NTNU
Det skapende universitet

Styringsprinsipper for slangerobot ved hindringsbasert fremdrift

Adrian Olsen

Master i teknisk kybernetikk

Innlevert:	Juni 2012
Hovedveileder:	Øyvind Stavdahl, ITK
Medveileder:	Pål Liljebäck, ITK

Norges teknisk-naturvitenskapelige universitet
Institutt for teknisk kybernetikk

NTNU – Trondheim
Norwegian University of
Science and Technology

Control Strategies for Obstacle-aided Locomotion in Snake Robots

Adrian Olsen

Project in Engineering Cybernetics
Trondheim, June 2012

Supervisor: Oyvind Stavdahl
CoSupervisor: Paal Liljeback

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

NTNU Norwegian University of Science and Technology

Project in Engineering Cybernetics

Control Strategies for Obstacle-aided Locomotion in Snake Robots
Faculty of Information Technology,
Mathematics and Electrical Engineering

Institute for Technical Cybernetics

Thesis written at NTNU, 2012

Printed by TAPIR-trykk

NTNU
Norges teknisk-naturvitenskapelige
universitet

Fakultet for informasjonsteknologi,
matematikk og elektroteknikk
Institutt for teknisk kybernetikk

HOVEDOPPGAVE

Kandidatens navn: **Adrian Olsen**

Fag: **Teknisk kybernetikk**

Oppgavens tittel (norsk): **Styringsstrategier for hindringsbasert fremdrift med slangeroboter**

Oppgavens tittel (engelsk): **Control Strategies for Obstacle-aided Locomotion in Snake Robots**

Oppgavens tekst:

Inspired by biological snake locomotion, snake robots carry the potential of meeting the growing need for robotic mobility in unknown and challenging environments. These mechanisms typically consist of serially connected joint modules capable of bending in one or more planes. The many degrees of freedom of snake robots make them difficult to control, but provide traversability in irregular environments that surpasses the mobility of the more conventional wheeled, tracked and legged forms of robotic mobility.

A unique feature of snake robot locomotion compared to other forms of robotic mobility is that ground irregularities are beneficial for the propulsion since they provide push-points for the robot. The term *obstacle-aided locomotion* was previously introduced by the snake robot research group at NTNU/SINTEF and captures the essence of this concept. The majority of previous research on control of snake robots has focused on open-loop control strategies for flat surface motion aimed at resembling gaits displayed by biological snakes. Only a few works in the literature consider control strategies related to obstacle-aided snake locomotion.

The goal of this thesis is to study the optimality of the shape of a snake robot during locomotion in an obstacle environment. Optimality considerations related to the shape of a snake robot with respect to its environment have received very limited focus in previous literature, but are highly relevant for practical applications of snake robots. The thesis is motivated by the following two subproblems:

1. How can we characterize the optimality of the shape of a snake robot with respect to its obstacle environment?

2. How can we achieve “optimal” shape control of a snake robot in an obstacle environment?

In order to study these two subproblems, the candidate shall:

1. Carry out a literature survey and describe previous work related to obstacle-aided locomotion and previous approaches for shape-based control of snake robots.

2. Propose various *optimality criteria* for the shape of a snake robot with respect to a fully

known set of obstacles. The optimality criteria should take into account properties related to the propulsive forces on the robot from its environment, and also the energy efficiency of the expected locomotion.

3. Propose one or several control strategies for shape control of a snake robot in a fully known obstacle environment aimed at being optimal with respect to one (or several) of the proposed *optimality criteria*.

4. Investigate the performance of the control strategy through simulations with an existing simulation model of snake robot locomotion.

5. If time permits it, carry out experiments with one of the existing snake robots at NTNU/SINTEF in order to study the validity of the derived theoretical results.

Oppgaven gitt:	09.01.2012
Besvarelsen leveres:	xx.06.2012

Utført ved Institutt for Teknisk kybernetikk

Faglig veiledere: Førsteamanuensis Øyvind Stavdahl, NTNU Inst. for Teknisk kybernetikk
PhD Pål Liljebäck, NTNU Inst. for Teknisk kybernetikk

Trondheim, den 09.01.2012

Øyvind Stavdahl
Faglærer

Abstract

This study investigates the optimality of the shape of a snake robot in relation to the obstacles in the environment around it. A certain focus is given to the challenge of finding an optimal set of contact forces based on some optimality criteria. A controller scheme for realizing these contact forces on a dynamic snake model with obstacle compliance was presented, but further development and testing is required to have a definite conclusion as to the success of this particular approach.

Sammendrag

Denne studien undersøker hvorvidt en slangerobots form er optimal med hensyn til hindringsmiljøet den befinner seg i. Et spesielt fokus blir gitt til utfordringen med å finne et optimalt sett med kontaktkrefter gitt forskjellige optimalitetskriterier. En regulator som realiserer de ønskede kontaktkreftene på en dynamisk slangerobotmodell i et hindringsmiljø foreslås, men videre utvikling og testing er nødvendig for å treffe en definitiv konklusjon om hvorvidt denne framgangsmåten er egnet eller ikke.

Acknowledgements

This report presents the results of my master thesis on obstacle-aided locomotion of snake robots which has been carried out in the period January 2012 - June 2012. The work has been conducted at the Department for Engineering Cybernetics at the Norwegian University of Science and Technology (NTNU) in Trondheim, Norway.

I would like to extend my gratitude to my supervisors Øyvind Stavdahl and Pål Liljebäck who introduced me to the world of snake robotics and this semester brought me further into the research group and into the exciting field that is snake robotics. I appreciate their many ideas, solid advice and always enthusiastic attitude. I also would like to thank Eleni Kelasidi for much appreciated advice towards the end of the project. Gratitude is also extended to Martin Mortensen for loyal friendship and epic proof reading.

Adrian Olsen - June 2012

Contents

Chapter 1
Introduction

1.1 Background and motivation

Snakes are fascinating creatures, different in many ways from other animals. They are among a select group of animals that move without the use of limbs, yet this does not seem to hinder them when it comes to moving swiftly, efficiently and flexibly. Despite unevenly and cluttered terrain, like heavy underbrush, gravel or along trees, snakes show a remarkable ability to overcome obstacles and even use them to their benefit for achieving propulsion. The biological study of snakes is the foundation on which the research into snake robots rests.

A possible definition for a snake robot is: *A snake robot is a mechanism of serially connected joint modules capable of bending in one or more planes*[1]. Each joint module can be viewed as an approximation to one piece of the vertebrae of a snake. Snake robots are scientifically interesting for several reasons. Firstly, there is a certain fascination of snakes and their unique and elegant way of movement, which leads to a desire to realize this within an artificial mechanism. Secondly, the shape of a snake, is replicated by a number of things we surround us with, e.g. ropes, hoses and wires. An understanding of snake movement could lead to improved automation and

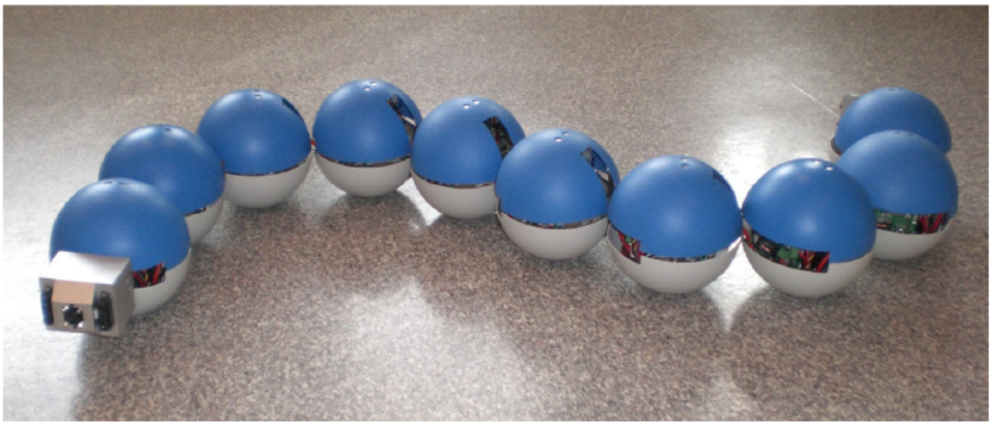


Figure 1.1: The snake robot Kulko developed at NTNU for locomotion in unstructured environments.

control of objects such as a fire hose. Consider a burning building too hot or damaged to risk sending in fire men. A robotic fire hose with vision sensors could enable the extinguishing of the fire from a safe distance. Thirdly, to increase the repertoire of robotic movement. We have already succeeded in making a number of robots with wheels, belts and a number of legs. However, snake-like robots, just like their animal counterparts, bring a new form of locomotion to the table. A snake-like robot could be able to move through pipes and narrow holes, navigate difficult terrain like mud or debris and lift its body to traverse obstacles or climb in the vertical direction. The range of wheeled or biped robots are limited when it comes to tasks such as these.

Research on snake robots have been conducted for several decades. The pioneer in the field of snake robotics was Shigeo Hirose who developed the first snake robot in 1972 [2]. In the wake of his endeavours, several complex and impressive robots have been developed. However, these robots are still confined to the safe and controlled surroundings of the lab and there are no applications of a snake robot that serves any practical purpose as of yet. There are several reasons for this, chief among them is the number of degrees of freedom a snake robot typically exhibits and that the interaction with the environment is in general more complex than that of other robots. To develop a snake robot that can function in the unstructured environments of the real world would require good models, robust, yet flexible controllers and a snake robot built to withstand water, dust and other dangers.

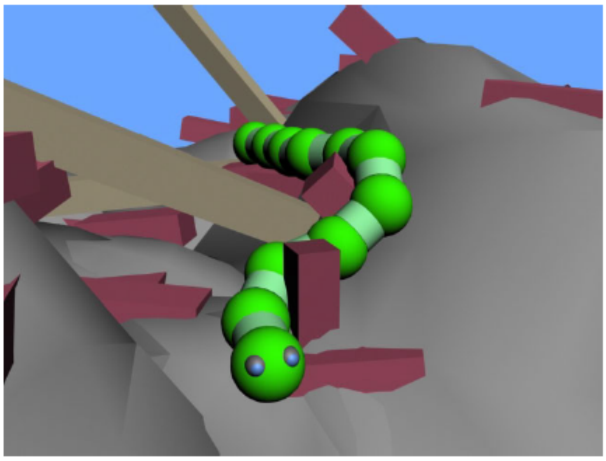


Figure 1.2: Snake robot locomotion in an unstructured environment.

This thesis targets the research into snake robot control with obstacles and specifically obstacle-aided locomotion. The term obstacle-aided locomotion is introduced and defined in Transeth et al. 2008a [3] as *"snake robot locomotion where the snake robot utilizes walls or other external objects, apart from the flat ground, for means of propulsion"*. Snakes are somewhat unique in their ability to treat obstacles not as hindrances, but rather as beneficial push-points. To realize the same behaviour in snake robots has been a goal in several studies, but due to complex models and controllers this is non-trivial.

This study aims to investigate the optimality of a snake robot's shape during locomotion in an obstacle environment and based on this, develop a controller that can control the snake during obstacle-aided locomotion.

1.2 Structure of this thesis

The thesis is structured as follows:

- **Chapter 2** contains a literature review on previous work and theory focused on snake robot locomotion in an unstructured environment.

- In **Chapter 3** we present a simple model of snake robot locomotion. The model is simplified by assuming static conditions which makes investigating the optimal shape of a snake robot easier.

- **Chapter 4** details a controller scheme that consists of Control Allocation method to find an optimal set of contact forces, a torque distribution algorithm giving a mapping from the static model to a dynamic N link model and an outline of the controller scheme.

- **Chapter 5** presents the simulation results of the optimality investigation and simulation of the dynamic system given the controller from Chapter 4.

- With **Chapter 6** we summarize the study with concluding remarks and notes on future work

Chapter 2
Previous work and theory

The following section provides an overview of previous works related to this study, that is, snake robot locomotion in an unstructured environment and shape based control of snake robots. It is not within the scope of this study to provide a more general review of snake robot literature and we refer to [1] or [4] for a broader overview.

A snake robot operating under realistic conditions must be able to navigate complicated and unstructured environment. However, the majority of studies done on snake robotics have investigated flat-surface locomotion. In order to realize the goal of practical snake robot appliances, more studies on control in unstructured environments should be undertaken. Unstructured environments adds to the complexity of the designing, modelling and controlling of the snake robot compared to that of flat-surface locomotion. In an unstructured environment the snake robot is required to sense the environment, e.g. by vision and contact sensors, and adapt its locomotion pattern accordingly. Thus, more advanced contact models and locomotion control are needed. We shall divide the studies done on snake robots in an unstructured environment based on how obstacles are handled, either by avoiding them or by utilizing them as push points, as this seems to be a critical assumption for the derived control approach.

2.1 Snake Robot Locomotion based on Obstacle Avoidance

In the field of mobile robotics the traditional approach to dealing with obstacles is to try to avoid them. Collisions might make the robot unable to progress and cause damage to equipment. For snake robots some challenges arises. Due to the oscillatory motion of the head it can be difficult to gain consistent sensory information of the area in front of the snake robot.

Hasanzadeh and Tootoonchi 2008 [5] introduced the Forward Head Serpentine (FHS) gait which aimed to minimize oscillations of the head link. Sensory equipment are generally attached to the head link and the FHS gait therefore aims to improve the measurements of these sensors by keeping the head link as fixed as possible in the direction of motion. A two-level PID control scheme was used to implement obstacle avoidance, where the snake robot entered ”obstacle avoidance”-mode when close to an obstacle.

Wu and Ma 2011 [6] developed a snake robot with contact force sensors along the snake and an IR-sensor at the head link and controlled it using a Central Pattern Generator (CPG) controller. A CPG-controller generates rhythmic control signals as input and has been studied in relation to snake-like robots on a number of occasions [7], [8], [9]. The snake robot was modelled in 3D and was able to adapt to changes in ground curvature so that if one link was temporarily suspended in the air, local link angle adaptation would occur, rotating neighbouring links so that the suspended link regained ground contact. For obstacle avoidance only 2D-scenarios were considered and the snake robot was found to successfully detect and avoid obstacles.

Several studies have implemented obstacle avoidance in snake robot by the use of Artificial Potential Field theory (APF). Briefly explained, APF models imaginary force fields around objects that are either repulsive or attractive on the robot. The target of motion, e.g. a waypoint, emits an attractive force field while obstacles, other robots or the robot itself emits repulsive force fields. The strength of these forces might increase as the robot gets closer. Ye et al. 2010 [10] utilized APF principles to implement a controller capable of obstacle avoidance where the repulsive forces from the obstacles directly influences the heading of the snake robot as it comes closer to the obstacle. Yagnik et al. 2010 [11] implemented APF by two-dimensional Gaussian functions and used an optimization technique called Simulated Annealing that enabled the snake robot to get out of local minimas created by a standard APF approach, i.e. a scenario where the repulsive forces from nearby obstacles renders the robot unable to move.

Instead of completely avoiding collisions with obstacles a less strict approach can be taken where the robot is allowed to collide with the obstacle, but the collision should be controlled so that no damage to the robot occurs. This approach was termed *obstacle accomodation* and investigated by Shan and Koren 1993 [12]. A kinematic approach to the problem was considered, where the constraints imposed on the snake robot by the obstacles was modelled. In Shan and Koren 1995 [13] a general inverse kinematics solution to a 2D robot in contact with obstacles was formulated.

2.2 Obstacle-aided Snake Robot Locomotion

Natural snakes seems to exploit the obstacles in their surroundings, using stones or branches as push points to move faster and more efficiently. This property of snake locomotion has inspired an approach where obstacles are not seen as a hindrance for the snake robot, but as a potential push point. To understand the concept, it can be beneficial to visualize the simple scenario

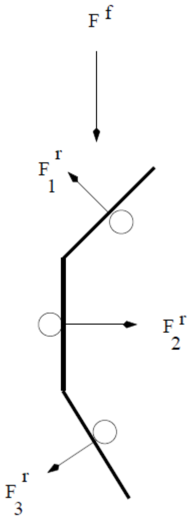


Figure 2.1: A snake body segment in contact with three obstacles. Image source [15]

shown in 2.1. The issues presented in Gray 1946 [14] shall be used to present the idea of obstacle-aided locomotion.

In his broad study of snake locomotion, Gray studied how external forces could be utilized to achieve forward propulsion even though these forces acted normally to the surface of the body. Although the goal was to show how flat surface locomotion was possible by using motion patterns like lateral undulation, this was visualized by letting three consecutive snake body segments be in contact with smooth rigid pegs, which has the same effect as friction forces in the lateral direction, and showing how potential energy in the axial muscles can be translated to kinetic energy, i.e. forward movement. This is in many ways analogues to obstacle-aided locomotion.

Consider the reaction forces from peg P_2 and P_3 in Fig 2.2 and the point of intersection p. It is evident that if the angles between the segments are not equal, that is $\alpha \neq \beta$, the reaction force from peg P_1 will not intersect at p and, given tension in the tensile element M_{r2}, the structure cannot be in

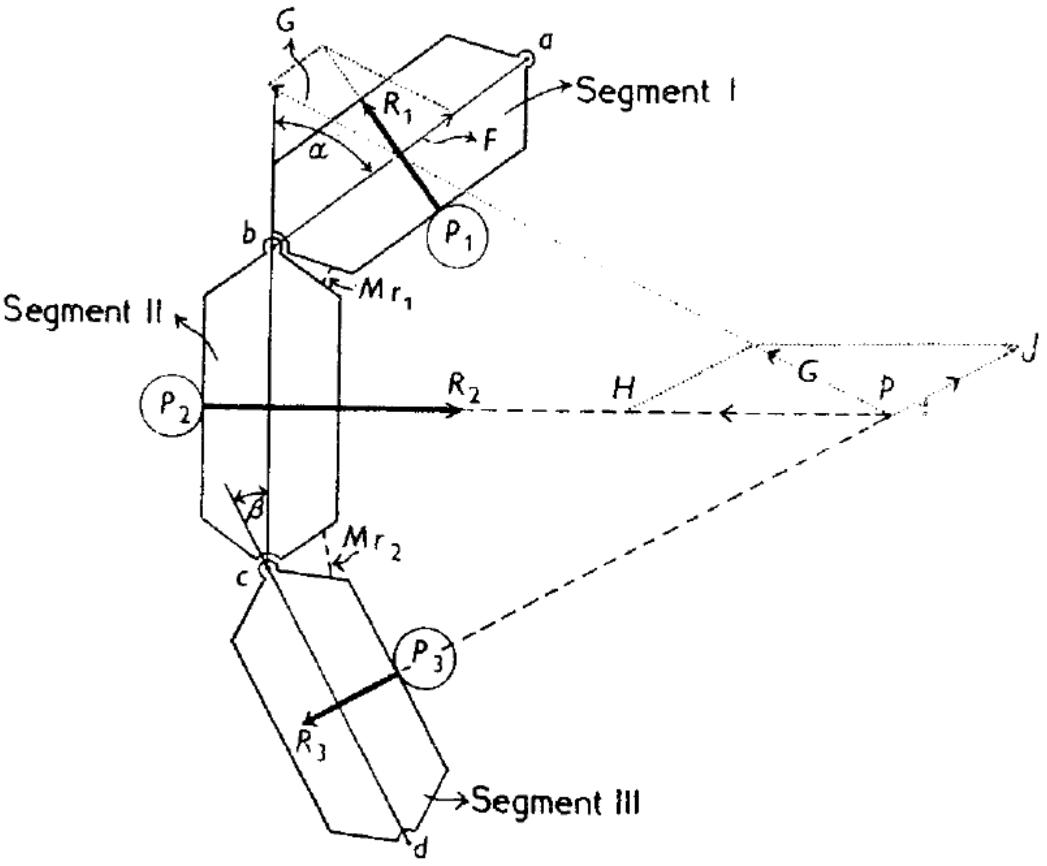


Figure 2.2: Three consecutive snake body segments in contact with three smooth rigid pegs. Contact forces R_3 and R_2 meet in the point P, but R_1 does not. Decomposing R_1 into two force components, one lying along the link axis, denoted F, and one aligned to meet in the point P, denoted G, shows the relation between the propulsive force F and the link configuration. Image source [14]

equilibrium. By formulating the force and torque balances, it can be shown that the propulsive force F in the direction of travel can be expressed as

$$F = \frac{T}{l}\frac{sin\alpha - sin\beta + sin(\alpha - \beta)}{1 + cos\alpha} \tag{2.1}$$

where T is the torque in M_{r2} and l is half the length of a snake body segment. Notice that if the angles become equal or zero there will be no propulsive force. If all the snake segment that constitutes the length of the snake are

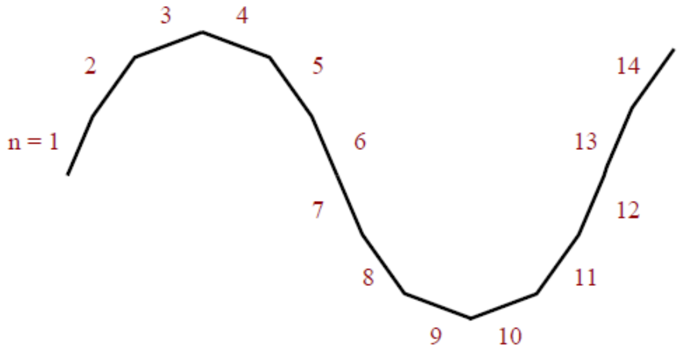


Figure 2.3: A snake robot with $n = 14$ links arranged to achieve a gliding form capable of forward propulsion. Image source [15]

arranged in such a matter it is possible to generate forward propulsion in the direction desired by manipulating the joint torques. The resulting motion is a wave-like pattern that closely resembles the movement of a real snake performing lateral undulation.

Building on the works by Gray, Hicks 2003 [15] showed the necessary conditions for locomotion via lateral undulation and provided a complete mathematical model to describe the gliding form presented by Gray.

Transeth et al. 2008a introduced and defined the term obstacle-aided locomotion. A hybrid mathematical model of the snake robot and its interaction with the environment was presented. As opposed to some other works, each link was modelled with both length and width and the areas at the end of each link were assumed to be shaped like semicircles. This closer resembles the shape of a physical snake robot link. The model included both contact forces from the obstacles as well as isotropic friction forces from the ground. Contact between the snake robot and an obstacles was described by a gap function, the distance between a link and an obstacle, and was modelled in a hybrid way, i.e. the snake is either fully in contact with the obstacle or it is not, as opposed to a spring-damper approach. In Transeth et al. 2008b [16] the framework was extended to a 3D snake robot model. Experiments where conducted using both simulations and a physical robot built as part of the research [17].

Bayraktaroglu and Blazevic 2005 [18] represented obstacles as spherical rigid bodies and investigated snake robot locomotion when the snake robot treated these obstacles as push points. Interaction between the snake robot and the rigid bodies was seen as a continuously sliding contact and was modelled through a spring-damper model due to the flexibility of a physical snake's skin. A physical robot was developed and constructed so that internal and pushing actions were decoupled by fitting lateral plates on each side of a robot link which were actuated separately from the joint actuators and were used to push against obstacles. Bayraktaroglu et al. 2006 [19] extends upon this work. A physical snake model was developed and a control algorithm for realizing lateral undulation in the presence of push points was presented. The lateral plates were no longer needed and instead the robot was fitted with sensors on each side, capable of detecting contact with a push point. Once a new push point has been detected by the head link an optimization process is commenced, fitting the shape of the snake robot between the first two push points after some criteria. The optimality of the shape is not considered.

Liljebäck 2009 extended a flat-surface 2D snake robot model to include an obstacle contact model [20]. The contacts were modelled as unilateral holonomic constraints, with the reasoning that an obstacle will prevent the snake robot from moving into the obstacle and will apply force in one direction, i.e. away from the obstacle. The task is to calculate the forces that satisfy this constraint and add to the equations of motion. The snake robot was controlled using a leader-follower scheme and a jam resolution scheme was developed that would examine the state of the snake robot and resolve potentially jammed links by rotating links so that the propulsive component of each link is increased. Liljebäck 2010a [21] presented a contact model that explicitly modelled the hybrid nature of the snake-obstacle interaction. This approach extended on [20] by calculating contact forces with respect to the normal direction of the robot links as opposed to the normal direction of the obstacles themselves, removing the necessity of explicitly considering the shape of the obstacle. Experiments were conducted with the physical snake robot presented in Liljebäck 2010b [22]. The wheel-less robot consisted of 10 links with a smooth hull and each link was fitted with contact sensors, enabling the snake robot to sense its exterior.

Chapter 3
A Static View on Obstacle-Aided Locomotion

In the following chapter we present a model of a snake robot during obstacle-aided locomotion. This model forms the basis for the latter discussions and simulations of obstacle-aided locomotion.

3.1 Model assumptions

The following assumptions have been made in the modelling of the snake robot

- The snake robot is assumed to be a continuous snake of finite length (i.e. with an infinite amount of discrete joints) in contact with a known, discrete set of obstacles.

- An obstacle is defined as an immovable and impassable object with a fixed position, but no extension, that is to say, point shaped.

- Furthermore it is assumed that there is no loss of energy due to ground fiction, friction along the obstacles or mechanical aspects in the joints of the snake.

- The extension in the transversal direction of the snake is insignificant, i.e. we control the position of each point along the snake so that it follows a given trajectory in space perfectly.

These assumptions result in a model that is somewhat removed from a realistic scenario, but it is hoped that each of these aspects can be conquered given time. We reserve a further discussion of this to a separate chapter and to future works.

3.2 Modelling of obstacle-aided locomotion

The modelling efforts are based on the formulation of a recursive Newton-Euler approach to modelling a robot manipulator as found in Spong 2006 [23] and adapted to include contact with obstacles. This has the added benefit that an already established modelling approach is applied to a field which has undergone comparatively little study. Consider Fig 3.1 which depicts a rigid link i, illustrated in yellow, with a single point of contact, illustrated in red, to the environment in addition to adjacent links.

Table 3.1 presents the mathematical symbols used to model the kinematics of the rigid link presented in Fig 3.1.

Assuming static conditions we can derive the following equations for force and torque balance:

$$f_i - R_i^{i+1} f_{i+1} + \kappa_i = 0 \tag{3.1}$$

$$\tau_i - R_i^{i+1} \tau_{i+1} + r_i \times (\kappa_i - R_i^{i+1} f_{i+1}) = 0 \tag{3.2}$$

[1] Note that according to our configuration, the force from link i on link $i+1$ would be f_{i+1} so this is the reaction force working in the opposite direction; this explains the negation. The force vector f_{i+1} is expressed in the coordinate frame of link $i+1$; this explains the rotation matrix R_i^{i+1}. The same reasoning applies to the torque.

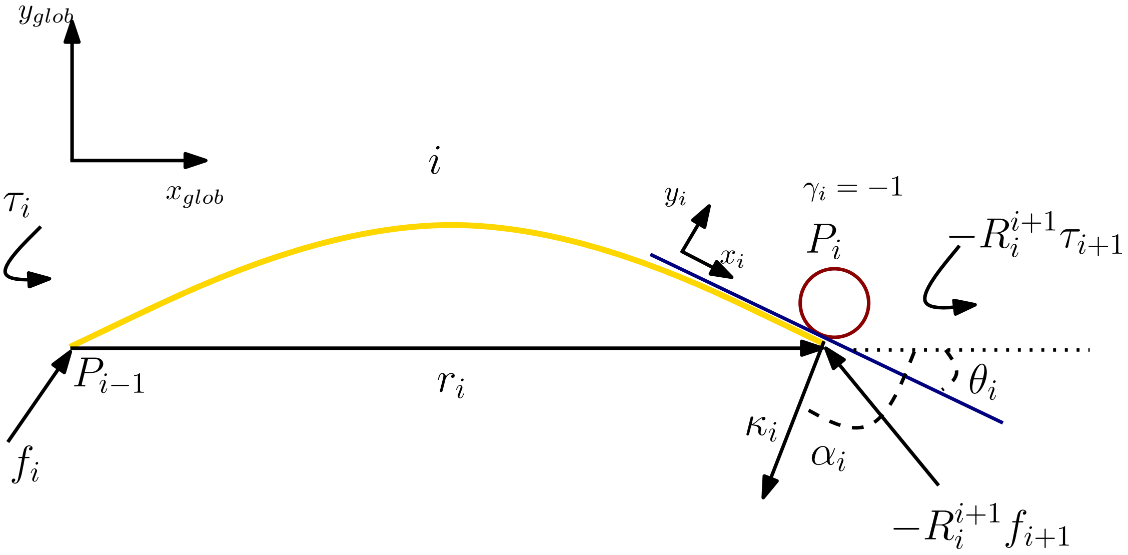


Figure 3.1: Illustration of the forces and torques acting on a rigid link i

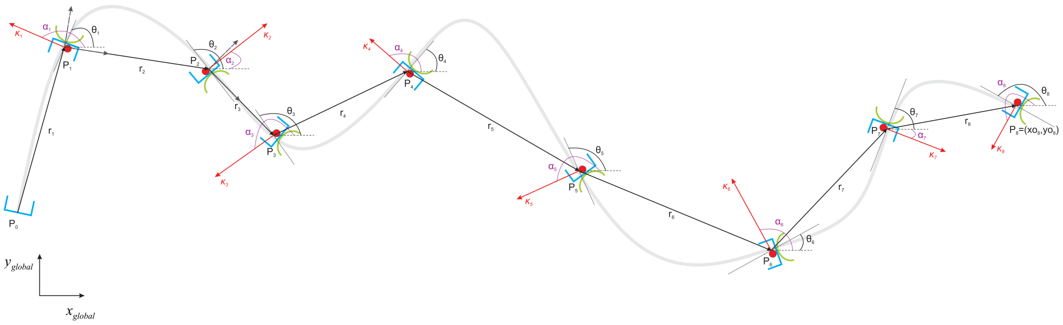


Figure 3.2: Continuous snake robot in contact with eight point shaped obstacles.

Consider now the full length of the snake robot as presented in Fig 3.2 where the robot is in contact with n point shaped obstacles $P_i\,, i \in [1, n]$. We can now divide the length of the robot into segments corresponding to the partition denoted link i in Fig 3.1. Thus, segment i is the segment from, *but not including*, obstacle P_{i-1} through, *and including*, obstacle P_i. Symbolically we write this as

$$segment_i = (P_{i-1}, P_i]$$

where (denotes an open interval and [denotes a closed interval. The excep-

Table 3.1: Parameters that characterize a rigid link i in contact with the environment and adjacent links.

Symbol	**Description**
P_i	Point of contact between link $i-1$ and link i.
P_{i+1}	Point of contact between link i and link $i+1$.
r_i	Vector from P_{i-1} to P_i
f_i	Force from link $i-1$ on link i.
τ_i	Torque from link $i-1$ on link i.
$-R_i^{i+1} f_{i+1}$	Force from link $i+1$ on link i. [1]
$-R_i^{i+1} \tau_{i+1}$	Torque from link $i+1$ on link i.
κ_i	The contact force from the obstacle on the link. Due to lack of friction between obstacle and snake robot this force is normal to the tangent of contact.
θ_i	Angle between tangent at P_i and global x-axis
γ_i	Contact parameter of link i. The value of the contact parameter is $\gamma_i = 1$ when the contact force points along the positive link y-axis y_i and $\gamma_i = -1$ when it points along the negative link y-axis. The value is $\gamma_i = 0$ when there is no contact.
α_i	Angle between contact force at P_I and global x-axis. The tangent angle θ_i and the contact angle α_i are always related through the equation $\alpha_i = \theta_i + \gamma_i \frac{\pi}{2}$.

tion to this partition is the ”tail segment”, $[P_0, P_1]$ which includes both the first obstacle and the end of the snake robot.

We can now attempt to derive the kinematics of the whole snake robot. Starting at the front of the robot we analyse the anterior segment, the ”head segment” (P_n and beyond), and recursively move backwards until we arrive at the posterior segment $[P_0, P_1]$ where we hope to find an expression for the force and torque balances of the whole robot. As the head segment is not affected by anything external to the robot it is straight forward to realize that it does not affect the analysis. Thus we start by applying equations

(3.1) and (3.2) on the segment $(P_{n-1}, P_n]$:

$$f_n - R_n^{n+1} f_{n+1} + \kappa_n = 0$$

$$\tau_n - R_n^{n+1} \tau_{n+1} + r_n \times (\kappa_n - R_n^{n+1} f_{n+1}) = 0$$

which, because there are no external forces f_{n+1}, simplifies to an expression where the torques and forces acting between the segments P_{n-2}, P_{n-1} and P_{n-1}, P_n is only determined by the contact force κ_n at P_n

$$f_n + \kappa_n = 0$$

$$\tau_n + r_n \times \kappa_n = 0$$

We now have an expression for f_n which we can use in the next equation as we move recursively backwards along the snake robot. The posterior segment $[P_0, P_1]$ is subject to the same process except, instead of a force f_1 from a posterior segment, we imagine a virtual string σ fastened to the rear of the snake, purposely hindering the snake from advancing, thus ensuring static conditions. This is illustrated in Fig 3.2 where the string is fastened to the fixed point P_f with the sting tensile force f_σ acting between P_f and the rear of the snake. For $[P_0, P_1]$ we get the following force and torque balance:

$$f_\sigma - R_1^2 f_2 + \kappa_1 = 0$$

$$- R_1^2 \tau_2 + r_1 \times (\kappa_1 - R_1^2 f_2) = 0$$

By recursively inserting into the forces balances for all the segments we end up with an expression of f_σ as a function of the set of contact forces $\{\kappa_i\}, i \in [1, n]$

$$\begin{aligned} f_\sigma &= f(\{\kappa_i\}) \\ &= -(\kappa_1 + R_1^2(\kappa_2 + R_2^3(... + R_{n-1}^n \kappa_n))) \end{aligned} \tag{3.3}$$

The same approach of recursive insertion apply to the torques.

$$\begin{aligned}
\tau_n &= -r_n \times \kappa_n \\
\tau_{n-1} &= -(R_{n-1}^n(r_n \times \kappa_n) + r_{n-1} \times (\kappa_{n-1} + R_{n-1}^n \kappa_n) \\
&\vdots \\
\tau_1 &= g(\kappa_i) \qquad (3.4)
\end{aligned}$$

giving, for torque number i,

$$\tau_i = -\sum_{k=i}^{n}\sum_{j=k}^{n}(R_1^k r_k) \times (R_1^j \kappa_j) \qquad (3.5)$$

The string σ and the string tensile force f_σ are constructs only used for modelling purposes ensuring that we have static conditions. The string tensile force is the force with which the snake robot acts upon the environment. If we imagine that the string was cut, and the snake robot follows the planned trajectory without drifting sideways from the path, the snake robot would traverse the given trajectory with an acceleration equal to

$$a_\sigma = f_\sigma / m \qquad (3.6)$$

where m is the total mass of the robot. If we could control f_σ, the snake robot's equation of motion would simplify to this scalar expression.

Equation (3.3) describes how f_σ can be found as a function of the set of contact forces $\{\kappa_i\}$ and equation (3.5) how the joint torques τ_i must be chosen to realize a given set of contact forces. A possible strategy for controlling the snake could thus be:

1. Specify desired forward propulsion f_σ.

2. Find an optimal set of contact forces $\{\kappa_i\}$ that realizes given f_σ.

3. Calculate and apply torques around obstacles τ_i as a function of the set of contact forces $\{\kappa_i\}$.

4. Find a mapping that maps the torques around obstacles to joint torques for a snake with N discrete joints.

In addition, as we have mentioned, we must enforce some kind of position control on the snake robot so that it follows a given trajectory. This trajectory must be chosen so that it interacts with the set of obstacles $\{P_i\}$ in the tangents specified by the tangent angle vector $\boldsymbol{\theta_i}$

Chapter 4

Control of Obstacle-Aided Locomotion

In the following chapter we present the control strategy for obstacle-aided locomotion. The first section contains an overview of control allocation and the specific optimization problems related to snake robotics. The following sections contains the rest of the controller scheme.

4.0.1 Control Objective

Fig 4.1 illustrates a spring tightly wound around a series of obstacles. Imagine pulling at a point on the spring so that it moves away from an obstacle and then letting go; the spring will snap back into position because of the

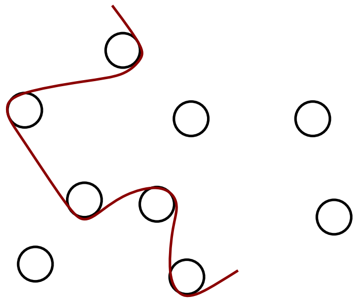


Figure 4.1: A spring tightly wound around obstacles. An illustration of the idea behind the controller scheme.

forces keeping it in place. Imagine that we could change the spring coefficient at a point to create a moment around one obstacle. This would force the whole spring to move to one side, along the spring's path. If we could control the snake robot to act like a spring with adjustable spring coefficients this would be a good illustration of how it would work.

The idea of the controller is to utilize obstacles as push points for propulsion. It is assumed that we have specified some path which makes the snake robot impact several obstacles. The objective of the controller is then to, at each time instance, find an input vector of joint torques that enables the snake robot to achieve propulsive force from the obstacles while tracking the path. The simplified static representation of a snake robot enables us to consider optimality of a contact force scenario and make sound decisions about which obstacles to utilize. One challenge lies in mapping the static scenario to the dynamic snake robot.

4.1 Control Allocation

The Control Allocation Problem is the problem of distributing a total desired control force among a set of redundant actuators [24]. It has been widely used in marine sectors, but the following discussion will argue for its relevance in snake robotics. For a system of n Degrees of Freedom (DOF) the generalized control forces $\boldsymbol{\tau_\sigma} \in \mathbb{R}^n$ must be distributed among the control inputs $\boldsymbol{u} \in \mathbb{R}^r$. Whenever $r > n$ the problem is referred to as an overactuated control problem, whereas $r < n$ results in an underactuated control problem. Control allocation is a method used on overactuated systems to find the distribution of control force inputs that is optimal in respect to some optimality criteria and has been used on marine vessels [25] and air planes [26]. The following introduction to control allocation will be based on Chapter 7.5 of Fossen 2002 [25].

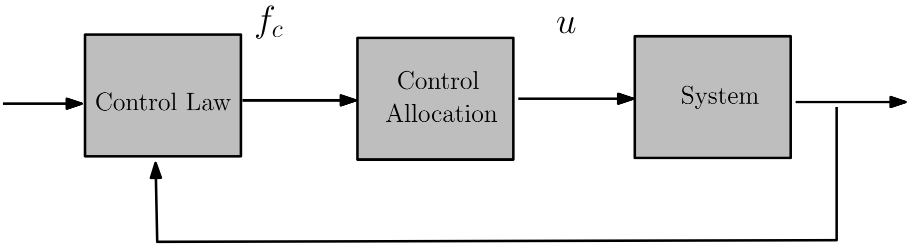


Figure 4.2: Block diagram of a feedback system with control allocation

4.1.1 Introduction to Control Allocation

A marine vessel may have a number of different actuators e.g. propellers, thrusters and fins. These typically vary in size, angle and energy cost. The forces and moments for a 6 DOF vessel can be written in terms of the force vector $\boldsymbol{f} = [F_x, F_y, F_z]^T$ as

$$\boldsymbol{\tau_\sigma} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{r} \times \boldsymbol{f} \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ F_z \\ F_z l_y - F_y l_z \\ F_x l_z - F_z l_x \\ F_y l_x - F_x l_y \end{bmatrix} \tag{4.1}$$

where $\boldsymbol{r} = [l_x, l_t, l_z]^T$ are the moment arms. This can be written more compactly as

$$\boldsymbol{\tau_\sigma} = \boldsymbol{T(\alpha)Ku} \tag{4.2}$$

where $\boldsymbol{u} \in \mathbb{R}^r$ and $\boldsymbol{\alpha} \in \mathbb{R}^p$ are control input vectors and $\boldsymbol{f} = \boldsymbol{Ku} \in \mathbb{R}^r$ is a vector of control forces. The matrix $\boldsymbol{K} \in \mathbb{R}^{r\times r}$, called the force coefficient matrix, is a diagonal matrix that weighs the different control inputs. The actuator configuration matrix $\boldsymbol{T(\alpha)} \in \mathbb{R}^{n\times r}$ is defined as a set of column vectors $\boldsymbol{t_i} \in \mathbb{R}^n$ which differs depending on the type of actuator. In 4 DOF, the following examples show the column vectors of the azimuth thruster and the main propeller respectively:

$$\boldsymbol{t}_{azimuth} = \begin{bmatrix} cos\alpha_i \\ sin\alpha_i \\ l_{yi}cos\alpha_i - l_{zi}sin\alpha_i \\ l_{xi}sin\alpha_i - l_{yi}cos\alpha_i \end{bmatrix}, \boldsymbol{t}_{main} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ l_{yi} \end{bmatrix}$$

In other words, the actuator configuration matrix $\boldsymbol{T(\alpha)}$ contains information on the geometric orientation of the actuator in regards to the vessel.

4.1.2 Unconstrained Control Allocation with Nonrotatable Actuators

The easiest control allocation problem is one where all actuators are unconstrained, that is, without bounds on the forces they can produce or the angles they can achieve, nor the respective time derivatives. It is also assumed that the actuators are not rotatable, i.e. with fixed orientation giving the following actuator configuration matrix $\boldsymbol{T} = \boldsymbol{T(\alpha_0)}$ where α_0 is a constant. To find an optimal distribution of control forces f we define the following unconstrained least-squares problem [27]

$$\begin{aligned} J = \min_{\boldsymbol{f}} \quad & \{\boldsymbol{f}^T\boldsymbol{W}\boldsymbol{f}\} \\ \text{s.t.} \quad & \boldsymbol{\tau_\sigma} - \boldsymbol{T}\boldsymbol{f} = \boldsymbol{0} \end{aligned} \tag{4.3}$$

where $\boldsymbol{W}$ is a positive definite weight matrix. It can be shown, the details are found in [25], that the optimal solution can be found as

$$\boldsymbol{u} = \boldsymbol{K}^{-1}\boldsymbol{T}_\omega\boldsymbol{\tau_\sigma} \tag{4.4}$$

with $\boldsymbol{T}_\omega = \boldsymbol{W}^{-1}\boldsymbol{T}^T(\boldsymbol{T}\boldsymbol{W}^{-1}\boldsymbol{T}^T)^{-1}$. Note that $(\boldsymbol{T}\boldsymbol{W}^{-1}\boldsymbol{T}^T)$ is here assumed to be nonsingular.

4.1.3 Constrained Control Allocation with Rotatable Actuators

Because of safety and energy reasons there will in general be some constraints on the actuators. On a vessel with azimuth thrusters we can also not assume that the actuator configuration is fixed, it would void the whole concept of azimuth thrusters. For $\boldsymbol{\alpha}$ not constant, the primary constraint is:

$$\boldsymbol{\tau_\sigma} = \boldsymbol{T(\alpha)f} \tag{4.5}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^p$ represents the azimuth angles. Typically, the thrusters operate in some sectors $\alpha_{i,min} \leq \alpha_i \leq \alpha_{i,max}$ with a limited turning rate $\dot{\boldsymbol{\alpha}}$. The actuator configuration matrix changes and has to be computed at each time sample. Where previously, it was possible to pre-check or pre-design $\boldsymbol{T(\alpha)}$ to be well behaved, we can no longer assume that $(\boldsymbol{T(\alpha)W}^{-1}\boldsymbol{T(\alpha)}^T)$ is nonsingular. Such a singularity means that no force can be produced in certain directions and may be derogatory to the performance of the system. The optimization problem is therefore adjusted to reflect the changes to the problem:

$$\begin{aligned}
J = \min_{\boldsymbol{f},\boldsymbol{\alpha},\boldsymbol{s}} & \left\{ \sum_{i=1}^{r} \bar{P}_i |f_i|^{3/2} + \boldsymbol{s}^T\boldsymbol{Qs} + (\boldsymbol{\alpha} - \boldsymbol{\alpha_0})^T\boldsymbol{\Omega}(\boldsymbol{\alpha} - \boldsymbol{\alpha_0}) \right. \\
& \left. + \frac{\rho}{\epsilon + det(\boldsymbol{T(\alpha)W}^{-1}\boldsymbol{T}^T\boldsymbol{(\alpha)}} \right\} \\
s.t. & \\
& \boldsymbol{T(\alpha)f} = \boldsymbol{\tau_\sigma} + \boldsymbol{s} \\
& \boldsymbol{f}_{min} \leq \boldsymbol{f} \leq \boldsymbol{f}_{max} \\
& \boldsymbol{\alpha}_{min} \leq \boldsymbol{\alpha} \leq \boldsymbol{\alpha}_{max} \\
& \Delta\boldsymbol{\alpha}_{min} \leq \boldsymbol{\alpha} - \boldsymbol{\alpha_0} \leq \Delta\boldsymbol{\alpha}_{max}
\end{aligned} \tag{4.6}$$

where

- $\sum_{i=1}^{r} \bar{P}_i |f_i|^{3/2}$ represents power consumption with $\bar{P}_i > 0$ being positive weights.

- s is the error between the desired total force and the generated total force, penalized by the term $\boldsymbol{s}^T\boldsymbol{Q}\boldsymbol{s}$. Large $\boldsymbol{Q}$ leads to a solution close to $\boldsymbol{s} \approx 0$ whenever possible.

- The constraints ensures that the forces, angles and angle rates is kept within a feasible region.

- The term
$$\frac{\rho}{\epsilon + det(\boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{W}^{-1}\boldsymbol{T}^T(\boldsymbol{\alpha})}$$
is added to avoid singular angle configurations ($det(\boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{W}^{-1}\boldsymbol{T}^T(\boldsymbol{\alpha}) = 0$. If $\epsilon \geq 0$ is chosen small this term will penalize singular configurations, i.e. it is large for these values of $\boldsymbol{\alpha}$. $\rho \geq 0$ is a weight for which a high value gives high manoeuvrability, but high power consumption.

The constrained control allocation optimization problem presented in (4.6) is a non-convex problem that is generally difficult to solve. Non-convex problems are characterized by having potentially a number of local optima, making it difficult to find the global optima and we cannot generally guarantee finding a feasible solution in a given amount of time [28]. For systems with a large update frequency, it would require a large amount of computation to run a non-convex optimization problem at every time step and Fossen 2002 [25] presents quadratic and linear approximations to the optimization problem.

4.2 Finding an optimal set of contact forces

The following section contains a discussion on finding the optimal set of contact forces that realizes a desired propulsion of the center of mass. The discussion is based on the static model that was presented in 3.2 and uses control allocation as an approach to find the optimal contact forces.

Control allocation has been widely used on applications like marine vessels and air planes, but less so on robotic applications. Many robotic appliances are underactuated, which makes control allocation a pointless pursuit. Indeed, the dynamic snake robot models presented in [20], [16] are underactuated systems. However, simplifying the snake robot model to a static case allows us to treat the snake robot in a way that is quite analogous to a ship with azimuth thrusters.

Consider again equation (3.3), $f_\sigma = f(\{\kappa_i\})$, which gives the propulsive force as a function of the obstacle forces. Each contact force is a function of its magnitude and its direction, i.e. $\kappa_i = |\kappa_i|\angle\kappa_i = |\kappa_i|\angle\alpha_i$. If we expand and express the result in the global x-y plane we can observe that finding $f_\sigma = f(\{\kappa_i\})$ is akin to a control allocation problem with a 3 DOF representation, where in addition to the total force in x- and y-direction, the moment around the z-axis is also considered. The desired propulsion vector will be denoted

$$\boldsymbol{\tau_\sigma} = \begin{bmatrix} \boldsymbol{f_\sigma} \\ \sum \tau_{com} \end{bmatrix}$$

As we are operating in 3 DOF the actuator configuration matrix $\boldsymbol{T(\alpha)} \in \mathbb{R}^3$ is quite simple, where column number i can be written as:

$$t_{obstacle,i} = t_i = \begin{bmatrix} cos\alpha_i \\ sin\alpha_i \\ l_{x,i}sin\alpha_i - l_{y,i}cos\alpha_i \end{bmatrix} \tag{4.7}$$

where $l_i = [l_{x,i}, l_{y,i}]^T$ is the moment arm. With each column in $\boldsymbol{T(\alpha)}$ given by equation (4.7) we derive the following:

$$\boldsymbol{\tau_\sigma} = \boldsymbol{T(\alpha)}|\kappa_i| \tag{4.8}$$

It is now possible to solve the minimization problem given in (4.6) to find a set of contact forces that realizes a desired $\boldsymbol{\tau_\sigma}$.

For a snake robot following a pre-planned trajectory where each tangent of contact is specified, constrained nonrotatable control allocation is relevant. Finding the optimal propulsive vector could be formulated as the solution that minimizes the error $\boldsymbol{\tau_\sigma} - \boldsymbol{T}\boldsymbol{f}$ while also minimizing the energy expenditure. The optimization problem then becomes

$$\begin{aligned} J &= \min_{\boldsymbol{f}} \Big\{ \boldsymbol{f}^T \boldsymbol{W} \boldsymbol{f} + \boldsymbol{s}^T \boldsymbol{Q} \boldsymbol{s} \Big\} \\ &\text{s.t.} \\ &\boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{f} = \boldsymbol{\tau_\sigma} + \boldsymbol{s} \\ &\boldsymbol{f}_{min} \leq \boldsymbol{f} \leq \boldsymbol{f}_{max} \end{aligned} \tag{4.9}$$

An alternative optimality criteria could be to minimize the variance of amplitude of the contact forces. If only the energy efficiency is taken into account an optimal solution would likely involve as few obstacles as possible, to limit the amount of power used to generate obstacle forces. However, a spring wrapped around several obstacles is more stable (meaning it won't slip away from the obstacles) than if it was just in contact with two. Minimizing the variance of the contact forces will ensure that the snake robot utilizes as many contacted obstacles as possible. We can write an expression for the variance as:

$$\sum_{i=1}^{n-1} \kappa_{i+1} - \kappa_i$$

Adding this term to the cost function gives us the following optimization problem

$$\begin{aligned} J &= \min_{\boldsymbol{f}} \Big\{ \boldsymbol{f}^T \boldsymbol{W} \boldsymbol{f} + p \sum_{i=1}^{n-1} |\kappa_{i+1} - \kappa_i| + \boldsymbol{s}^T \boldsymbol{Q} \boldsymbol{s} \Big\} \\ &\text{s.t.} \\ &\boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{f} = \boldsymbol{\tau_\sigma} + \boldsymbol{s} \\ &\boldsymbol{f}_{min} \leq \boldsymbol{f} \leq \boldsymbol{f}_{max} \end{aligned} \tag{4.10}$$

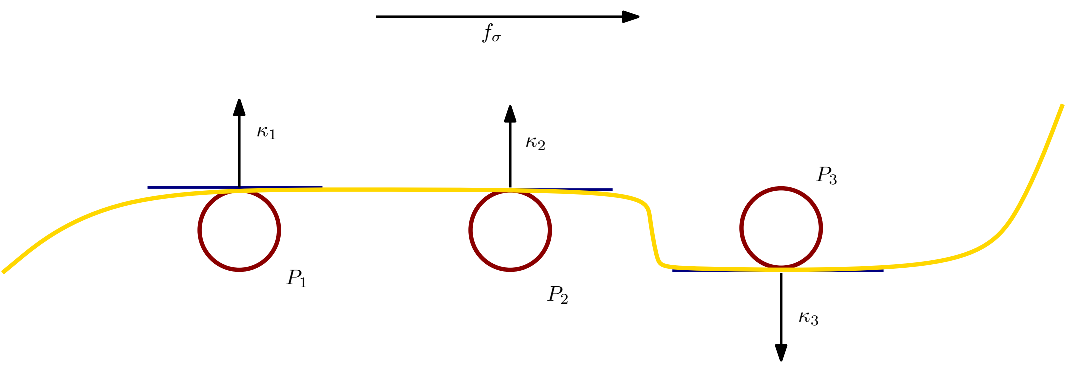


Figure 4.3: An example of a configuration that is unable to produce the desired propulsive force

where p is a scalar weight and the energy efficiency term is still included.

Here, it is assumed that the snake robot follows a planned path and is not allowed to change the tangent of contact with an obstacle. The control variable is therefore only the magnitude of the contact force. This will make some configurations unable to produce a desired propulsive vector, an example illustrated in Fig 4.3. By examining $\boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{T}(\boldsymbol{\alpha})^T$ it is possible to predict if this is the case, as it will be singular if the given configuration is not able produce force in certain directions. When planning the path of a snake robot, care should be taken to avoid such singular configurations. A strategy for path planning in an unstructured environment is however not within the scope of this thesis.

Allowing small changes in the tangents of contact is a way to compensate for a sub-optimal path, providing flexibility and possibly efficiency. It is possible to imagine a scenario where a new obstacle is reached and some local optimization around the old obstacles is beneficial. The necessary augmentations to (4.9) to allow for small changes in $\boldsymbol{\alpha}$ is given by (4.6) and is repeated here for consistency

$$\begin{aligned}
J = \min_{\boldsymbol{f},\boldsymbol{\alpha},\boldsymbol{s}} \Bigg\{ & \boldsymbol{f}^T \boldsymbol{W} \boldsymbol{f} + \boldsymbol{s}^T \boldsymbol{Q} \boldsymbol{s} + (\boldsymbol{\alpha} - \boldsymbol{\alpha_0})^T \boldsymbol{\Omega} (\boldsymbol{\alpha} - \boldsymbol{\alpha_0}) \\
& + \frac{\rho}{\epsilon + det(\boldsymbol{T(\alpha)}\boldsymbol{T}^T\boldsymbol{(\alpha)}} \Bigg\} \\
s.t. & \\
& \boldsymbol{T(\alpha)f} = \boldsymbol{\tau_c} + \boldsymbol{s} \\
& \boldsymbol{f}_{min} \leq \boldsymbol{f} \leq \boldsymbol{f}_{max} \\
& \boldsymbol{\alpha}_{min} \leq \boldsymbol{\alpha} \leq \boldsymbol{\alpha}_{max} \\
& \Delta\boldsymbol{\alpha}_{min} \leq \boldsymbol{\alpha} - \boldsymbol{\alpha_0} \leq \Delta\boldsymbol{\alpha}_{max}
\end{aligned} \tag{4.11}$$

Recall that the last term of the cost function was added to prevent singular configurations.

4.3 Torque Distribution

In the following section we describe a possible method for mapping the torques of the static case, where each snake segment is defined as detailed in Section 3.2, over to a snake robot with N discrete joints, each of length l.

Solving the control allocation problem for a desired propulsive vector τ_σ gives a set of contact forces $\boldsymbol{\kappa}$. The static model presented in Section 3.2 gives us a relation between contact forces and torques, as described in equation (3.5). These torques can be viewed as the torques from one snake segment to another, or the torques around an obstacle. For a snake robot with N discrete joints, we are however, interested in knowing the joint torques. Fig 4.4 shows an example scenario.

Starting from the head of the snake, let d_i be the distance from joint number i to the first obstacle in front of this joint, as illustrated in Fig 4.5. As we progress backwards along the snake, this is done for each joint. This

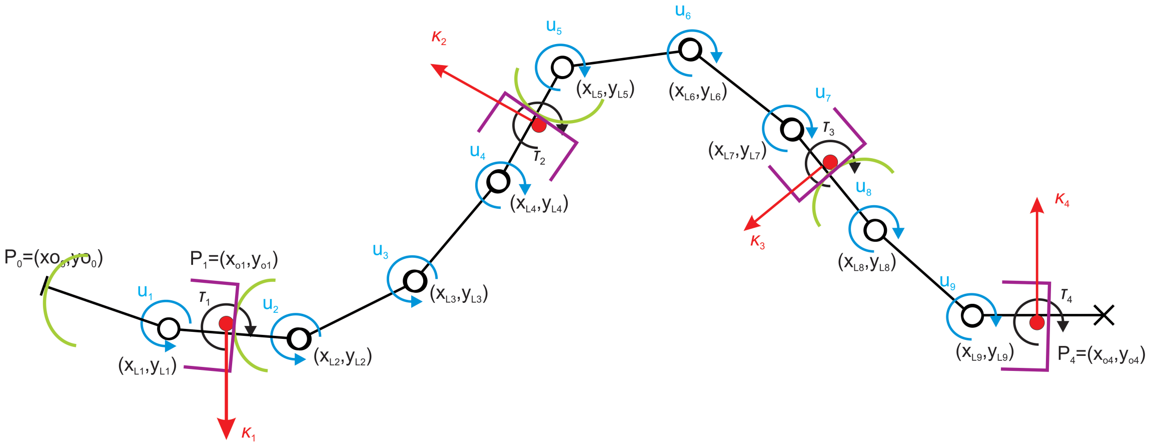


Figure 4.4: A snake robot with 10 links in contact with four obstacles. The moments around each obstacle T_i should be mapped to joint torques u_i

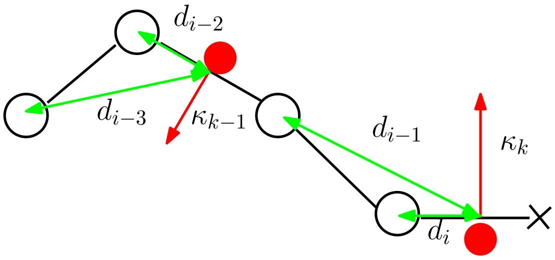


Figure 4.5: d_i is the distance between joint number i and the first obstacle in front of the joint

gives the following relation between the moment around obstacle k and the joint torque at joint i

$$u_i = \tau_k + f_k \times d_i \tag{4.12}$$

A possible way to implement this is to iterate along the snake, keeping track of the latest obstacle and calculate the distance from each joint to the current latest obstacle.

4.4 Controller Scheme

In the following section we present a potential controller scheme for the snake robot. We assume that the snake robot is described by a dynamical model in an obstacle environment

4.4.1 Controller implementation

A desired propulsive vector τ_σ is calculated, based on the given path of the snake. Using Control Allocation, a set of contact vectors $\boldsymbol{\kappa}$ is calculated, based on some optimality criteria. By using the static representation of a snake robot presented in (3.3) and (3.4) the moments around each obstacle is calculated. With the mapping from moment around an obstacle to joint torques given by (4.12), the input vector $\boldsymbol{u}$ is calculated and applied to the system. In the assumptions for the static model we assume that the extension in the transversal direction of the snake is insignificant. That is, if an obstacle lie along the path that we choose for the snake robot, we must have some kind of position control that ensures a gliding motion along the obstacle. The controller is presented on a block diagram form in Fig 4.6

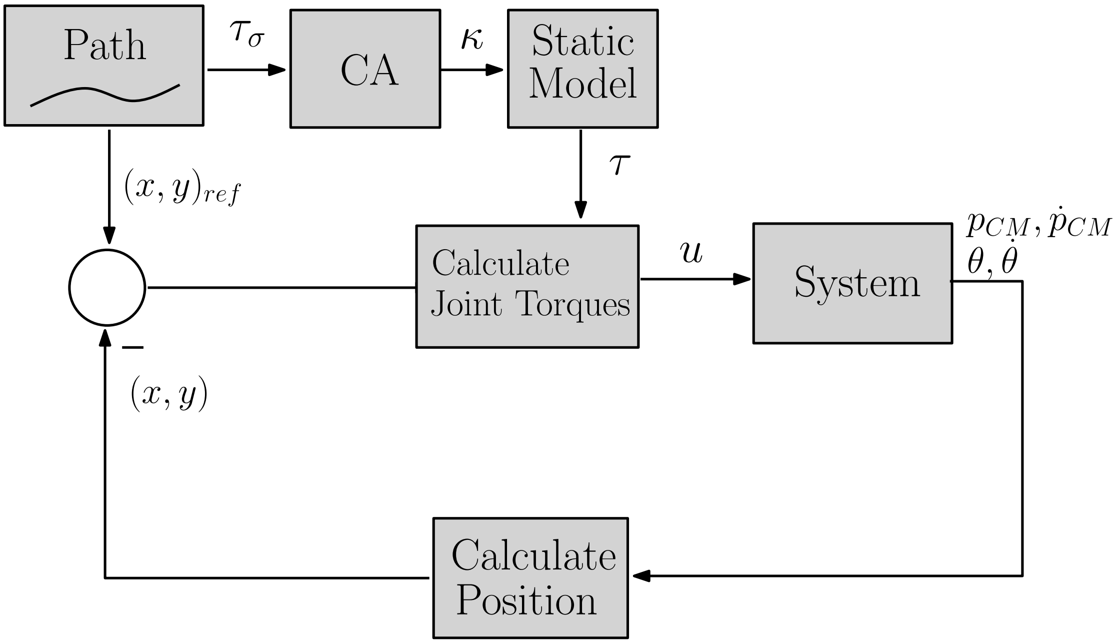


Figure 4.6: Block diagram of the controller scheme

Chapter 5 Simulation results

5.1 Calculating the optimal set of contact forces

In the following is a presentation of the calculation results of the discussion regarding how to find an optimal set of contact forces. Calculations are shown for the scenarios of both a constant tangent angle and a non-constant tangent angle. Note that constant tangent angle implies a prespecified tangent angle whereas a non-constant tangent angle implies that we allow the tangent angle to deviate from what is prespecified.

The calculations was done on a computer running Windows 7 and Matlab R2010b and are based on the model and nomenclature described in Section 3.2. The optimization problem is solved using the Matlab function *fmincon* running the sqp solver. As initialization, a set of obstacles $\boldsymbol{P} = [\boldsymbol{P_x}, \boldsymbol{P_y}]^T$, an initial set of tangent angles $\boldsymbol{\theta} = [\alpha_1, \alpha_2, ..., \alpha_n]^T$ and corresponding contact parameters $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, ..., \gamma_n]^T$ was given.

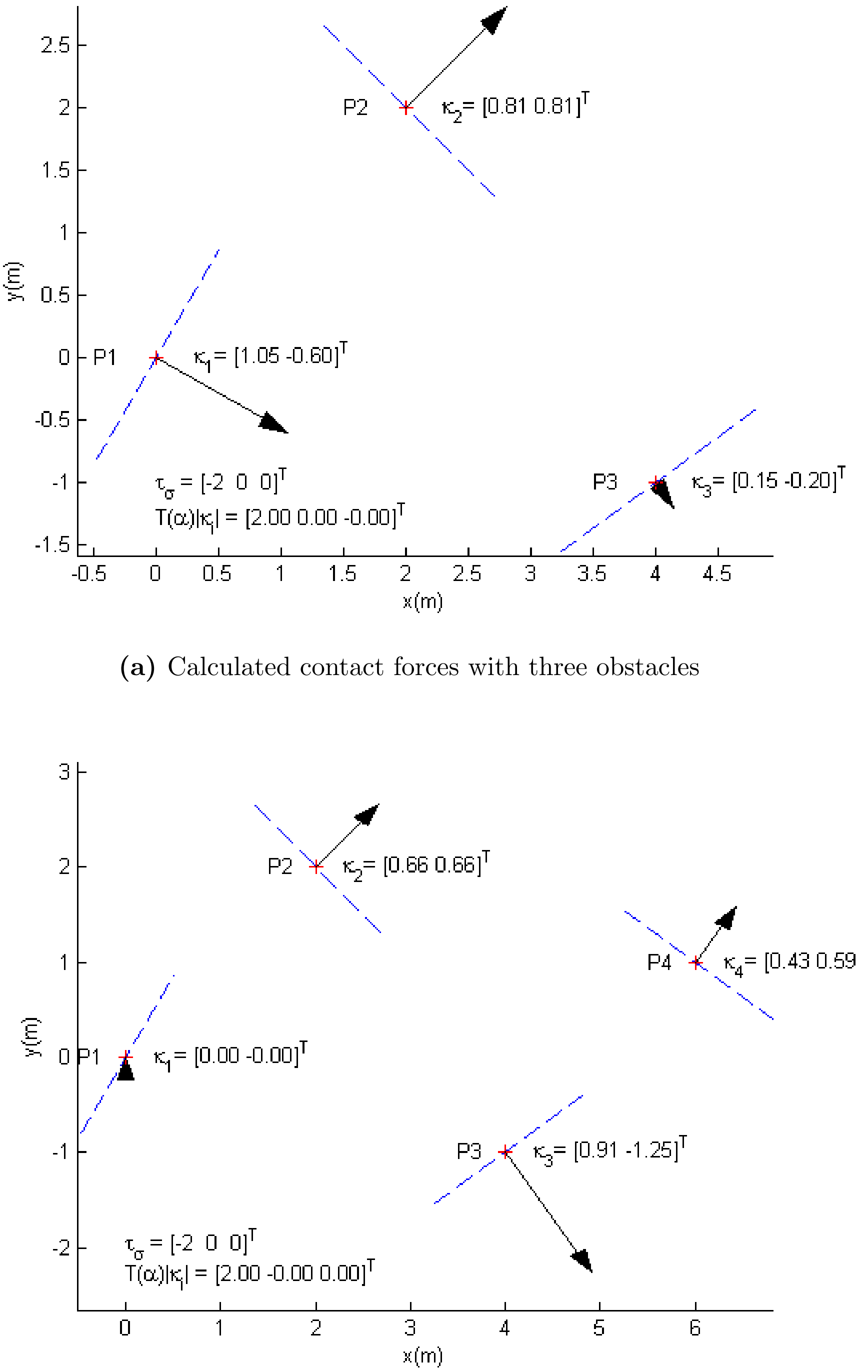
5.1.1 Energy efficiency with prespecified tangent angles

Finding the optimal set of contact forces for the case of prespecified tangent angles was solved by optimizing (4.9) with $n = 3$ and $n = 4$ obstacles. The desired propulsion vector was the same in both cases and, excluding the last obstacle, the same is true for the parameters related to obstacles. The following parameters was given:

$$\begin{aligned}
\boldsymbol{P} = \begin{bmatrix} P_1 & ... & P_4 \end{bmatrix} &= \begin{bmatrix} 0 & 2 & 4 & 6 \\ 0 & 2 & -1 & 1 \end{bmatrix} \\
\boldsymbol{\theta} &= \begin{bmatrix} -30^\circ & 45^\circ & -54^\circ & 54^\circ \end{bmatrix}^T \\
\boldsymbol{\gamma} &= \begin{bmatrix} -1 & 1 & -1 & 1 \end{bmatrix}^T \\
\boldsymbol{\tau_\sigma} &= \begin{bmatrix} -2 & 0 & 0 \end{bmatrix}^T
\end{aligned}$$

The results of the calculations are presented in Fig 5.1 and it can be seen that the calculated propulsion vector matches the desired propulsion vector very well. This implies that the center of mass should experience an acceleration of $2m/s$ in positive x-direction, $0m/s$ in y-direction and that the total angular momentum is 0. It is interesting to note that the optimal set changes when a new obstacle is encountered. The most valued contact force in Fig a is κ_1, but in Fig b it is zero.

Simulations was also done with $n = 5$ obstacles to show how the difference in chosen tangent angles (though still constant tangent angles) affects the outcome of the calculations. The parameters given was



(a) Calculated contact forces with three obstacles

(b) Calculated contact forces with four obstacles

Figure 5.1: Optimal set of contact forces for three and four obstacles respectively. Each obstacle is marked by a red cross. The tangent of contact is marked by a blue, dashed line. The force at each point is marked by the arrow, the direction of the arrow indicating the direction of the force and the length indicating the magnitude. τ_σ is the desired propulsion vector while $T(\alpha)|\kappa_i|$ is the calculated propulsion vector.

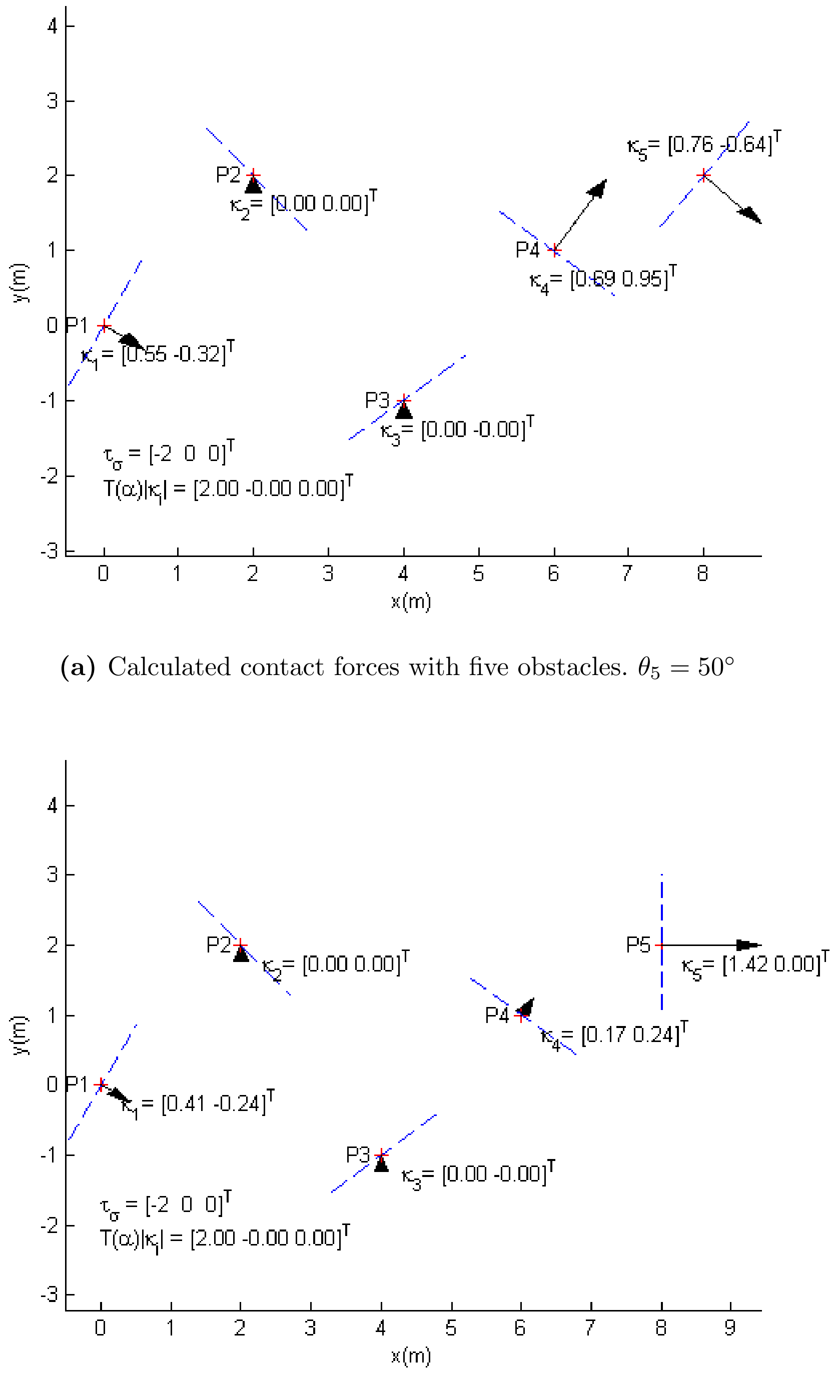
$$\begin{aligned}
\boldsymbol{P} = \begin{bmatrix} P_1 & ... & P_4 \end{bmatrix} &= \begin{bmatrix} 0 & 2 & 4 & 6 & 8 \\ 0 & 2 & -1 & 1 & 2 \end{bmatrix} \\
\boldsymbol{\theta} &= \begin{bmatrix} -30^\circ & 45^\circ & -54^\circ & 54^\circ & x \end{bmatrix}^T \\
\boldsymbol{\gamma} &= \begin{bmatrix} -1 & 1 & -1 & 1 & -1 \end{bmatrix}^T \\
\boldsymbol{\tau_\sigma} &= \begin{bmatrix} -2 & 0 & 0 \end{bmatrix}^T
\end{aligned}$$

The results are presented in Fig 5.2. For the given propulsive vector, $\theta_5 = 90^\circ$ is very beneficial and thus, obstacle five is very favoured, as can be seen in Fig 5.2b. Of interest is also the fact that only three obstacles are ever chosen. Because the optimization considers the energy efficiency of a given set of contact forces, the best solution is to use as few obstacles as possible.

5.1.2 Energy efficiency with non-constant tangent angles

Finding the optimal set of contact forces for the case of non-constant tangent angles was solved by optimizing (4.11) with $n = 4$ obstacles. The following parameters was given where instead of a set of fixed tangent angles $\boldsymbol{\theta}$, we supply a vector of initial values subject to appropriate constraints:

$$\begin{aligned}
\boldsymbol{P} = \begin{bmatrix} P_1 & ... & P_4 \end{bmatrix} &= \begin{bmatrix} 0 & 2 & 4 & 6 \\ 0 & 2 & -1 & 1 \end{bmatrix} \\
\boldsymbol{\theta_0} &= \begin{bmatrix} -30^\circ & 45^\circ & -54^\circ & 54^\circ \end{bmatrix}^T \\
\boldsymbol{\delta} &= x \begin{bmatrix} 1 & ... & 1 \end{bmatrix}^T \\
\boldsymbol{\gamma} &= \begin{bmatrix} -1 & 1 & -1 & 1 \end{bmatrix}^T \\
\boldsymbol{\tau_\sigma} &= \begin{bmatrix} -2 & 0 & 0 \end{bmatrix}^T
\end{aligned}$$



(a) Calculated contact forces with five obstacles. $\theta_5 = 50°$

(b) Calculated contact forces with five obstacles. $\theta_5 = 90°$

Figure 5.2: Optimal set of contact forces for five obstacles. Each obstacle is marked by a red cross. The tangent of contact is marked by a blue, dashed line. The force at each point is marked by the arrow, the direction of the arrow indicating the direction of the force and the length indicating the magnitude. τ_σ is the desired propulsion vector while $T(\alpha)|\kappa_i|$ is the calculated propulsion vector.

where we constrain the force angles $\boldsymbol{\alpha}$ by the following relation

$$\boldsymbol{\alpha}_{min} = \boldsymbol{\alpha} - \boldsymbol{\delta} \leq \boldsymbol{\alpha} \leq \boldsymbol{\alpha} + \boldsymbol{\delta} = \boldsymbol{\alpha}_{max} \tag{5.1}$$

Fig 5.3 shows the results with $\boldsymbol{\delta} = 5$ and $\boldsymbol{\delta} = 15$. It is clearly beneficial to use as few obstacles as possible to achieve the desired propulsion vector. In Fig 5.3a, the obstacle P_3 contributes somewhat while in Fig 5.3b the effect is barely noticeable. The forces at P_1 and P_2 have been angled so that they are able to realize the desired force while still maintaining moment balance.

5.1.3 Minimize force variance with prespecified tangent angles

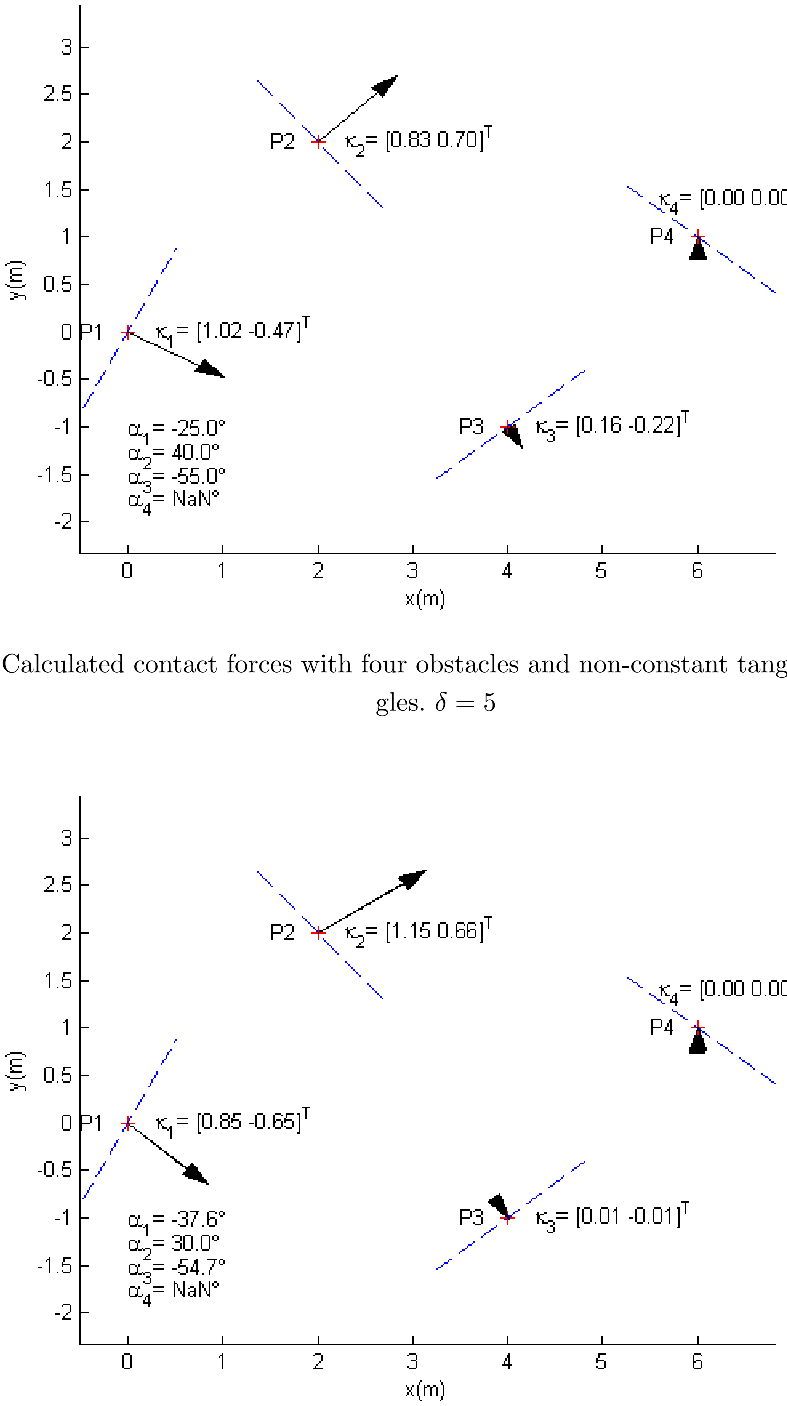
Finding the optimal set of contact forces that minimized the variance of the contact forces was solved by optimizing (4.10) with $n = 4$ obstacles. The following parameters was given to initialize the system:

$$\begin{aligned}
\boldsymbol{P} = \begin{bmatrix} P_1 & ... & P_4 \end{bmatrix} &= \begin{bmatrix} 0 & 2 & 4 & 6 \\ 0 & 2 & -1 & 1 \end{bmatrix} \\
\boldsymbol{\theta} &= \begin{bmatrix} -30^\circ & 45^\circ & -54^\circ & 54^\circ \end{bmatrix}^T \\
\boldsymbol{\gamma} &= \begin{bmatrix} -1 & 1 & -1 & 1 \end{bmatrix}^T \\
\boldsymbol{\tau_\sigma} &= \begin{bmatrix} -2 & 0 & 0 \end{bmatrix}^T
\end{aligned}$$

The results of the optimization is presented in Fig 5.4. The desired propulsion vector is achieved while utilizing every obstacle for propulsion

5.2 Simulating a dynamic system with the controller based on the static model

In the following section we discuss the possibility of using the torques found as a function of the static contact forces as an input to a dynamic snake



(a) Calculated contact forces with four obstacles and non-constant tangent angles. $\delta = 5$

(b) Calculated contact forces with four obstacles and non-constant tangent angles. $\delta = 15$

Figure 5.3: Optimal set of contact forces for four obstacles and non-constant tangent angles. Each obstacle is marked by a red cross. The initial tangent of contact is marked by a blue, dashed line. The force at each point is marked by the arrow, the direction of the arrow indicating the direction of the force and the length indicating the magnitude. α_i is the optimized contact force angle

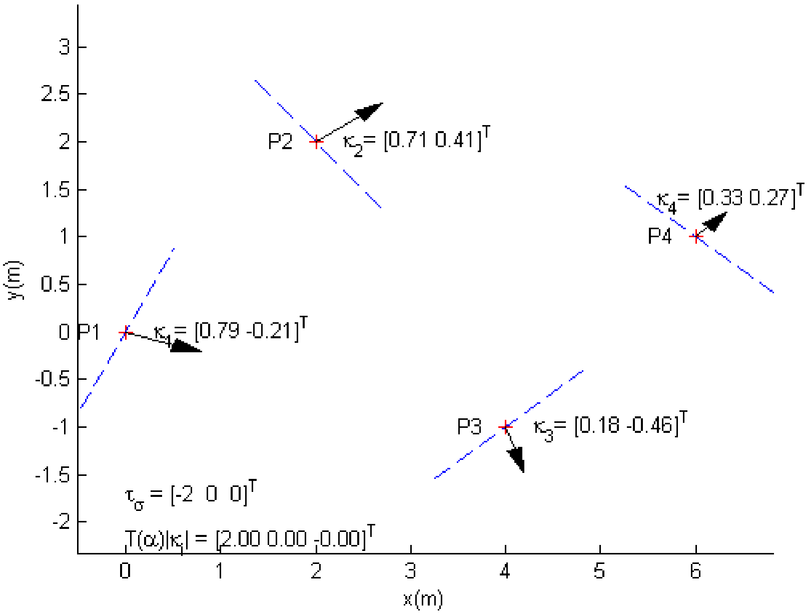


Figure 5.4: Result of optimization based on minimum variance of the contact forces. Each obstacle is marked by a red cross. The initial tangent of contact is marked by a blue, dashed line. The force at each point is marked by the arrow, the direction of the arrow indicating the direction of the force and the length indicating the magnitude. τ_σ is the desired propulsion vector while $T(\alpha)|\kappa_i|$ is the calculated propulsion vector.

robot model. This section tries to combine the controller scheme presented in Chapter 3 with a dynamic model.

5.2.1 A brief introduction to the dynamic model

The dynamic model is based on the work of Liljebäck 2011 [1] and will only be briefly introduced in this report. We refer to [1] for further details regarding development and analysis of the model.

The snake robot consists of N rigid links, each of length $2l$ interconnected by $N-1$ motorized joints. All N links have the mass m and moment of inertia $J = 1/3ml^2$. We assume that the mass of each link is uniformly distributed so that the link centre of mass (CM) is located at its centre point. See figures 5.5 and 5.6 for an illustration of the snake robot.

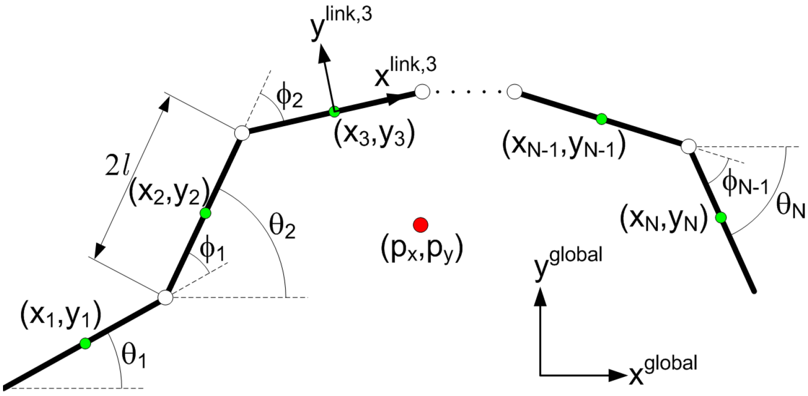


Figure 5.5: The kinematic parameters of the snake robot

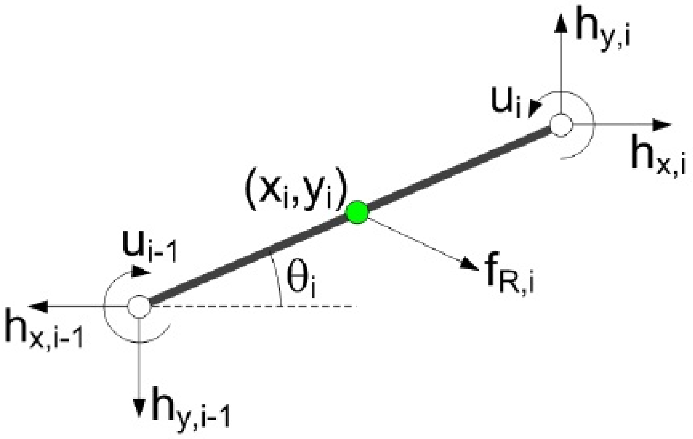


Figure 5.6: Forces and torques acting on each link of the snake robot

It can be shown that the snake robot equations of motion are as follows: The sum of forces on link i in global frame coordinates is given by

$$m\ddot{x}_i = f_{R,x,i} + h_{x,i} - h_{x,i-1} \tag{5.2a}$$

$$m\ddot{y}_i = f_{R,y,i} + h_{y,i} - h_{y,i-1} \tag{5.2b}$$

Table 5.1: Parameters that characterize the snake robot

Symbol	**Description**
N	The number of links.
l	Half the length of a link.
m	Mass of each link.
J	Moment of inertia of each link.
θ_i	Angle between link i and global x axis.
ϕ_i	Angle of joint i.
(x_i, y_i)	Global coordinates of the CM of link i.
(p_x, p_y)	Global coordinates of the CM of the robot.
u_i	Actuator torque exerted on link i from link $i+1$.
u_{i-1}	Actuator torque exerted on link i from link $i-1$.
$f_{R,x,i}$	Friction force on link i in the x direction.
$f_{R,y,i}$	Friction force on link i in the y direction.
$h_{x,i}$	Joint constraint force in x direction on link i from link $i+1$
$h_{y,i}$	Joint constraint force in y direction on link i from link $i+1$
$h_{x,i-1}$	Joint constraint force in x direction on link i from link $i-1$
$h_{x,i-1}$	Joint constraint force in y direction on link i from link $i-1$

The sum of torques for link i is given by

$$J\ddot{\theta}_i = u_i - u_{i-1} - l\sin\theta_i(h_{x,i} + h_{x,i-1}) + l\cos\theta_i(h_{y,i} + h_{y,i-1}) \tag{5.3}$$

Letting $\boldsymbol{x} = [\boldsymbol{\theta}^T\ \boldsymbol{p}^T\ \dot{\boldsymbol{\theta}}^T\ \dot{\boldsymbol{p}}^T]^T \in \mathbb{R}^{2n+4}$ be the state variable of the system and by expressing (5.2) and (5.3) on vector form for all N links we can write the system in state space form as

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{\theta}} \\ \dot{\boldsymbol{p}} \\ \ddot{\boldsymbol{\theta}} \\ \ddot{\boldsymbol{p}} \end{bmatrix} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}) \tag{5.4}$$

To introduce external obstacles forces to the model, consider that the snake is either in contact with an obstacle or not. This approach to contact modelling makes the model a hybrid model. Contact with an obstacle can be modelled as a unilateral velocity constraint on the contacted link, expressed as

$$\gamma_i v_{n,1} \geq 0 \tag{5.5}$$

where $v_{n,1}$ is the normal direction velocity of link i and γ_i is the contact parameters of link i as defined in Table 3.1. The hybrid model of the system uses two concepts called the *jump* map/set and *flow* map/set. A jump occurs when a link impacts or detaches from an object and we say that a link is an element of the jump set, if that link is in contact with an obstacle. Flow occurs when the jump set is empty, which means that there are no impacts and the state vector evolves according to the flow map. Without presenting the details of the modelling process, the complete hybrid snake robot model can be written as

$$\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}) \ \ , \boldsymbol{x} \in \boldsymbol{C} \tag{5.6a}$$

$$\boldsymbol{x}^{+} = \boldsymbol{G}(\boldsymbol{x}) \ \ , \boldsymbol{x} \in \boldsymbol{D} \tag{5.6b}$$

where $\boldsymbol{C}$ is the flow set and $\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})$ is the flow map of the system and $\boldsymbol{D}$ is the jump set and $\boldsymbol{G}(\boldsymbol{x})$ is the jump map of the system. This concludes the introduction to the dynamic model used in the simulations.

5.2.2 Simulation

We were not able to show that the proposed controller could properly control the snake to utilize the obstacles as push-points. Results done for simulating the snake robot for one time step and applying the forces perpendicular to select robot links proved inconclusive, yet promising in terms of achieving the desired propulsion vector. However, no functioning position controller was implemented, thus making simulation over longer periods of time useless as the snake would just move away from the obstacles.

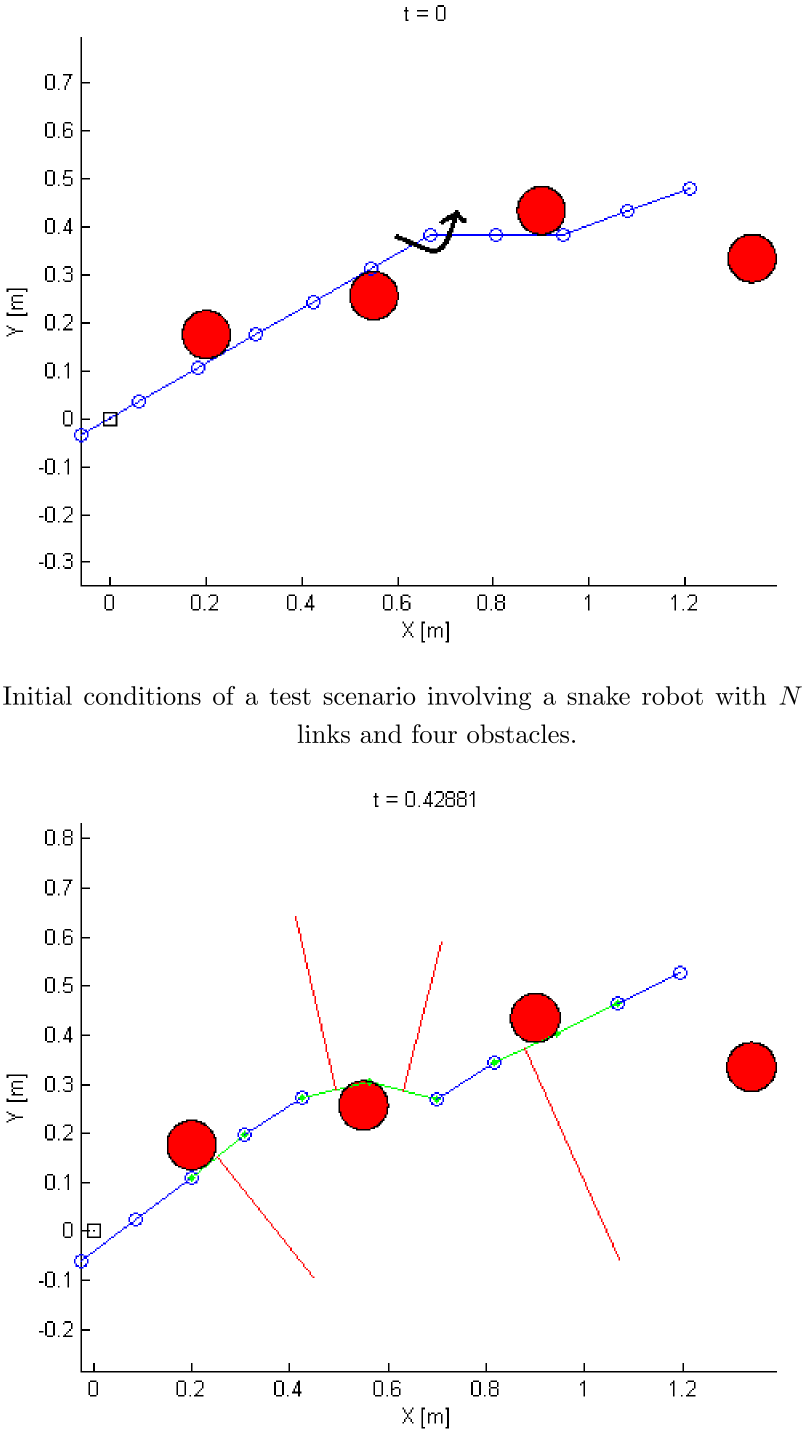
A test case like the one illustrated in Fig **??** was set up to motivate obstacle-aided locomotion. No position controller was implemented for the scenario, but by actuating key joints it was possible to make the snake robot move forward along the obstacles. Because there was no control of the position this only worked as long as the snake had no place to move, i.e. it was bounded by the obstacles. Still, it motivates the usefulness of obstacle-aided locomotion and shows that it was possible to achieve desired forward propulsion.

5.2.3 Discussion of simulating on the dynamic model

We weren't able to show conclusive results that the control scheme could control the snake robot when applied on a dynamic simulator. There might be several reasons for that.

- Finding a good position controller turned out to be more difficult than foreseen and is thus not added in the controller scheme. In the simple scenario presented above we were still unable to control the snake robot to actually follow the path while also applying the joint torques that would give the desired contact forces. Note that path following in this context is not the same as the path following controller presented in e.g. Liljebäck, Haugstuen 2010 [29] because this controller only requires the heading to converge to given reference value, thus creating an oscillatory motion around the path, while we require every joint angle to follow the given path. The assumption from Section 3.1 was that we could track the path perfectly and was required for the static model to have any relevance for a dynamic case.

- When deciding on a desired propulsive vector τ_σ we examine the path of the snake and see what the propulsive vector has to be in order



(a) Initial conditions of a test scenario involving a snake robot with $N = 10$ links and four obstacles.

(b) Simulation after 0.42 seconds.

Figure 5.7: Simulation of a test scenario where the initial position of the snake is configured to be in contact with three obstacles. Torque is applied at joint 6 and the snake moves as a result of the obstacle forces. The red lines are obstacle forces.

for the snake to follow the path. There is, however, no feedback loop designed for specifying a new τ_σ. Finding an expression for the velocity along the curve and controlling that could be a possible solution, though no such controller has been proposed here.

- Finding a mapping from the static situation to the dynamic situation is probably more complex than expressed in this thesis. The assumption that we can find the next propulsive vector by looking at the snake path requires us to know the time step which we generally don't.
- Without a good position controller the snake robot would never lie close to an obstacle for long enough to verify that the snake robot was able to utilize an obstacle as an efficient push-point.
- The dynamic model assumes that each contact force affects the center of mass of the link, which creates scenarios as that shown in **??**b where one obstacle produces force on the center of mass of two links, even though the center of mass is not in contact with the obstacle itself.

Chapter 6 Concluding Remarks and Future Work

In this study we have reviewed previous works on snake robotic control with obstacles. Secondly, we presented a static model of a snake robot in contact with obstacles and investigated different optimality criteria for the shape of the snake robot under certain assumptions and found ways to choose a set of optimal contact forces given a desired propulsion vector. Lastly, we proposed a controller scheme based on calculating the input torques as a function of the desired contact forces and a position controller. No final position controller was presented, however, and thus no real investigation of performance could be done.

6.1 Concluding Remarks

Snake robot control in an unstructured environment is a challenging control problem. There seems to be two different strategies of handling the obstacles. Either trying to avoid them or by using them as push points. This study has favoured the latter approach, as this seems to be how snakes operate in their natural habitat. There has been presented various snake robot models that accounts for obstacle forces, but common for all of them is the fact that they become quite complex and mathematical proofs and stability analysis can be difficult. This study looked at a simplified case where a snake robot in contact with obstacles was modelled in a static way. Thus, various optimization problems could be tested for finding a set of contact forces that matched a given propulsion vector. If energy efficiency is valued highly, the snake robot should tend to choose as few obstacles as possible while still achieving propulsion. It is also possible to attempt to balance the forces on all the obstacles available, which is probably robust in terms of achieving propulsion.

A controller based on the static model in addition to a joint position controller was proposed and was tried with a dynamic model of the snake robot in an obstacle environment. No conclusions could be guaranteed to whether the approach of static modelling is viable or if it's too simplifying, but certainly without a functional position controller it was not possible to make it work in general.

6.2 Future work

Creating a functional controller seems to be the most important step in order to validate the concept presented in this study with the added emphasis on the joint position control aspect of the controller. An analysis regarding stability and robustness of such a controller would then be required. If successful on a simulator, the controller should be tested on a physical snake robot.

The static approach to modelling the snake robot and its contacts with the environment was based on several assumptions. Eliminating some of these assumptions would make the modelling approach more accurate.

In terms of optimality of shape, little focus was given to the relation between choosing the tangents of contact and the path in between the obstacles. Given friction, this will effect any attempts to optimize energy efficiency and should be investigated. This could possibly be combined with a study on how the snake robot can sense its surroundings and choose an optimal path accordingly.

Bibliography

[1] P. Liljebäck, “Modelling, development, and control of snake robots,” Ph.D. dissertation, Norwegian University of Science and Technology (NTNU), Dept. of Engineering Cybernetics, 2011.

[2] S. Hirose, *Biologically Inspired Robots: Snake-Like Locomotors and Manipulators.* Oxford: Oxford University Press, 1993.

[3] A. A. Transeth, R. I. Leine, C. Glocker, K. Y. Pettersen, and P. Liljebäck, “Snake robot obstacle aided locomotion: Modeling, simulations, and experiments,” *IEEE Trans. Rob.*, vol. 24, no. 1, pp. 88–104, 2008.

[4] A. A. Transeth, K. Y. Pettersen, and P. Liljebäck, “A survey on snake robot modeling and locomotion,” *Robotica*, vol. 27, pp. 999–1015, 2008.

[5] S. Hasanzadeh and A. Tootoonchi, “Obstacle avoidance of snake robot moving with a novel gait using two-level pid controller,” in *Robotics, Automation and Mechatronics, 2008 IEEE Conference on*, sept. 2008, pp. 427 –432.

[6] X. Wu and S. Ma, “Development of a sensor-driven snake-like robot sr-i,” in *Information and Automation (ICIA), 2011 IEEE International Conference on*, june 2011, pp. 157 –162.

[7] K. Inoue, S. Ma, and C. Jin, "Optimizationof cpg-network for decentralized control of a snake-like robot," in *Robotics and Biomimetics (ROBIO). 2005 IEEE International Conference on*, 2005.

[8] A. Crespi and A. J. Ijspeert, "Online optimization of swimming and crawling in an amphibious snake robot," *IEEE Trans. Robotics*, vol. 24, no. 1, pp. 75–87, 2008.

[9] S. Hasanzadeh and A. Tootoonchi, "Adaptive optimal locomotion of snake robot based on cpg-network using fuzzy logic tuner," in *Robotics, Automation and Mechatronics, 2008 IEEE Conference on*, sept. 2008, pp. 187 –192.

[10] C. Ye, D. Hu, S. Ma, and H. Li, "Motion planning of a snake-like robot based on artificial potential method," in *Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference on*, dec. 2010, pp. 1496 –1501.

[11] D. Yagnik, J. Ren, and R. Liscano, "Motion planning for multi-link robots using artificial potential fields and modified simulated annealing," in *Mechatronics and Embedded Systems and Applications (MESA), 2010 IEEE/ASME International Conference on*, july 2010, pp. 421 –427.

[12] Y. Shan and Y. Koren, "Design and motion planning of a mechanical snake," *IEEE Trans. Syst. Man Cyb.*, vol. 23, no. 4, pp. 1091–1100, July-August 1993.

[13] ——, "Obstacle accommodation motion planning," *IEEE Trans. Robotic. Autom.*, vol. 11, no. 1, pp. 36–49, 1995.

[14] J. Gray, "The mechanism of locomotion in snakes," *J. Exp. Biol.*, vol. 23, no. 2, pp. 101–120, 1946.

[15] G. P. Hicks, "Modeling and control of a snake-like serial-link structure," Ph.D. dissertation, North Carolina State University, 2003.

[16] A. A. Transeth, R. I. Leine, C. Glocker, and K. Y. Pettersen, "3D snake robot motion: Non-smooth modeling, simulations, and experiments," *IEEE Trans. on Robotics*, vol. 24, no. 2, pp. 361–376, April 2008.

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

[18] Z. Bayraktaroglu and P. Blazevic, "Understanding snakelike locomotion through a novel push-point approach," *J. Dyn. Syst. - Trans. ASME*, vol. 127, no. 1, pp. 146–152, March 2005.

[19] Z. Y. Bayraktaroglu, A. Kilicarslan, A. Kuzucu, V. Hugel, and P. Blazevic, "Design and control of biologically inspired wheel-less snake-like robot," in *Proc. IEEE/RAS-EMBS Int. Conf. Biomedical Robotics and Biomechatronics*, February 2006, pp. 1001–1006.

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

[21] P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "A hybrid model of obstacle-aided snake robot locomotion," in *Proc. IEEE*

Int. Conf. Robotics and Automation, Anchorage, AK, USA, 2010, pp. 675–682.

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

[23] M. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control.* John Wiley & Sons, 2006. [Online]. Available: http://books.google.no/books?id=wGapQAAACAAJ

[24] O. Harkegard, "Efficient active set algorithms for solving constrained least squares problems in aircraft control allocation," in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 2, dec. 2002, pp. 1295 – 1300 vol.2.

[25] T. I. Fossen, *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles.* Trondheim, Norway: Marine Cybernetics, 2002.

[26] H. dong Wang, J. qiang Yi, and G. liang Fan, "Autonomous reconfigurable flight control system design using control allocation," in *Systems and Control in Aerospace and Astronautics, 2008. ISSCAA 2008. 2nd International Symposium on*, dec. 2008, pp. 1 –6.

[27] T. Fossen and S. Sagatun, "Adaptive control of nonlinear underwater robotic systems," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, apr 1991, pp. 1687 –1694 vol.2.

[28] J. Nocedal and S. Wright, *Numerical optimization*, ser. Springer series in operations research. Springer, 2006. [Online]. Available: http://www.google.com/books?id=eNlPAAAAMAAJ

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