring points of $\boldsymbol{q}_N$ lie in the infeasible region $\mathcal{F}'_N(\boldsymbol{\phi})$ on all sides, then there exists no rigid motion that would not cause a penetrating configuration. We can formalize this as:

Definition 9 *A snake robot is under form closure in the configuration* $\boldsymbol{q} = (\boldsymbol{\phi}, \boldsymbol{q}_N)$ *if, and only if,* $\boldsymbol{q}_N$ *is an isolated point of* $\mathcal{F}_N(\boldsymbol{\phi})$.

The region of all configurations that are under form closure can then be defined as:

Definition 10 *The form closed region* $\mathcal{F}^{FC} \subset \delta\mathcal{F}$ *is the region of all configurations* $\boldsymbol{q}$ *where the snake robot is under form closure.*

The form closed region is a subset of the boundary of the free region as one intuitively needs contact with the environment to be able to achieve form closure. The above definitions allow us to investigate the properties of the form closed region. We propose that:

Definition 11 *A continuous trajectory* $\boldsymbol{q}(t)$ *is form closed if it is embedded in the form closed region such that* $\boldsymbol{q}(t) : \mathcal{T} \mapsto \mathcal{F}^{FC}$.

Theorem 5 *Consider a form closed trajectory* $\boldsymbol{q}^d(t)$ *where* $\boldsymbol{q}^d(t) = (\boldsymbol{\phi}^d(t), \boldsymbol{q}_N^d(t))$. *If a snake robot* $\boldsymbol{q}(t) = (\boldsymbol{\phi}(t), \boldsymbol{q}_N(t))$ *is placed such that* $\boldsymbol{q}(0) = \boldsymbol{q}^d(0)$ *and actuated by* $\boldsymbol{\phi}(t) = \boldsymbol{\phi}^d(t)$, *then it follows that* $\boldsymbol{q}_N(t) = \boldsymbol{q}_N^d(t) \, \forall \, t \in \mathcal{T}$.

From a geometric perspective, the above states that if a snake robot is placed in an initial form closed configuration and actuated along a desired trajectory in the form closed region, its pose is completely determined by its joint angles for the entirety of the trajectory. The proof of the above theorem follows from Definition 9: any departure of the head pose $\boldsymbol{q}_N(t)$ from the desired head pose $\boldsymbol{q}_N^d(t)$ would have to enter the infeasible region $\mathcal{F}'_N(\phi)$ as $\boldsymbol{q}_N^d(t)$ is an isolated point in $\mathcal{F}_N(\phi)$. It follows from Theorem 5 that:

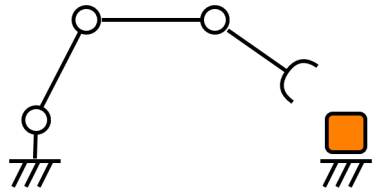


Figure 4.23: A tethered robot manipulator in $\mathcal{W}$ where $\mathcal{W} = \mathbb{R}^2$ accompanied by an untethered block (orange). The system has an identical configuration space representation to a snake robot.

Corollary 3 *For any form closed trajectory where* $\boldsymbol{q}(t) = (\boldsymbol{\phi}(t), \boldsymbol{q}_N(t))$*, the head pose of the snake robot* $\boldsymbol{q}_N(t)$ *is completely determined by its joint angles* $\boldsymbol{\phi}(t)$*.*

Corollary 4 *When limited in motion to the form closed region* $\boldsymbol{q} \in \mathcal{F}^{FC}$*, the configuration space of the snake robot is constrained from* $\mathbb{R}^{N-1} \times SE(2)$ *to* $\mathbb{R}^{N-1}$*.*

Corollary 5 *When limited in motion to the form closed region* $\boldsymbol{q} \in \mathcal{F}^{FC}$*, the snake robot is fully actuated.*

The above statements might best be understood by studying a similar system parameterized by the same configuration space. Consider a tethered planar robot manipulator of $N-1$ links with a gripper on the end accompanied by an untethered block, as shown in Figure 4.23. Assume that the gripper is capable of producing a form closed grasp on the block, and that we disregard the kinematics of the gripper itself. The configuration space of the system is given by the $N-1$ joints of the manipulator and the free motion of the block. Thus the system has a configuration space $\mathbb{R}^{N-1} \times SE(2)$, identical to that of the snake robot.

If the manipulator was to grip the block with a form closed grasp, the system is constrained, and the motion of the block is completely determined by the motion of the manipulator. In this case, the configuration space of the system is constrained to $\mathbb{R}^{N-1}$. This notion is transferable to a snake robot where the unactuated motion is that of the untethered snake robot and form closure is achieved by unilateral contact with fixed obstacles instead of using a gripper.

An important feature of the form closed region is the possibility for locomotion by slithering. Form closure is strictly a property of the robots' contact points with the obstacles and the geometry of the bodies in the vicinity of these points. The snake robot can slide along the obstacles while still remaining in form closure, although this might seem counter-intuitive given that the goal of form closure is to immobilize the snake robot. This feature is examined further in Section 4.4.4.

While the kinematic constraints on the snake robot in general are non-holonomic due to discontinuous contact with obstacles, we can make some assumptions on the constraints when limiting our motion to the form closed region.

In the general case, the dynamics of the pose $\boldsymbol{q}_N$ is highly non-linear and discontinuous. Its dynamics are unactuated, and is determined by the internal

dynamics of the robot, in conjunction with the external forces acting on the robot from the obstacles [11]. Consequentially, it is difficult to determine whether a given actuation may cause the robot to lose any of its current contact points with its environment. We propose that

Theorem 6 *When under form closure, no motion in $\mathcal{Q}_N$ can cause a snake robot to depart from a contact point.*

The above theorem states that it is impossible to "accidentally" lose contact with an obstacle due to the unactuated dynamics of the robot. Whether or not the snake robot departs from an obstacle is given entirely by its actuated dynamics. The proof of the theorem follows directly from Definition 9, as any motion in $\mathcal{Q}_N$ when under form closure would cause the robot to enter the physically infeasible region $\mathcal{F}'_N(\boldsymbol{\phi})$.

On the Computation and Properties of Form Closure

The study of form closure is well developed from the field of robotic grasping, where form closure is used to calculate suitable grasps for objects with different geometries. The form closure problem is complex, as the obstacles may have complex geometries. Because of this, there is no general method for computing form closure. However, methods exist that allows for form closure computation for a large subset of scenarios using approximations of the involved geometries. These methods can be applied efficiently using numerical calculations and are reviewed in detail in Section 4.3 and [47], and thus only a brief introduction to its content is given in this section.

The most widely adapted approximations is that of 1$^{\text{st}}$ order form closure [50], in which the contact geometry between the robot and the obstacles is approximated to the 1$^{\text{st}}$ order. This method has the advantage that it guarantees the finiteness of the reaction forces between the robot and the obstacles when subject to a finite external force, but it requires a minimum of four unique contact points in the planar case.

Second, and higher order form closure [51], is possible for articulated planar snake robots with only three contact points (or even two in some limited scenarios), but do not give any guarantees as to the finiteness of the reaction forces, and are significantly more difficult to calculate. Because of the guarantee of finite reaction forces and its computational simplicity, we consider 1$^{\text{st}}$ order form closure as the most relevant in the context of snake locomotion.

Robustness and Form Boundedness

Form closure is a strict condition requiring the head pose $\boldsymbol{q}_N$ to be completely enclosed by $\mathcal{F}'_N(\boldsymbol{\phi})$, which in theory would require perfect knowledge of the geometry of the snake robots' environment and of the robot itself. In a real-life scenario, the joint angles might deviate slightly from their desired angles, the body of the snake might deform under load, the estimated position of the obstacles might be inaccurate or the obstacles might shift under interaction. As such, form closure in

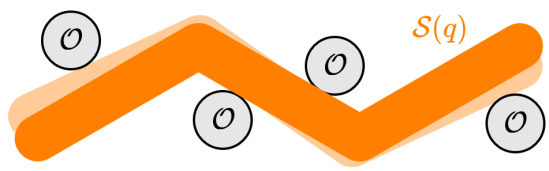


Figure 4.24: An articulated snake robot $\mathcal{S}(q)$ with $N = 3$ links in a workspace $\mathcal{W} = \mathbb{R}^2$ which is form bounded by the obstacles in its environment. The lightly shaded geometry shows an alternative pose of the robot caused by a rigid motion.

its strictest mathematical sense is nearly impossible to achieve, making a control strategy based on strict form closure fragile. To address this problem we introduce the relaxed condition of *form boundedness* and define:

Definition 12 *A snake robot is form bounded in the configuration* $\boldsymbol{q} = (\boldsymbol{\phi}, \boldsymbol{q}_N)$ *if the connected component of* $\boldsymbol{q}_N$ *in* $\mathcal{F}_N(\boldsymbol{\phi})$ *is bounded.*

One such configuration is shown in Figure 4.24. A robot which is form bounded, as opposed to form closed, is allowed some "wiggle room" around its current pose, but cannot leave a neighborhood of its current configuration without causing a penetration. Form closure is indeed a special case of form boundedness where the connected component contains only $\boldsymbol{q}_N$. This allows us to define a region of form bounded configurations and trajectories:

Definition 13 *The form bounded region* $\mathcal{F}^{FB} \subset \mathcal{F}$ *is the set of all configurations* $\boldsymbol{q}$ *that are form bounded.*

Definition 14 *A trajectory* $\boldsymbol{q}(t)$ *is form bounded if it is embedded in the form bounded region such that* $\boldsymbol{q}(t) : \mathcal{T} \mapsto \mathcal{F}^{FB}$

The above definition allows us to restate Theorem 5 in terms of form boundedness as:

Theorem 7 *Consider a form bounded trajectory* $\boldsymbol{q}^d(t)$*, where* $\boldsymbol{q}^d(t) = (\boldsymbol{\phi}^d(t), \boldsymbol{q}_N^d(t))$*. If a snake robot* $\boldsymbol{q}(t) = (\boldsymbol{\phi}(t), \boldsymbol{q}_N(t))$ *is placed such that* $\boldsymbol{q}(0) = \boldsymbol{q}^d(0)$ *and actuated by* $\boldsymbol{\phi}(t) = \boldsymbol{\phi}^d(t)$*, then it follows that* $\boldsymbol{q}_N(t)$ *remains in the connected component of* $\boldsymbol{q}_N^d(t) \, \forall \, t \in \mathcal{T}$

This implies that under form bounded locomotion, the robot will remain within some bounded neighborhood of its desired trajectory, although it may not track the undulation path perfectly. The proof is conceptually similar to that of Theorem 5, as any $\boldsymbol{q}_N(t)$ leaving the connected component of $\boldsymbol{q}_N^d(t)$ would require $\boldsymbol{q}$ to enter $\mathcal{F}'_N(\boldsymbol{\phi})$, as the connected component is bounded by $\mathcal{F}'_N(\boldsymbol{\phi})$. A visual representation of the different forms of boundedness is shown in Figure 4.25. Unlike form closure, form boundedness does not constrain the configuration space of the system and does not necessarily guarantee that the snake robot is fully actuated.

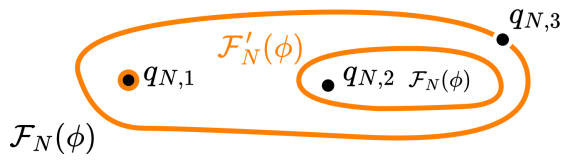


Figure 4.25: A visual representation of three different head poses in c-space. $\boldsymbol{q}_{N,1}$ is under form closure as it is an isolated point of $\mathcal{F}_N(\boldsymbol{\phi})$. The configuration $\boldsymbol{q}_{N,2}$ is form bounded as its connected component is bounded. $\boldsymbol{q}_{N,3}$ has neither form closure nor form boundedness.

4.4.4 Demonstration

The following demonstration showcases the locomotion of a simulated snake robot in a cluttered 2D-environment. The demonstration aims to show the predictable behavior of the robot when locomoting in the form closed region, and how its dynamics are affected when leaving the region.

We are specifically studying a form of OAL known as *lateral undulation* [2, 72], which is shown in Fig 4.26. During lateral undulation, the robot slithers along obstacles in a manner such that every point on its body trails the position of its head, with minimal lateral slippage. It achieves this by propagating the geometry of its body from its head towards its tail. We refer to the path traced by the snake robots body during locomotion as the *undulation path*.

Lateral undulation is the preferred mode of propulsion for biological snakes [1]. As they undulate through cluttered terrain, the snakes appear to chose their contact points with their environment in a way that is beneficial for their locomotion. Intuitively, there exists configurations where a lateral undulation gait would push the snake off its undulation path or cause the snake to lose propulsion entirely. Biological snakes actively choose paths through their environment that

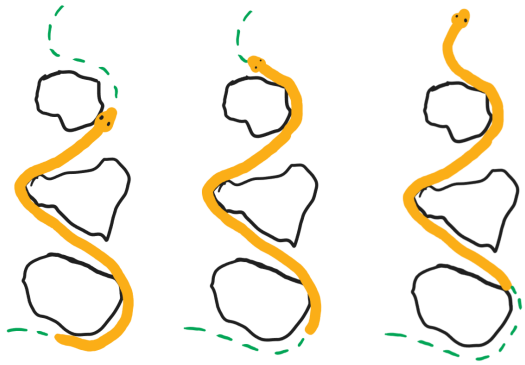


Figure 4.26: A snake (orange) moving along a path (dashed green) in an environment cluttered with obstacles (black), by lateral undulation.

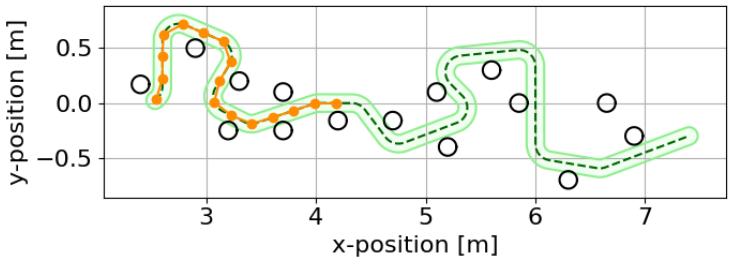


Figure 4.27: The undulation path followed by the snake robot in the demonstration. The path itself is marked as an dashed green line with a shaded area showing the area ideally traced by the snake robots' body as it moves along the path. The cylindrical obstacles are marked by black circles. The initial configuration of the snake robots' spine is shown by orange connected circles.

prevent these kinds of configurations [73].

The endeavor of producing meaningful locomotion with lateral undulation in snake robots warrants a geometric condition that identifies these kinds of undulation paths and configurations. Form closure can be used for this purpose.

The demonstration builds on a physics based simulation run on the MuJoCo physics engine [74]. The intended undulation path and the position of the obstacles are shown in Figure 4.27, and are designed so that the snake robot will remain in the form closed region for an initial portion of the path and exit the form closed region as $t > 31.0$s. For the following demonstration, the robots' trajectory is designed to have 1^{st} order form closure. The physical parameters of the simulated snake robot are given in Table 4.1, and were chosen to resemble the Boa snake robot, as introduced in Chapter 2.

Table 4.1: Physical parameters of the simulated snake robot

Number of links:	15	(unitless)
Link length:	0.2	m
Link width:	0.16	m
Friction coefficient:	0.1	(unitless)
Link mass:	0.4	kg
Maximum actuator torque:	3	Nm

As the undulation path is continuous it is, in general, not possible to overlay an articulated snake robot perfectly on the path. The problem of interpolating an articulated snake robot on a continuous path is addressed in [75], where the head of the robot is placed on the path in a desired position and each joint is consecutively placed on the path behind the head as shown in Figure 4.27. By doing this form of approximation, it is a matter of simple geometry to calculate the desired joint angles for any interpolation of the robot on the path. The path itself was chosen manually as suitable path planning algorithms using form closure are still in development. Two considerations were made during the planning of the undulation:

1. The undulation path is straight within a link length's radius of each obstacle. This is to ensure that the joints of the snake robot are straight as they slide past the obstacle to prevent discontinuities and collisions.

2. The path is created from straight line segments and arcs of constant radius. This is a pattern commonly seen in biological snakes [73] as they attempt to form a shortest possible path between two contact points, but are restricted by a minimum curvature in their body to avoid tissue damage.

Under form closure, the speed at which the shape of the snake robot is propagated backwards along its body is equal to its propulsive speed along the undulation path. In biological snakes this speed typically remains constant during locomotion [76] and as such is set to a constant speed of 0.1 m/s for the following demonstration.

The joint angles $\boldsymbol{\phi}(t)$ of the simulated snake robot are driven to the desired joint angles $\boldsymbol{\phi}^d(t)$ by a PD control loop. While this is a naive approach to the low-level control of the robot, it serves to show the efficacy of form closure as a condition for undulation-based locomotion, even when using simple low-level controllers. This controller configuration also guarantees overall passivity and thus rudimentary stability properties of the closed-loop system. Renderings of the simulation are shown in Figure 4.28. The trajectory of the robots' head compared to its desired position on the undulation path is shown in Figure 4.29, and its deviation from the desired head position is shown in Figure 4.30. The actuator torque applied throughout the simulation is shown in Figure 4.31.

As seen in Figure 4.28 - 4.30, the robot follows the undulation path when applying a lateral undulation gait as long as it remains in the form closed region. Figure 4.30 shows the tracking error between the head position and its desired position, which remains largely between 0.05 m and 0.1 m. The error arises from the fact that while the head remains largely on the path, it lags slightly behind its desired position. This is likely due to friction and the inability of the PD control loop to perfectly track the desired joint angles $\boldsymbol{\phi^d(t)}$.

As the robot loses form closure at $t = 31.0\,\mathrm{s}$, it rapidly deviates from the undulation path, as shown in Figure 4.29 and 4.30. In Figure 4.28, the robot shows significant lateral slippage after losing form closure.

Figure 4.31 shows that during locomotion, the actuation of the joints remains under the saturation limit of 3.0 Nm. Some oscillations are apparent in the actuation, which can partially be attributed to rudimentary tuning of the PD control loop and the presence of friction.

4.4.5 Discussion and Future Works

While this paper focused on planar snake robots inhabiting $\mathcal{W} = \mathbb{R}^2$, all theorems in this paper can be generalized to snake robots operating in $\mathcal{W} = \mathbb{R}^3$ where the space of head poses is expanded to $\mathcal{Q}_N = SE(3)$. In this case it is required to have a minimum of seven contact points between a snake robot and its environment to

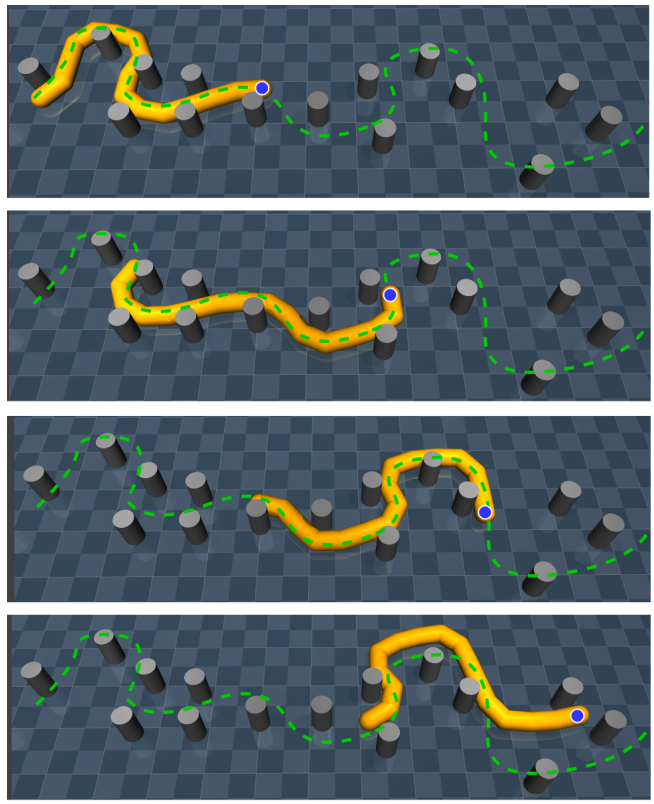


Figure 4.28: Still-frames from the simulation showing the locomotion of the snake robot through the cluttered planar environment at times $t = 0$s, 15s, 30s, 40s. The dashed green line shows the undulation path of the snake robot and the blue circle shows the location of the robots' head.

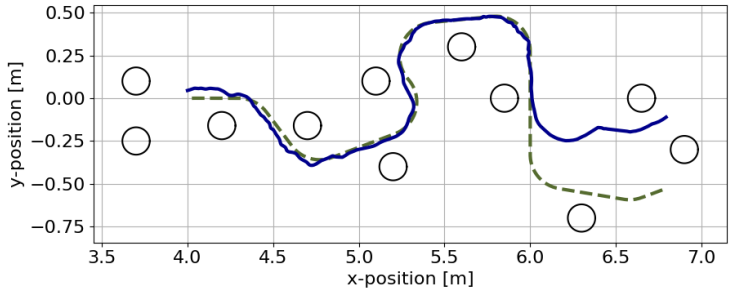


Figure 4.29: The blue line shows the position of the snake robots' head, while the dashed line shows the undulation path. The robot loses form closure as the head passes the point $\boldsymbol{p}_N = (6, 0)$.

achieve 1st order form closure. While the complexity of calculating form closure increases in a 3-dimensional workspace, the structure of the form closure problem remains similar. Thus, the findings in this paper are relevant when studying snake robot modeling in three dimensions.

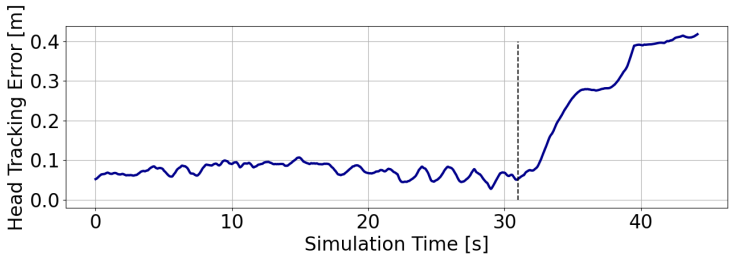


Figure 4.30: The blue line shows the euclidean distance from the head position $\boldsymbol{p}_N$ to its desired position along the undulation path. The dashed line at $t = 31.0$ indicates the time when the snake robot loses form closure.

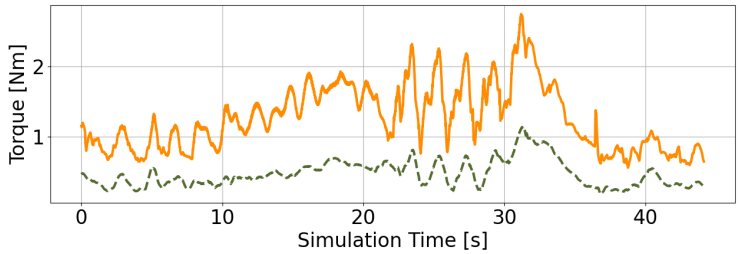


Figure 4.31: The dashed line shows the average absolute actuation torque of the joints over time. The orange line shows the actuation of the joint with the greatest actuation torque at any given time.

As form closure is a strictly geometric condition, it does not consider friction. The effect of this is twofold: Form closure guarantees that the snake robot is immobilized, even when friction is near nonexistent. On the other hand, there intuitively exist form closed configurations where an attempted motion would cause the propulsive forces to be canceled out by equal frictional reaction forces from the obstacles. In this case the robot would be jammed in its current configuration. Consequently, a form closed configuration does guarantee that locomotion is possible in a scenario with friction. The resolution of jammed configurations is treated in some detail in [32], but further research is necessary to identify criteria for jam avoidance.

Biological snakes are often observed to produce undulatory locomotion in cluttered environments using three or less contact points, and in some cases using only a single contact point [72]. This alludes that form closure is an overly strict condition, and that there may exist less strict conditions which still allow for undulatory locomotion. Of particular interest are the notions of *partial form closure* and *force closure* [48, 77].

Partial form closure, as shown in Figure 4.32 is a less strict condition than form closure which constrains the unactuated dynamics of the robot only in directions that are not beneficial to locomotion. Form closure and form boundedness are invariant to travel direction in that it is inconsequential which end of the robot we identify as its head. Thus, the robot can effectively "reverse" along the same

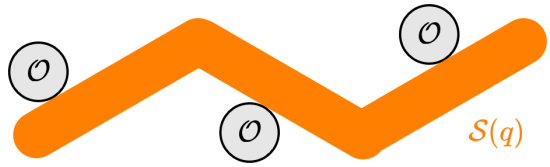


Figure 4.32: An articulated snake robot under partial form closure. Assuming that the snake robot is traveling towards the right, its unactuated dynamics is constrained towards the left, allowing only motions that are beneficial for the robots' locomotion.

trajectory that was applied to achieve its current configuration. While partial form closure may be achieved with fewer contact points than form closure, it does not possess this property.

Force closure is a related concept to form closure which takes friction into account. By including constraining forces from friction, it is possible to identify a superset of the form closed configurations that still guarantees that the unactuated dynamics remain constrained.

In this paper we treated form boundedness from a purely mathematical perspective. As this is a novel concept, further research may be done in developing analytical or numerical methods for identifying form bounded configurations. We also recognize a weakness of form boundedness: The loss of form closure in a given configuration does not guarantee that it will become form bounded before losing its constrainedness entirely.

Although form closure is a well-researched topic, this work is the first to utilize form closure as a condition for locomotion. The development of path planning algorithms and control strategies that build on the concepts of form closure constitutes an interesting topic for future research.

4.4.6 Acknowledgments

We would like to thank Oscar Mørk for his contributions to the design of the snake robot simulator.

4.5 Analytic Computation of Form Closure

Determining whether a configuration is form closed is done numerically by the way of convex optimization in existing literature. This subsection contains only a proposition to how form closure may be determined analytically, but no explicit and formal proof. Because of this, it is discussed here in the future works section, and not as an integral part of the thesis.

We recall that the act of determining if a configuration is under 1^{st} order form closure is equivalent to determining if a polyhedral cone $\mathcal{V}^1 \subset \mathbb{R}^k$ in k-dimensional c-space contains only the origin such that $\mathcal{V}^1 = \{0\}$. The polyhedral cone is defined by the intersection of a series of half spaces $\mathcal{N}_i$ such that $\mathcal{V}^1 = \cap_i \mathcal{N}_i$.

Any vector $\boldsymbol{e} \in \mathcal{V}^1$ is said to be an extreme ray of $\mathcal{V}^1$ if it cannot be written as the linear combination of any two linearly independent vectors in $\mathcal{V}^1$. In a 3-dimensional c-space, the extreme rays lie on the the edges of the polyhedral cone. This notion can be generalized to k-dimensional space, where the the extreme rays represent the equivalent of an edge in k-dimensional space. We pave the way for our computations by making some assumptions:

- A polyhedral cone contains only its origin if and only if it has no extreme rays. By this assumption, the task of computing form closedness equates to proving that the polyhedral cone $\mathcal{V}^1$ has no extreme rays.
- If a ray $\boldsymbol{v}$ is in the polyhedral cone such that $\boldsymbol{v} \in \mathcal{V}^1$ then any positive scaling of the ray is also in the cone.
- The converse is also true, in that for any ray outside the polyhedral cone, any positive scaling is also outside the cone.
- Checking whether a point is inside a polyhedral cone or not is computationally efficient as it only requires the evaluation of the defining inequalities of the polyhedral cone.

Each half-space $\mathcal{N}_i$ is bounded by a hyperplane $\delta\mathcal{N}_i$. The extreme rays can only exist at the intersection of $k-1$ of the bounding hyperplanes, where each of the intersecting hyperplanes are non-parallel.

By computing the intersection of all "$k-1$-sized" subsets of hyperplanes we create a set of rays where the extreme rays might exist. We denote this subset $\mathcal{E} \subset \mathbb{R}^k$ which we term the *critical rays*. Any extreme ray $\boldsymbol{e} \in \mathcal{V}^1$ also satisfies $\boldsymbol{e} \in \mathcal{E}$.

This leads to our first proposition: $\mathcal{E} \cap \mathcal{V}^1 = \emptyset \implies \mathcal{V}^1 = \{0\}$. If the critical rays are not a part of the polyhedral cone, then no extreme ray is part of the polyhedral cone, in which case we have form closure. The space $\mathcal{E}$ consists of a union of rays, and it is sufficient to check if a single point on each ray is in the polyhedral cone to check if each respective ray is in the polyhedral cone. Because $\mathcal{E}$ consists of only rays, the *unit critical rays* $\hat{\mathcal{E}} = \mathcal{E} \cap \mathcal{S}^{k-1}$, where $\mathcal{S}^{k-1}$ is the unit hyper-sphere in $k-1$ dimensions, is a finite set.

This leads to our second proposition: $\hat{\mathcal{E}} \cap \mathcal{V}^1 = \emptyset \implies \mathcal{E} \cap \mathcal{V}^1 = \emptyset$. If the unit critical rays are not in a polyhedral cone, then none of the critical rays are in the polyhedral cone.

By using the first and second proposition as lemma, we find our third proposition: $\hat{\mathcal{E}} \cap \mathcal{V}^1 = \emptyset \implies \mathcal{V}^1 = \{0\}$. If none of the unit critical rays $\hat{\mathcal{E}}$ are in the polyhedral cone, then the cone must contain only its origin, that again confirms form closedness. The computation of $\hat{\mathcal{E}}$ is numerically feasible, and so is the computation $\hat{\mathcal{E}} \cap \mathcal{V}^1$, that gives an analytic test for form closure that is numerically feasible.

A formal proof of the aforementioned propositions, would result in an analytic test for form closure, that may be more efficient that previous methods based on convex optimization.

As a final remark, we would like to point out the geometrical significance of the critical rays $\mathcal{E}$. Recall that in 1^{st} order form closure analysis, each point in c-space represents a twist. The critical rays can, in a geometrical sense, be though of as critical twists. The critical twists equate to twists that, when applied to a snake robot in physical space, causes a roll-slide motion (c.f. Section 4.3) in at least $k - 1$ contact points.

These, in a sense, represent the twists where the fewest obstacles contribute to constraining the body. If the body is immobilized for these critical twists, then any other hypothetical twist will have more obstacles counteracting its motion and thus be physically impossible. Thus checking if the body is immobilized for the critical twists is sufficient to conclude that it is completely immobilized, i.e., in form closure.

REFERENCES

[1] Carl Gans. "How Snakes Move". In: *Scientific American* 222.6 (1970), pp. 82–99. ISSN: 00368733, 19467087. URL: http://www.jstor.org/stable/24925828 (visited on 05/30/2023).

[2] James Gray. "The mechanism of locomotion in snakes". In: *Journal of experimental biology* 23.2 (1946), pp. 101–120.

[3] Jindong Liu, Yuchuang Tong, and Jinguo Liu. "Review of snake robots in constrained environments". In: *Robotics and Autonomous Systems* 141 (2021), p. 103785. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2021.103785. URL: https://www.sciencedirect.com/science/article/pii/S0921889021000701.

[4] Pål Liljebäck et al. "Mamba-A waterproof snake robot with tactile sensing". In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2014, pp. 294–301.

[5] Aksel Andreas Transeth et al. "Snake Robot Obstacle-Aided Locomotion: Modeling, Simulations, and Experiments". In: *IEEE Transactions on Robotics* 24.1 (2008), pp. 88–104. DOI: 10.1109/TRO.2007.914849.

[6] Jostein Løwer, Damiano Varagnolo, and Øyvind Stavdahl. "Improved Jacobian matrix estimation applied to snake robots". In: *Frontiers in Robotics and AI* 10 (2023).

[7] Jostein Løwer et al. "Proprioceptive contact force and contact point estimation in a stationary snake robot". In: *IFAC-PapersOnLine* 55.38 (2022), pp. 160–165.

[8] Jostein Løwer et al. "A Novel Model for Link Dynamics in Planar Snake Robots Using Internal Constraint Force Sensing". In: *Proceedings 2023 CCTA*. IEEE.

[9] Jostein Løwer. *Boa Snake Robot*. https://github.com/Boa-Snake-Robot. 2023.

[10] Frederik Veslum. "Assessment of the Mamba snake robot sensor system". unpublished. 2020.

[11] Pål Liljebäck et al. *Snake robots: modelling, mechatronics, and control*. Springer Science & Business Media, 2012. Chap. 2.

[12] Michael C. Yip and David B. Camarillo. "Model-Less Feedback Control of Continuum Manipulators in Constrained Environments". In: *IEEE Transactions on Robotics* 30.4 (2014), pp. 880–889. DOI: 10.1109/TRO.2014.2309194.

[13] Michael Yip and David Camarillo. "Model-Less Hybrid Position/Force Control: A Minimalist Approach for Continuum Manipulators in Unknown, Constrained Environments". In: *IEEE Robotics and Automation Letters* 1 (July 2016), pp. 1–1. DOI: 10.1109/LRA.2016.2526062.

[14] Ryo Ariizumi and Fumitoshi Matsuno. "Dynamic analysis of three snake robot gaits". In: *IEEE Transactions on Robotics* 33.5 (2017), pp. 1075–1087.

[15] H. Durrant-Whyte and T. Bailey. "Simultaneous localization and mapping: part I". In: *IEEE Robotics Automation Magazine* 13.2 (2006), pp. 99–110. DOI: 10.1109/MRA.2006.1638022.

[16] Tony Owen. "Biologically Inspired Robots: Snake-Like Locomotors and Manipulators by Shigeo Hirose Oxford University Press, Oxford, 1993, 220pages, incl. index (£ 40)". In: *Robotica* 12.3 (1994), pp. 282–282.

[17] Zeki Y Bayraktaroglu. "Snake-like locomotion: Experimentations with a biologically inspired wheel-less snake robot". In: *Mechanism and Machine Theory* 44.3 (2009), pp. 591–602.

[18] Tito Lu Tang Chen, Shu-Hung Liu, and Jia-Yush Yen. "A bio-mimetic snake-like robot: Sensor based gait control". In: *2008 IEEE Workshop on Advanced robotics and Its Social Impacts*. IEEE. 2008, pp. 1–6.

[19] Aksel Andreas Transeth, Pal Liljeback, and Kristin Y Pettersen. "Snake robot obstacle aided locomotion: An experimental validation of a non-smooth modeling approach". In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2007, pp. 2582–2589.

[20] Sigurd Aksnes Fjerdingen et al. "Adaptive snake robot locomotion: A benchmarking facility for experiments". In: *European Robotics Symposium 2008*. Springer. 2008, pp. 13–22.

[21] Stefan R Taal, Hiroya Yamada, and Shigeo Hirose. "3 axial force sensor for a semi-autonomous snake robot". In: *2009 IEEE International Conference on Robotics and Automation*. IEEE. 2009, pp. 4057–4062.

[22] Pål Liljeback et al. "Snake robot locomotion in environments with obstacles". In: *IEEE/ASME Transactions on Mechatronics* 17.6 (2011), pp. 1158–1169.

[23] Pal Liljeback, Oyvind Stavdahl, and Anders Beitnes. "SnakeFighter-development of a water hydraulic fire fighting snake robot". In: *2006 9th International Conference on Control, Automation, Robotics and Vision*. IEEE. 2006, pp. 1–6.

[24] Juan Gonzalez-Gomez et al. "Toward the sense of touch in snake modular robots for search and rescue operations". In: *Proc. ICRA 2010 Workshop "Modular Robots: State of the Art*. 2010, pp. 63–68.

[25] Irja Gravdahl et al. "Modeling for Hybrid Obstacle-Aided Locomotion (HOAL) of Snake Robots". In: *IFAC-PapersOnLine* 55.20 (2022). 10th Vienna International Conference on Mathematical Modelling MATHMOD 2022, pp. 247–252. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2022.09.103. URL: https://www.sciencedirect.com/science/article/pii/S2405896322012940.

[26] Pal Liljeback et al. "Hybrid Modelling and Control of Obstacle-Aided Snake Robot Locomotion". In: *IEEE Transactions on Robotics* 26.5 (2010), pp. 781–799. DOI: 10.1109/TRO.2010.2056211.

[27] Fasheng Wang et al. "Unscented Particle Filter for Online Total Image Jacobian Matrix Estimation in Robot Visual Servoing". In: *IEEE Access* 7 (2019), pp. 92020–92029. DOI: 10.1109/ACCESS.2019.2927413.

[28] K. Hosoda and M. Asada. "Versatile visual servoing without knowledge of true Jacobian". In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*. Vol. 1. 1994, 186–193 vol.1. DOI: 10.1109/IROS.1994.407392.

[29] D.I. Kosmopoulos. "Robust Jacobian matrix estimation for image-based visual servoing". In: *Robotics and Computer-Integrated Manufacturing* 27.1 (2011), pp. 82–87. ISSN: 0736-5845. DOI: https://doi.org/10.1016/j.rcim.2010.06.013. URL: https://www.sciencedirect.com/science/article/pii/S0736584510000645.

[30] Azad Shademan, Amir-massoud Farahmand, and Martin Jägersand. "Robust Jacobian estimation for uncalibrated visual servoing". In: *2010 IEEE International Conference on Robotics and Automation*. 2010, pp. 5564–5569. DOI: 10.1109/ROBOT.2010.5509911.

[31] Jiang Qian and Jianbo Su. "Online estimation of image Jacobian matrix by Kalman-Bucy filter for uncalibrated stereo vision feedback". In: *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*. Vol. 1. 2002, 562–567 vol.1. DOI: 10.1109/ROBOT.2002.1013418.

[32] Pal Liljeback, Kristin Y. Pettersen, and Øyvind Stavdahl. "Modelling and control of obstacle-aided snake robot locomotion based on jam resolution". In: *2009 IEEE International Conference on Robotics and Automation*. 2009, pp. 3807–3814. DOI: 10.1109/ROBOT.2009.5152273.

[33] Jan Kronqvist, Andreas Bernal David E. andLundell, and Ignacio E. Grossmann. "A review and comparison of solvers for convex MINLP". In: *Optimization and Engineering* (2019).

[34] S.J Julier and J.K Uhlmann. "Unscented filtering and nonlinear estimation". In: 92.3 (2004), pp. 401–422. ISSN: 0018-9219.

[35] E.A. Wan and R. Van Der Merwe. "The unscented Kalman filter for nonlinear estimation". In: *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*. 2000, pp. 153–158. DOI: 10.1109/ASSPCC.2000.882463.

[36] Franz-Georg Wieland et al. "On structural and practical identifiability". In: *Current Opinion in Systems Biology* 25 (2021), pp. 60–69. ISSN: 2452-3100. DOI: https://doi.org/10.1016/j.coisb.2021.03.005. URL: https://www.sciencedirect.com/science/article/pii/S245231002100007X.

[37] Erwin Coumans. *Bullet Real-Time Physics Simulation*. https://github.com/bulletphysics/bullet3. 2008.

[38] Akshay Agrawal et al. "A rewriting system for convex optimization problems". In: *Journal of Control and Decision* 5.1 (2018), pp. 42–60.

[39] Steven Diamond and Stephen Boyd. "CVXPY: A Python-embedded modeling language for convex optimization". In: *Journal of Machine Learning Research* 17.83 (2016), pp. 1–5.

[40] Roger Labbe. *Filterpy*. https://github.com/rlabbe/filterpy. 2017.

[41] Jaroslav Hrdina, Aleš Návrat, and Petr Vašík. "Control of 3-link robotic snake based on conformal geometric algebra". In: *Advances in Applied Clifford Algebras* 26 (2016), pp. 1069–1080.

[42] Takeshi Kano, Naoki Matsui, and Akio Ishiguro. "3D Movement of Snake Robot Driven by Tegotae-Based Control". In: Springer, July 2019, pp. 346–350. ISBN: 978-3-030-24740-9. DOI: 10.1007/978-3-030-24741-6_35.

[43] Mikhail V. Khlebnikov. "Control of Linear Systems Subjected to Exogenous Disturbances: Combined Feedback". In: *IFAC-PapersOnLine* 49.13 (2016). 12th IFAC Workshop on Adaptation and Learning in Control and Signal Processing ALCOSP 2016, pp. 111–116. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2016.07.936. URL: https://www.sciencedirect.com/science/article/pii/S2405896316312241.

[44] Rudolph Emil Kalman. "A New Approach to Linear Filtering and Prediction Problems". In: *Transactions of the ASME–Journal of Basic Engineering* 82.Series D (1960), pp. 35–45.

[45] Ian Chivers and Jane Sleightholme. "An introduction to Algorithms and the Big O Notation". In: *Introduction to programming with Fortran*. Springer, 2015, pp. 359–364.

[46] Timothy A. Davis. *Direct methods for sparse linear systems*. Society for Industrial and Applied Mathematics, 2006.

[47] E. Rimon and J. Burdick. "On force and form closure for multiple finger grasps". In: *Proceedings of IEEE International Conference on Robotics and Automation*. Vol. 2. 1996, 1795–1800 vol.2. DOI: 10.1109/ROBOT.1996.506972.

[48] Antonio Bicchi. "On the closure properties of robotic grasping". In: *The International Journal of Robotics Research* 14.4 (1995), pp. 319–334.

[49] Antonio Bicchi and Vijay Kumar. "Robotic grasping and contact: A review". In: *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065)*. Vol. 1. IEEE. 2000, pp. 348–353.

[50] Elon Rimon and Joel W. Burdick. "A configuration space analysis of bodies in contact-I. 1st order mobility". In: *Mechanism and Machine Theory* 30 (6 1995), pp. 897–912. ISSN: 0094114X. DOI: 10.1016/0094-114X(95)00002-G.

[51] Elon Rimon and Joel W. Burdick. "A configuration space analysis of bodies in contact-II. 2ND order mobility". In: *Mechanism and Machine Theory* 30 (6 1995), pp. 913–928. ISSN: 0094114X. DOI: 10.1016/0094-114X(95)00003-H.

[52] Shinichi Hirai. "Analysis and planning of manipulation using the theory of polyhedral convex cones". PhD thesis. Kyoto University, 1991.

[53] Zhen Huang, Qinchuan Li, and Huafeng Ding. "Basics of Screw Theory". In: *Mechanisms and Machine Science* (Jan. 2013), pp. 1–16. DOI: 10.1007/978-94-007-4201-7_1.

[54] E.N. Kuznetsov. "Underconstrained structural systems". In: *International Journal of Solids and Structures* 24.2 (1988), pp. 153–163. ISSN: 0020-7683. DOI: https://doi.org/10.1016/0020-7683(88)90026-1. URL: https://www.sciencedirect.com/science/article/pii/0020768388900261.

[55] F. Reuleaux. *Kinematics of Machinery*. Macmillan, 1876.

[56] Chandler Davis. "Theory of positive linear dependence". In: *American Journal of Mathematics* 76.4 (1954), pp. 733–746.

[57] Shigeo Hirose and Yoji Umetani. "Kinematic control of active cord mechanism with tactile sensors". In: *Transactions of the Society of Instrument and Control Engineers* 12.5 (1976), pp. 543–547.

[58] Takuro Takanashi et al. "Obstacle-Aided Locomotion of a Snake Robot Using Piecewise Helixes". In: *IEEE Robotics and Automation Letters* 7.4 (2022), pp. 10542–10549. DOI: 10.1109/LRA.2022.3194689.

[59] Trushant Majmudar et al. "Experiments and theory of undulatory locomotion in a simple structured medium". In: *Journal of the Royal Society, Interface / the Royal Society* 9 (Feb. 2012), pp. 1809–23. DOI: 10.1098/rsif.2011.0856.

[60] Xanthippi Markenscoff, Luqun Ni, and Christos H. Papadimitriou. "The Geometry of Grasping". In: *The International Journal of Robotics Research* (1990).

[61] Jurek Czyzowicz, Ivan Stojmenovic, and Jorge Urrutia. "IMMOBILIZING A POLYTOPE". In: *Workshop on Algorithms and Data Structures* (1991).

[62] M Chasles. "Note sur les propriétés générales du système de deux corps semblables entr'eux". In: Bulletin des Sciences Mathématiques, Astronomiques, Physiques et Chemiques, 1830, pp. 321–326.

[63] Baxi Chong et al. *Gait design for limbless obstacle aided locomotion using geometric mechanics*. 2023. arXiv: 2302.06561 [cs.RO].

[64] Tatsuya Takemori, Motoyasu Tanaka, and Fumitoshi Matsuno. "Hoop-Passing Motion for a Snake Robot to Realize Motion Transition Across Different Environments". In: *IEEE Transactions on Robotics* 37.5 (2021), pp. 1696–1711. DOI: 10.1109/TRO.2021.3063438.

[65] Kristian G. Hanssen et al. "Path Planning for Perception-Driven Obstacle-Aided Snake Robot Locomotion". In: *2020 IEEE 16th International Workshop on Advanced Motion Control (AMC)*. 2020, pp. 98–104. DOI: 10.1109/AMC44022.2020.9244366.

[66] Belal A. Elsayed et al. "Mobile Manipulation Using a Snake Robot in a Helical Gait". In: *IEEE/ASME Transactions on Mechatronics* 27.5 (2022), pp. 2600–2611. DOI: 10.1109/TMECH.2021.3114168.

[67] Dongfang Li et al. "Adaptive Path Following Controller of a Multijoint Snake Robot Based on the Improved Serpenoid Curve". In: *IEEE Transactions on Industrial Electronics* 69.4 (2022), pp. 3831–3842. DOI: 10.1109/TIE.2021.3075851.

[68] Tatsuya Takemori, Motoyasu Tanaka, and Fumitoshi Matsuno. "Adaptive Helical Rolling of a Snake Robot to a Straight Pipe With Irregular Cross-Sectional Shape". In: *IEEE Transactions on Robotics* 39.1 (2023), pp. 437–451. DOI: 10.1109/TRO.2022.3189224.

[69] Ivan Virgala et al. "A snake robot for locomotion in a pipe using trapezium-like travelling wave". In: *Mechanism and Machine Theory* 158 (2021), p. 104221. ISSN: 0094-114X. DOI: https://doi.org/10.1016/j.mechmachtheory.2020.104221. URL: https://www.sciencedirect.com/science/article/pii/S0094114X20304389.

[70] Guodong Qin et al. "Adaptive trajectory control of an under-actuated snake robot". In: *Applied Mathematical Modelling* 106 (2022), pp. 756–769. ISSN: 0307-904X. DOI: https://doi.org/10.1016/j.apm.2022.02.001. URL: https://www.sciencedirect.com/science/article/pii/S0307904X22000646.

[71] Fabian Reyes and Shugen Ma. "On planar grasping with snake robots: Form-closure with enveloping grasps". In: *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*. IEEE. 2014, pp. 556–561.

[72] J. GRAY and H. W. LISSMANN. "The Kinetics of Locomotion of the Grass-Snake". In: *Journal of Experimental Biology* 26.4 (Feb. 1950), pp. 354–367. ISSN: 0022-0949. DOI: 10.1242/jeb.26.4.354. eprint: https://journals.biologists.com/jeb/article-pdf/26/4/354/1333678/354.pdf. URL: https://doi.org/10.1242/jeb.26.4.354.

[73] Perrin E. Schiebel, Alex M. Hubbard, and Daniel I. Goldman. "Comparative study of snake lateral undulation kinematics in model heterogeneous terrain." In: *Integrative and comparative biology* (2020).

[74] Emanuel Todorov, Tom Erez, and Yuval Tassa. "MuJoCo: A physics engine for model-based control". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012, pp. 5026–5033. DOI: 10.1109/IROS.2012.6386109.

[75] Pal Liljeback et al. "A control framework for snake robot locomotion based on shape control points interconnected by Bézier curves". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012, pp. 3111–3118. DOI: 10.1109/IROS.2012.6386117.

[76] Bruce C Jayne. "What Defines Different Modes of Snake Locomotion?" In: *Integrative and Comparative Biology* 60.1 (Apr. 2020), pp. 156–170. ISSN: 1540-7063. DOI: 10.1093/icb/icaa017. eprint: https://academic.oup.com/icb/article-pdf/60/1/156/33549853/icaa017.pdf. URL: https://doi.org/10.1093/icb/icaa017.

[77] Bruno Siciliano, Oussama Khatib, and Torsten Kröger. *Springer handbook of robotics*. Vol. 200. Springer, 2008.

ISBN 978-82-326-7508-1 (printed ver.)
ISBN 978-82-326-7507-4 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU

Norwegian University of
Science and Technology