Norwegian University of Science and Technology

# Hybrid Position/Force Control for Obstacle-aided Locomotion in Snake Robots

TTK4550 Teknisk Kybernetikk, Fordypningsprosjekt


*Report by:*

Torjus Klafstad

*Supervised by:*

Associate Professor Øyvind Stavdahl

NTNU: Norges teknisk-naturvitenskapelige universitet


January 8, 2019

# Abstract

Although snake robots are widely talked about in several branches of science, most if not all implementations of snake robot locomotion are based on ad-hoc-methods and preprogrammed gaits and movements, for instance applying a sine curve to joint angle for lateral undulation. Current snake robot theory lacks an analytical solution for locomotion, something hybrid position/force control could provide.

In this report, the applicability of Hybrid Position/Force Control for Obstacle-Aided Locomotion in Snake Robots is researched. Specifically, the method described in West and Asada (1985) of modeling constraints due to contact with the environment as virtual joints and splitting the joint space into one position controlled and one force controlled subset by use of projection matrices is applied to a snake robot model. The method is then shown to produce correct results for some static cases, verified with trivial static analysis. Further directions for research are presented.

Although it needs quite a lot more testing and developing, hybrid position/force control for snake robots seems promising as a new, better method for obstacle-aided locomotion. This report aims to serve as a thorough and comprehensive guide for further research and development.

# Nomenclature

| | |
|---|---|
| $\phi_{ci}$,$\tau_{ci}$ | Position and force variables, respectively, associated with the virtual rotational joint of contact point $i$ |
| $\tau_i$ | Actuator torque for joint $i$ |
| $\mathbf{c}_i$ | Position of the $i$-th contact point |
| $\mathbf{J}$ | Jacobian matrix |
| $\mathbf{J}_c$ | Jacobian for a contact point |
| ${}^{a}_{b}\mathbf{T}$ | The transformation matrix from frame $a$ to frame $b$ |
| $x_{ci}$,$f_{ci}$ | Position and force variables, respectively, associated with the virtual translational joint of contact point $i$ |
| CKC(s) | Closed Kinematic Chain(s) |
| DOF(s) | Degree(s) of freedom |
| OAL | Obstacle-Aided Locomotion |

# Table of Contents

# List of Figures

Chapter 1

# Introduction

Snake robots are an interesting and complex part of robotics that is currently researched globally. They are a link between biology and technology, as a way of delving deeper into nature for inspiration and progress in science. Snake robots serve both a practical and theoretical purpose; the practicality of a robot that can move into areas humans cannot speaks for itself, and the theory helps with deeper understanding of the movement of limbless creatures. However, most, if not all, snake robots are modeled and controlled based on heuristics and predetermined gaits. Current snake robots are (very) good at *mimicking* the behavior and movement of real snakes, but a snake robot that *thinks* and *acts* like a biological snake is far more desirable. Obstacle-aided locomotion is a likely way forward. Pushing against the environment creating propelling forces is the basis for almost all kinds of snake locomotion (and for nearly all other species, but not primarily on the plane), and a "correct" model of this could greatly improve both performance and understanding of snake robots.

The method proposed in this report for creating a better snake robot model is built on the principle of *hybrid position/force control*[1], more precisely the method presented in West

[1]In "position/force control", "position" implies "position and orientation" and "force" implies "force and

and Asada (1985), in which constraints due to contact with the environment are modeled as virtual joints in the robot. Intuition says that this method is applicable and descriptive, and the goal of this report is, by use of mathematics and robotics theory, to show to which degree that intuition is correct.

## 1.1 Motivation

A new form of control for snake robots is motivated by the current lack of analytical approaches for snake robot locomotion. Current implementations are suboptimal in the sense that they are ad-hoc methods. The form of hybrid position/force control presented in West and Asada (1985) could, with some modifications, be applied to a snake robot utilizing multiple kinds of locomotion. Hybrid control provides an analytical approach, based on physics and linear algebra, which will in turn give mathematically "correct" results. These can further be used, since there are analytical expressions, to find and create optimal paths and trajectories.

Applying the theory to obstacle-aided locomotion for the snake robot seems the best choice. Obstacles fit well into the constraint model of West and Asada (1985). Further work may open the possibility for using obstacle-aided locomotion as a basis for all snake locomotion, which then makes having a strong foundation utterly necessary. Properly modeling this will produce a qualitatively better solution to the snake locomotion problem. This kind of model may be a solution to several problems previously encountered within snake robot locomotion, for instance the *jam detection/jam resolution*-problem discussed in Liljebäck et al. (2012).

torque". This will apply for the rest of this paper as well.

## 1.2 Problem Description

The purpose of this project is to examine the possibilities of hybrid position/force control for obstacle-aided locomotion in snake robots. This is done by first carrying out a literature review on previous work within hybrid position/force control theory, and describing the state of the art with relevance to snake robots. A couple of idealized, static scenarios, which are easily verifiable, are then given to illustrate the applicability of hybrid position/-force control for snake robots in cluttered environments. Simulations of the scenarios are carried out in MATLAB to see if the intuition is correct (for those cases). The results are evaluated, and topics and directions for further research into this problem are presented. The structure of this report is as follows:

1. Literature Review and Theory
2. Application of the Theory to a Snake Robot
3. Simulation of Different Examples
4. Results, Discussion, and Topics for Further Research

Chapter 2

# Literature Review and Theoretical Background

In this chapter a literature review on the relevant subjects is carried out. Relevant results from the papers are presented, and necessary theoretical background for further understanding is described. It is, for the rest of this paper, assumed that the reader possesses sufficient knowledge of robot kinematics and linear algebra.

## 2.1 Hybrid Position/Force Control for Constrained Robot Manipulators

Raibert and Craig (1981) were among the first to introduce the concept of hybrid position/-force control. The idea was based around controlling both force and position for a robot manipulator within the same model. This is often desirable, for instance when polishing a car with a robot. It is then essential both to apply a sufficient force to the surface of the car, as well as moving the end effector over the entirety of the surface to polish the whole car. West and Asada (1985) further expanded on this idea by allowing constraints at places in contact with the robot at other areas than the end effector. They further generalized the theory to be applicable to any constrained robot problem with any number of constraints.

### 2.1.1 Constrained Manipulators Problems

A robot manipulator performing a task will more often than not be constrained in one way or another. Constraints on a manipulator mainly occur in two different forms, namely position constraints and force constraints.
A position constraint could be a physical wall or object in the work space which a robot simply would not be able to move through. The constraint is, naturally, active when the robot is in contact with the surface of the constraint.
Force constraints have a somewhat less intuitive explanation. For a rigid, frictionless surface, a manipulator would be able to apply arbitrarily large forces normal to the surface, as the surface would provide equally large reactions. It is, however, not possible to apply a force to nothing, so any directions tangent to a surface are directions where force is constrained (to be zero).

### 2.1.2 Raibert & Craig

All information in the following paragraph is collected from Raibert and Craig (1981), and is interpreted here for the sake of clarity and understanding of the discussed subject.

### 2.1.3 Hybrid Position/Force Control

The motivation of Raibert and Craig for creating a hybrid position/force controller was, as it usually is, based on optimization, both in control and cost. Their results indicate that for manipulator problems where the end effector is constrained, hybrid control of both position and force at the end effector provides efficient and accurate control.

### The Hybrid Controller

Raibert and Craig's hybrid controller begins with defining a constraint frame {C}, a Cartesian coordinate frame which relates to the task geometry. Their hybrid controller idea is that each joint torque actuation signal $\tau_i$ is made up of a sum of components, with each component corresponding to an either force controlled or position controlled DOF in {C}. Each DOF in {C} is either position or force controlled, decided on by inspection. A selection matrix **S** is utilized for selecting between position and force controlled DOFs.

This hybrid controller setup works well enough, but lacks built-in methods for constraints other than at the end effector. This would complicate the choosing of a constraint frame immensely, as it would have to be continuously updated. Luckily, West and Asada (1985) improved upon the otherwise solid basis of Raibert and Craig (1981) with the possibility for constraints other than at the end effector.

## 2.1.4 West & Asada

This section reiterates some of the results from the hybrid controller design of West and Asada (1985), with the intention of making this report sufficient for understanding the subject of interest.

### Closed Kinematic Chains

Closed kinematic chains, or CKCs, are essential for analysis of manipulators constrained by contact with the environment. A robot manipulator is generally an open kinematic chain, i.e. one end of the robot is connected to a base, while the rest of the robot is only connected to the other links. Such a robot will normally have as many DOFs at the end effector as it has links.

In a CKC the end of the kinematic chain is rigidly connected to a surface. The contact point is rigid, hence no movement in any direction is possible at the end of the chain. With no movement possible, there can be no work done at the end either. These properties of CKCs are essential for West and Asada's hybrid controller design.

West and Asada diverged from the constraint frame design of Raibert and Craig (1981), as this method would, with constraints due to contact between the manipulator and the environment at a point other than at the end effector, be inefficient and computationally demanding. They proposed a different way of modeling constraints: as a number of artificial joints linking the manipulator to the environment, creating a CKC. This makes the constraint part of the robot kinematics, and opens up for a different approach to designing the position/force controller.

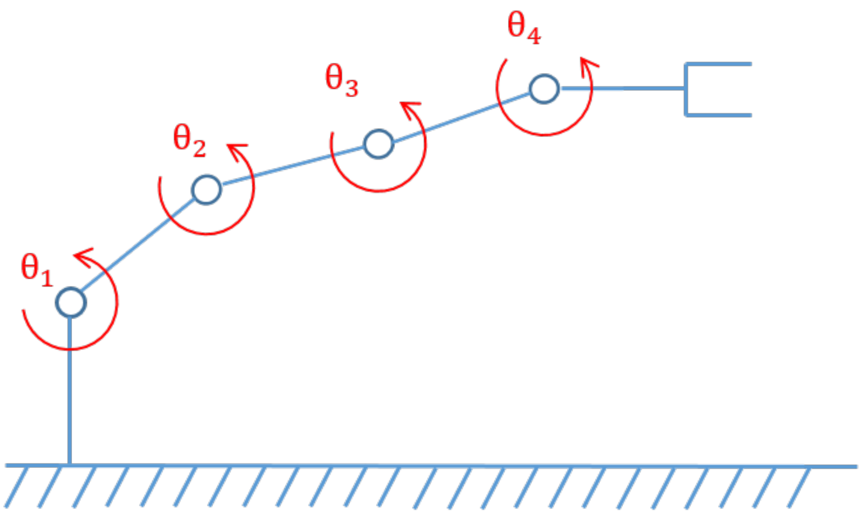



**Figure 2.1:** An open kinematic chain

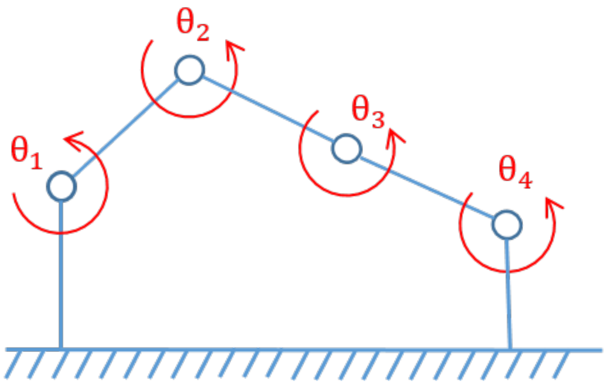



**Figure 2.2:** A closed kinematic chain

### Allowable Space

A robot forming a CKC will have a Jacobian matrix describing its kinematics. West and Asada call this matrix $\mathbf{J}_c$, and for simplicity the same notation will be used here. The properties mentioned in section 2.1.4 can be expressed as equations:

$$\delta \mathbf{x}_c = \mathbf{J}_c \delta\theta = \mathbf{0} \tag{2.1}$$

which has solutions:

$$\delta\theta = (\mathbf{I} - \mathbf{J}_c^+ \mathbf{J}_c)\mathbf{y} \tag{2.2}$$

where the superscript $+$ denotes a generalized inverse and $\mathbf{y}$ is an arbitrary vector. The second property, that no work can be done at the end of the kinematic chain, can be formalized as an equation as well:

$$\boldsymbol{\tau}^T \delta\theta = \boldsymbol{\tau}^T (\mathbf{I} - \mathbf{J}_c^+ \mathbf{J}_c)\mathbf{y} = 0 \tag{2.3}$$

with solution

$$\boldsymbol{\tau} = (\mathbf{J}_c^+ \mathbf{J}_c)^T \mathbf{z} \tag{2.4}$$

where $\mathbf{z}$ is an arbitrary vector, and $\boldsymbol{\tau}$ is a vector of joint forces. Now West and Asada recognize that the matrices $[\mathbf{I} - \mathbf{J}_c^+ \mathbf{J}_c]$ and $[\mathbf{J}_c^+ \mathbf{J}_c]^T$ are orthogonal (2.5), and therefore span the joint space of the CKC. The matrices are idempotent and are therefore projectors, projecting onto what West and Asada call *the allowable forces and directions*.

$$[\mathbf{I} - \mathbf{J}_c^+ \mathbf{J}_c] \cdot [\mathbf{J}_c^+ \mathbf{J}_c]^T = 0 \tag{2.5}$$

This follows from the analysis of CKCs. An intuitive interpretation of equations (2.2) and (2.4) supports this: given desired position and force vectors **y** and **z**, the matrices will project these onto position and force vectors allowed by the CKC configuration. West and Asada introduce a naming convention for these kinds of matrices:

$$_j\mathbf{P}_{ap} = [\mathbf{I} - \mathbf{J}_c^+ \mathbf{J}_c] \tag{2.6}$$

$$_j\mathbf{P}_{af} = [\mathbf{J}_c^+ \mathbf{J}_c]^T \tag{2.7}$$

where the $j$-pre-subscript indicates a projection in joint space, subscript $a$ for allowable and subscript $p$,$f$ for position and force, respectively. Further subscript convention used is $w$-pre-subscript denoting a work space coordinate frame, and subscript $e$ for essential position and force directions.

#### Essential Space and Coordinate Transform

West and Asada describe the robot task in work space with the help of what they call essential variables. They define essential position variables as the $\alpha$ directions at the end effector in need of position control, and the $\beta$ directions in contact with the environment in need of force control. Grouping vectors $\mathbf{e}_{pi}$ and $\mathbf{e}_{fj}$ in matrices

$$\mathbf{E}_p = [\mathbf{e}_{p1}, \ldots \mathbf{e}_{p\alpha}] \tag{2.8}$$

$$\mathbf{E}_f = [\mathbf{e}_{f1}, \ldots \mathbf{e}_{f\beta}] \tag{2.9}$$

gives a basis for constructing orthogonal projections onto the essential variable spaces:

$$ {}_w\mathbf{P}_{ep} = \mathbf{E}_p(\mathbf{E}_p^T\mathbf{E}_p)\mathbf{E}_p^T \tag{2.10} $$

$$ {}_w\mathbf{P}_{ef} = \mathbf{E}_f(\mathbf{E}_f^T\mathbf{E}_f)\mathbf{E}_f^T \tag{2.11} $$

Note that the projections (and essential directions and forces) are defined in work space.

### 2.1.5 The Controller Design

To be able to both project onto the allowable and essential directions, the projections need to be defined in the same space. How to decide which one is explained in West and Asada (1985). The relevant transformation requires the system Jacobian $\mathbf{J}$, and transforms the essential variables into the joint space:

$$ {}_j\mathbf{P}_{ep} = \mathbf{J}^+ \ {}_w\mathbf{P}_{ep}\mathbf{J} \tag{2.12} $$

$$ {}_j\mathbf{P}_{ep} = \mathbf{J}_c^T \ {}_w\mathbf{P}_{ef}(\mathbf{J}_c^T)^+. \tag{2.13} $$

There is, almost definitely, a typographical error in West and Asada (1985) equations (19) and (20), corresponding to equations (2.12), (2.13) above.
The projection onto the essential space and allowable space are presumably not equal. West and Asada here construct a filter for a desired joint motion/torque. This filter both projects onto the allowable joint space and retains the essential directions of the desired joint motion/torque. A such filter is, in joint space,

$$ {}_j\mathbf{F}_p =_j \mathbf{P}_{ap}(\ {}_j\mathbf{P}_{ep}\ {}_j\mathbf{P}_{ap})^+\ {}_j\mathbf{P}_{ep} \tag{2.14} $$

$$ {}_j\mathbf{F}_f =_j \mathbf{P}_{af}(\ {}_j\mathbf{P}_{ef}\ {}_j\mathbf{P}_{af})^+\ {}_j\mathbf{P}_{ef}. \tag{2.15} $$

The final stage is implementing this filter in a hybrid controller scheme, which is not shown in this report, but can be seen in West and Asada (1985).

## 2.2 Snake Locomotion

The purpose of this section is to give a brief overview of some methods biological snakes use for movement. Most of the information on lateral undulation comes from Liljebäck et al. (2012), and obstacle-aided locomotion from Transeth et al. (2008); Sanfilippo et al. (2017).

### 2.2.1 Lateral Undulation

Lateral undulation is the most commonly seen form of snake locomotion. The snake moves forward by moving its body in a wave pattern, its body pushing against the environment. As the snake moves forward, the wave propagates backwards through the body. During lateral undulation, every part of the body is always moving. In addition, if the snake doesn't slip, then every point of the snake touches the same point on the path. Most snakes will lift their body at points not contributing efficiently to forward locomotion, i.e. at and around the peaks of the body curve (here force vectors would point perpendicular to the desired direction). This form of locomotion is obviously useless in frictionless environments [1].

### 2.2.2 Concertina Locomotion

In concertina locomotion, the snake is usually moving in a confined area, for instance in a pipe or between two walls. The snake then curves its body and pushes against the sides of its confinement. This creates an anchor in the environment, allowing the snake to move its head forward, uncurling. The snake continues to curl its body up and stretch it out, which is where the name concertina(bokmål: trekkspill) comes from.

### 2.2.3 Obstacle-Aided Locomotion

Obstacle-aided locomotion sounds counterintuitive. Obstacles usually hinder progress in one way. For a snake, however, an obstacle in its path will often provide means for propulsion. For instance, lateral undulation could be simplified down to OAL. The friction between the snake and the surface is, on a microscopic scale, the snake pushing against irregularities, obstacles, on the surface, which then again creates a net forward force. In concertina locomotion the snake utilizes the confined space it is in to anchor itself and create propulsion. The confined space is again an obstacle for the snake used for locomotion. Some of the anchor effect, however, comes from friction at the anchor points.

OAL can be used on a frictionless surface as long as there are rigid obstacles present. By forming its body into any shape in contact with several obstacles, the snake can move itself by applying force to the contact points. Changing the angle of contact with the obstacles leaves the snake open to move in any desired direction. This is simply the sum of forces working on the snake moving it forward, the same principle as in lateral undulation, only discrete. This is depicted in figures 2.3 and 2.4.

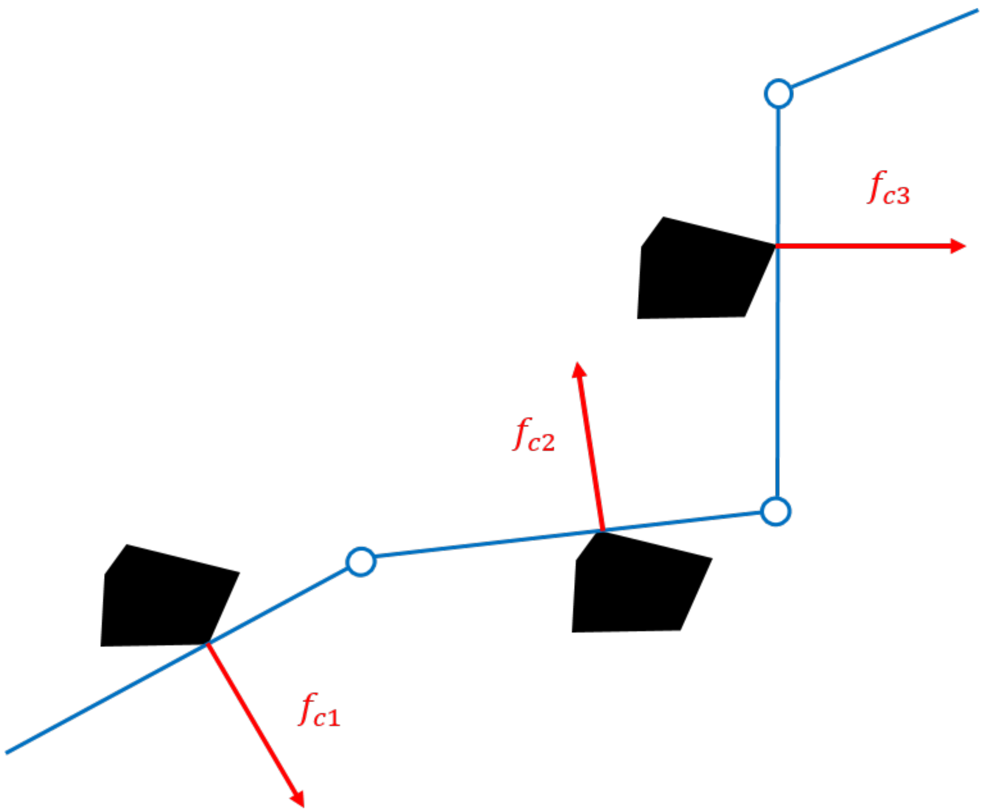



**Figure 2.3:** A snake robot using nearby obstacles for locomotion

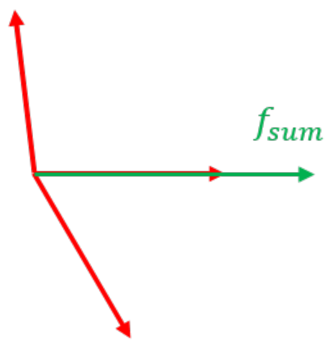



**Figure 2.4:** The sum of forces acting on the robot in fig 2.3

Chapter 3

# Application to the Snake Robot

OAL has been shown by Transeth et al. (2008) to be usable for a wheel-less robot. Their implementation, however, relies on parametrizing the gait of the robot as a sinusoidal curve, limiting the robot to strictly defined environments. The method of hybrid position/force control for OAL will in theory allow the robot to follow qualitatively different trajectories.

A snake robot using OAL will always be in contact with a number of obstacles in its environment. Then there will be forces acting on the snake normal to the point of contact, and these forces are what create propulsion. Controlling the forces at these contact points is then clearly of interest. The snake robot will be aiming to follow a certain trajectory as well, which makes position control of interest too. Controlling them separately is definitely possible, but probably not the best way. Hybrid position/force control, as formulated in West and Asada (1985), separates the joint space into two orthogonal subspaces. One is the direction of position control, the other the direction of force control. It is then possible to control both aspects simultaneously within the same control architecture.

Any form of propulsion is regrettably out of scope for this report. Justification for the applicability of hybrid position/force control will be provided through analysis of static examples and models. For the remainder of the report, the following assumptions are made:

1. The robot has $n$ joints, with all links length $l$ and end effector link length $l_h$.
2. The robot is massless.
3. The robot and the environment is frictionless.
4. The robot has no lateral extension.
5. The obstacles have no spatial extent, considered points in the plane.
6. Only static examples are considered.

## 3.1 Snake Robot Model

In this section, a model for a planar snake robot is given. The design is based on the one described in Stavdahl (2018).

### 3.1.1 Disconnection from Reference Frame

A snake robot differs from other robot manipulators in one essential way: a snake robot is not connected to a base coordinate frame by physical joints. A locomotive snake robot will necessarily progress along a path in an environment, with the path most likely defined by obstacles. It is still of interest and importance to know where the snake is situated in a reference frame. The disconnection between an inertial frame and the snake robot can still be modeled adhering to robot kinematics principles. The tail end of the robot will have some coordinates $\mathbf{p_0} = \begin{bmatrix} x_0 & y_0 & \phi_0 \end{bmatrix}^T$ in the reference frame. Connecting the robot to the origin is done by introducing three virtual joints: a rotation $\phi_0$ around the (global) vertical axis, a translation $y_0$ along the new $y$-axis, and a translation $x_0$ along the new $x$-axis. These terms are trivially included in the forward kinematics. A depiction of these virtual joints is shown in figure 3.2.

### 3.1.2 Other Remarks

Some things to note: in static cases, the coordinates of the tail are assumably known or of no importance. During locomotion, however, the coordinates are subject to, and most likely will change. In a realistic, physical scenario the coordinates must be found in some way. Without too much thought laid into the subject, a few possible solutions are presented. Solution one is a position sensor giving the $x$,$y$-coordinates in a grid, with $\phi_0$ explicitly found equal to $\phi_1$ in the world frame. The other solution is the robot having active memory of where it began and where it's path has gone, and by *proprioception* calculate the tail coordinates.

Another note is that the current configuration with the rotation as the initial link will result in the coordinates in the reference frame actually not being $\begin{bmatrix} x_0 & y_0 & \phi_0 \end{bmatrix}^T$, but instead $\begin{bmatrix} x_0 \cos\phi_0 - y_0 \sin\phi_0 & x_0 \sin\phi_0 + y_0 \cos\phi_0 & \phi_0 \end{bmatrix}^T$. However, this is a trivial calculation. An argument could be made for placing the rotational joint last, but the current configuration (subjectively) looks better.

The end effector is depicted slightly differently from normal. The argument is simply that it looks more like a snake and helps discern it from a common robot manipulator.

A certain color scheme is decided. Blue indicates physical joints of the robot. The virtual joints are depicted in green. Variable names, physical properties, and properties relating to the task are in red. Coordinate frames are in black.

## 3.2 Constraint Model

The hybrid controller for the snake robot will be based on the hybrid controller in West and Asada (1985), and it is therefore necessary to find a way to model the constraints. In obstacle-aided locomotion, the robot aims to be in contact with the obstacles on its path, as they are essential for propulsion. This is equivalent to a constraint where a part of the robot is fixed to the environment. West and Asada have shown that such a constraint can be modeled as a set of artificial joints. Then, how to model it for a snake robot? The proposed solution is quite similar to West and Asada's: a translational joint parallel to the link of the robot, and a revolute joint around the vertical axis. A visual description is given in figure 3.1. Note figure 3.1c, which is the proposed representation of point obstacles in contact with the snake robot: a combination of a translational and revolute joint, with the obstacle being on the same side of the robot joint as the circle. It would be a more correct representation with the circle center on the joint, but that would make it indiscernible on which side the obstacle is located.

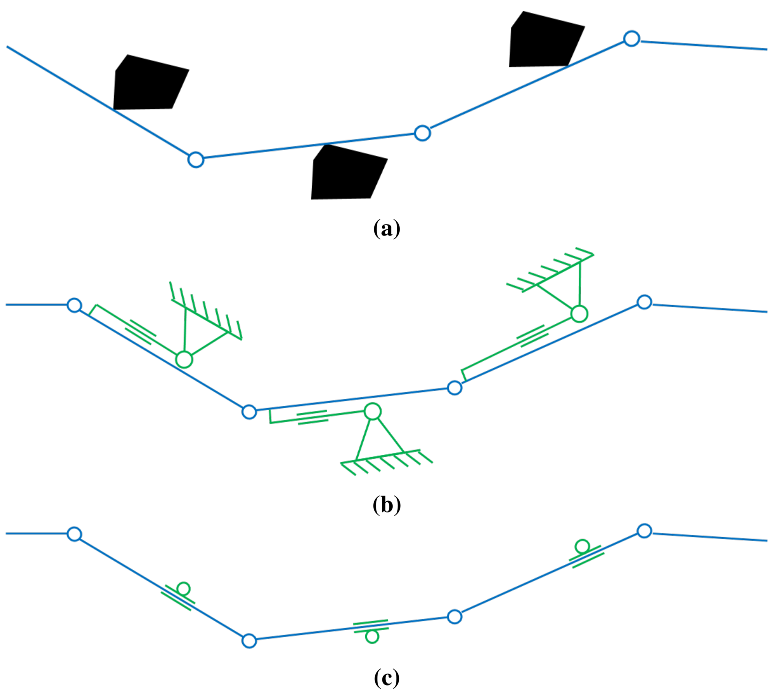




**Figure 3.1:** Visual representation of constraint model: (a) Snake robot with three points of contact with the environment; (b) constraint modeled with a translational and rotational joint; (c) simplified version of constraint model

## 3.3 Snake Robot Kinematics

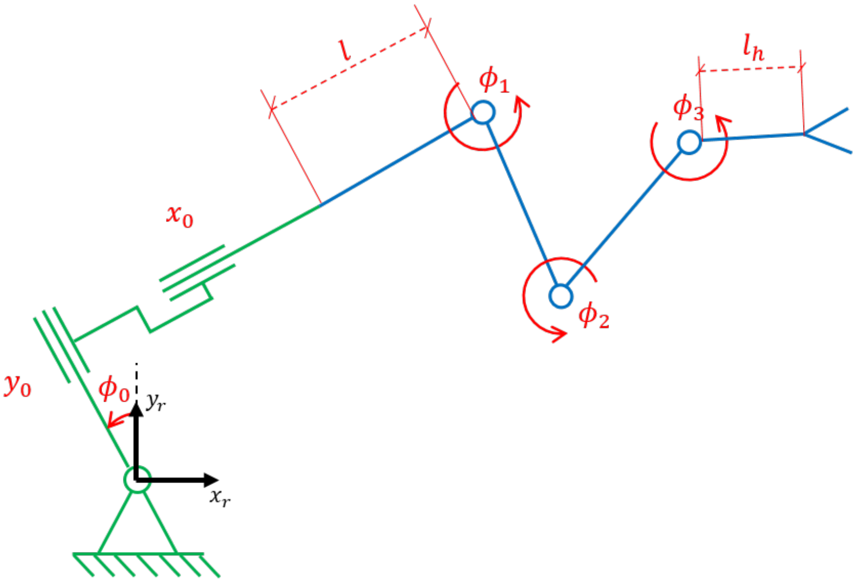



**Figure 3.2:** Model of a snake robot with three joints

Finding the forward kinematics of a planar snake robot is trivial. For the three-linked robot in figure 3.2, given coordinates $[x_0, y_0, \phi_0]$ of the tail link (the first link of the robot), length $l$ of the robot links, and joint coordinates $[\phi_1, \phi_2, \phi_3]$, the transformation from the reference frame to the end effector is

$$ {}^r_h\mathbf{T} = \mathbf{R}_z(\phi_0)\mathbf{D}_y(y_0)\mathbf{D}_x(x_0)\mathbf{D}_x(l)\mathbf{R}_z(\phi_2)\mathbf{D}_x(l)\mathbf{R}_z(\phi_2)\mathbf{D}_x(l)\mathbf{R}_z(\phi_3)\mathbf{D}_x(x_h) \quad (3.1)$$

where

$$\mathbf{R}_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{D}_x(x) \quad = \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{D}_y(y) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix}$$

are rotational and translational matrices around and along the respective axes in two dimensions. Adding another link will simply result in another translational and rotational element in equation (3.1).

The transformation matrix will be of the form

$$ {}^{r}_{h}\mathbf{T} = \begin{bmatrix} [\mathbf{R}(\phi_h)] & [\mathbf{o}_h^r] \\ \mathbf{0} & 1 \end{bmatrix} $$

where $\phi_h$ and $[\mathbf{o}_h^r] = \begin{bmatrix} x_h & y_h \end{bmatrix}^T$ is the snake head's orientation and origin in the reference frame. **R** is a rotation matrix in 2-dimensional space, i.e. a matrix of the form $\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$, so the angle $\phi_h$ can be found by taking the inverse tangent of the positive sine term divided by the cosine term.

A vector of generalized coordinates **q** for the robot in figure 3.2 will be

$$ \mathbf{q} = \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 & \phi_0 & y_0 & x_0 \end{bmatrix}^T \tag{3.2} $$

and the coordinates of the head of the robot are

$$ \mathbf{p}_h^r = \begin{bmatrix} x_h & y_h & \phi_h \end{bmatrix}^T = \begin{bmatrix} f_x(\mathbf{q}) & f_y(\mathbf{q}) & f_\phi(\mathbf{q}) \end{bmatrix}^T \tag{3.3} $$

The time derivative of the head position is then

$$ \dot{\mathbf{p}}_h^r = \dot{\mathbf{f}}(\mathbf{q}) \tag{3.4} $$

$$ = \frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} \tag{3.5} $$

$$ = \mathbf{J}\dot{\mathbf{q}} \tag{3.6} $$

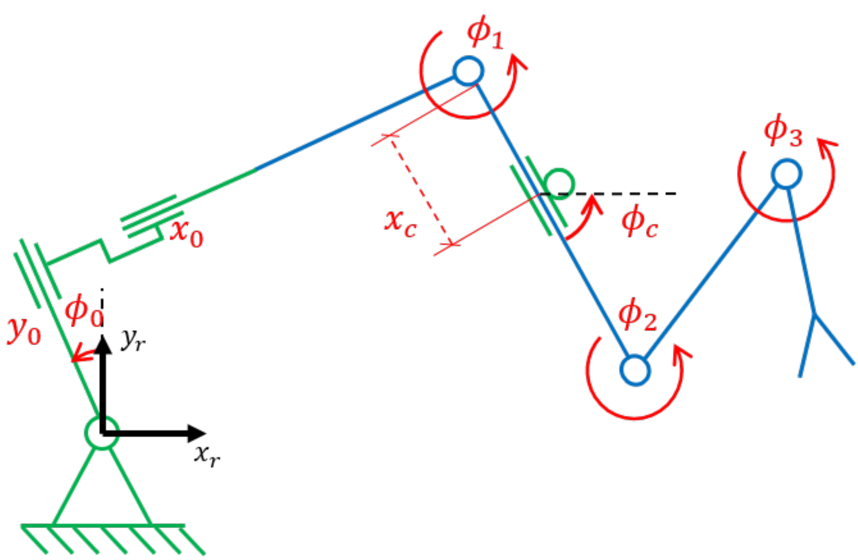



**Figure 3.3:** Snake Robot with modeled constraint

where $\mathbf{J}$ is the Jacobian matrix:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_x(\mathbf{q})}{\partial q_1} & \frac{\partial f_x(\mathbf{q})}{\partial q_2} & \dots & \frac{\partial f_x(\mathbf{q})}{\partial q_5} \\ \frac{\partial f_y(\mathbf{q})}{\partial q_1} & \frac{\partial f_y(\mathbf{q})}{\partial q_2} & \dots & \frac{\partial f_y(\mathbf{q})}{\partial q_5} \\ \frac{\partial f_\phi(\mathbf{q})}{\partial q_1} & \frac{\partial f_\phi(\mathbf{q})}{\partial q_2} & \dots & \frac{\partial f_\phi(\mathbf{q})}{\partial q_5} \end{bmatrix} \tag{3.7}$$

The procedure for finding the contact Jacobians is quite similar. Consider the snake robot with the constraint model in figure 3.3. The transformation becomes

$$^r_c\mathbf{T} = \mathbf{R}_z(\phi_0)\mathbf{D}_y(y_0)\mathbf{D}_x(x_0)\mathbf{D}_x(l)\mathbf{R}_z(\phi_1)\mathbf{D}_x(x_c)\mathbf{R}_z(\phi_c) \tag{3.8}$$

Following the same procedure as for the end effector gives the contact Jacobian $\mathbf{J}_c$. Note that introducing constraints will increase the number of generalized coordinates; the new $\mathbf{q}$ is

$$\mathbf{q} = \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 & \phi_0 & y_0 & x_0 & x_c & \phi_c \end{bmatrix}^T. \tag{3.9}$$

## 3.4 Projections

In the case of a multiple constrained robot, such as the snake robot, more constraints will give more contact Jacobians for use with West and Asada's controller design. The question then arises: how to create projections onto the allowable spaces? West and Asada (1985) provide a solution to this as well.

Consider the case with a robot constrained by contact with the environment at $i$ places. Then it is possible to find the pair of projection matrices $({}_j\mathbf{P}_{api}, {}_j\mathbf{P}_{afi})$ corresponding to each contact point $c_i$.

The multiple constrained robot will need to abide by all the position constraints, i.e. it cannot physically move other than in the directions specified by the allowable motion projections. Hence, a new projection matrix must project onto directions allowable for each constraint alone. One such projection is the projection onto the *intersection* of the spaces spanned by the allowable position projectors.

All allowable forces must be able to be exerted. Therefore a projection matrix must be able to project onto all allowable force vectors, the span of the projection must be equal to all allowable force directions. This is done by projecting onto the *union* of the spaces spanned by the allowable force projectors.

A lot of the joints are passive, specifically all the virtual joints. Since they are passive, the forces in them cannot be controlled. If unattended to, this could and most likely would result in the allowable force projection projecting onto uncontrollable directions. There is, however, an simple workaround for this: constructing a diagonal matrix $\mathbf{A}$, with square dimension equal to the number of (virtual and actual) joints, where a 1 on the diagonal indicates an active joint and 0 a passive. Projecting onto the intersection of $\mathbf{A}$ and ${}_j\mathbf{P}_{af}$ solves the issue.

There are several ways to project onto the intersection or union of two or matrices. Examples are given in the Appendix.

## 3.5 Task Analysis

The task of the snake robot is, ultimately, to move along a path defined by obstacles present in its environment to reach some goal destination, utilizing the obstacles for locomotion. Disjointing the problem, the task is to

1. apply forces to the contact points resulting in propulsion, and
2. have the snake follow a certain path while moving.

In the static case, this reduces to

1. apply forces to the contact points, maintaining force balance, and
2. have the snake adjust to a certain path.

How to describe these tasks? For static cases, the first part at least is quite simple: the force part of the task is to have forces at the contact points, described with the desired force directions at those points. And for the static cases, these directions are the allowed force directions! Or possibly a linear combination of the allowed directions. Hence a dimension of West and Asada (1985), the essential force directions, falls away.

The second point is the position controlled part of the hybrid controller. Here the task is for the snake to have the same geometrical shape as a predefined path. For static cases, it is reasonable to assume that the robot is in contact with the contact points that keeps it

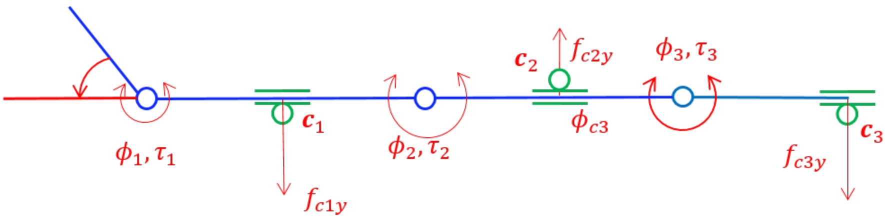



**Figure 3.4:** An example of position control of snake robot. Desired path in red.

stationary, and that at these points the robot lies on the optimal path. Then the complimentary joint directions, that are not used for force control, can be used to align the robot to the path. A simple example of this is given in figure 3.4. In the figure, the contact forces $f_c = \begin{bmatrix} f_{c1y} & f_{c2y} & f_{c3y} \end{bmatrix}^T$ only depend on the torques in the second and third joint; the allowed force direction lie in the ($\tau_2$,$\tau_3$)-plane (in this case the only allowed direction is $\tau_2 = \tau_3$, which will be shown in a later section). This leaves $\phi_1$ (or the torque signal $\tau_1$) free for position control.

For the task specification, a mapping from the path to the joint angles is required. In this simple example it is clear that the joint angle $\phi_1$ should be zero, but how can this be put into a desired value for a controller? Obviously the allowed values for $\phi_1$ are not the same as the desired/essential values, so the connection used for force control cannot be applied for position control. This is one of the snags of this problem formulation. Since the positional task is not defined at the end effector, West and Asada's essential position variables are not easily described and are possibly not implementable, requiring a different way of solving that part of the problem.

No definitive, correct solution to the above problem in this report, and is subject for further research. One possible solution is nevertheless given below.

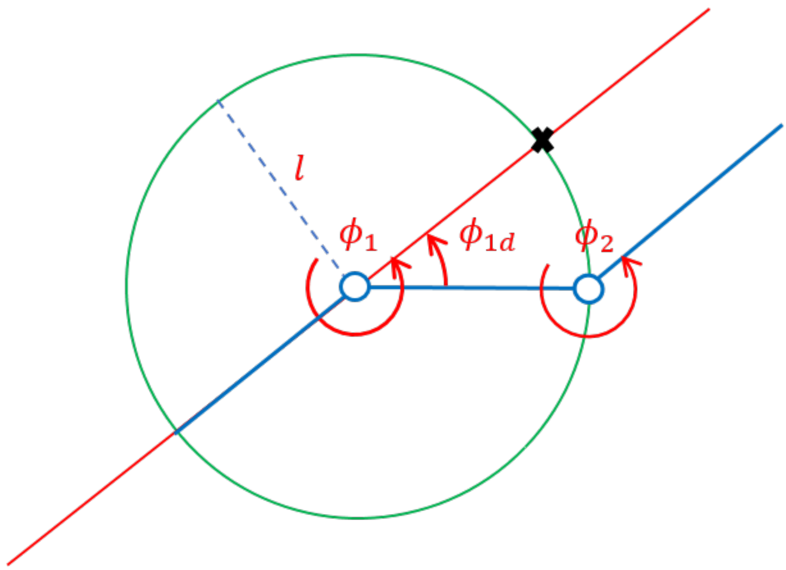



**Figure 3.5:** Depiction of the proposed solution for position control.

### 3.5.1 Outline of Solution to Position Control

The control should focus on the actual joint positions being on the curve, disregarding the rest of the link. This method requires at least one joint position to be on the desired trajectory. From one such position, the desired joint angle of *that* joint can be found by finding the intersection between a circle with radius $l$, the link length, around the joint, and the optimal path parametrization. The desired joint angle is then the angle of the vector between the joint position and the intersection in the world frame. The idea is shown in figure 3.5, with the desired joint angle $\phi_{1d}$. For the snake robot this procedure must naturally be repeated for each joint not on the path, suggesting an iterative approach.
This method works on paper, but could be computationally ineligible, given the increased complexity from the large amount of joints, and the possibility of changes in the environment. It has not been implemented or tested in any way.

### 3.5.2 Notes on an Optimal Path

The optimal path must be found, but this is not quite in the scope of this report. Some properties of an optimal path, however, are presented without much proof. Firstly, the path is designed to consist of arcs and straight line segments. There are several reasons for this. One, to reduce friction in the desired direction, although this only becomes necessary for locomotion in environments with friction. Two, to allow for simple calculations of distance between current position and desired position.

The arcs should in this case have the minimal radius allowed by the physical properties of the robot, and the path should have a minimum amount of turns as well, again for minimization of friction.

The path must be decided from the obstacles present in the environment, i.e. the optimal path is the optimal path given a set of (utilized or not) obstacles.

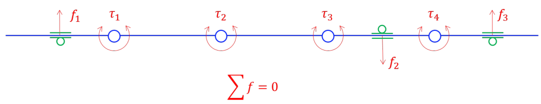
Chapter 4

# Examples and Simulations

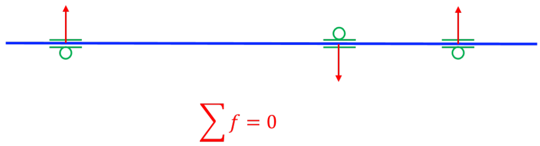
## 4.1 Geometric Justification for Static Examples

As only static examples will be discussed in this report, it is important to establish what constitutes a static example and how it may take form. The examples discussed will mostly consider *bracing*, i.e. cases where the robot exerts forces on its environment but remains unmoving.

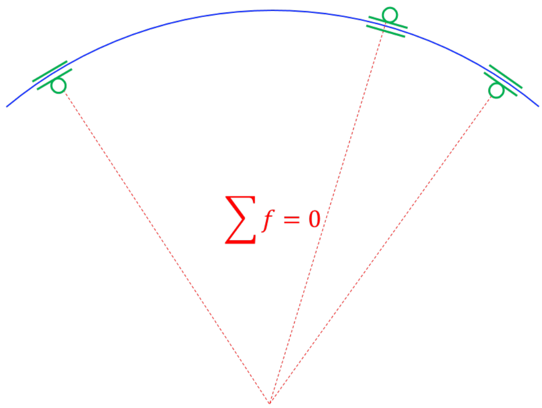
Consider the case depicted in figure 4.1a; a snake robot lying in a straight line in contact with three obstacles. This obstacle configuration will not be able to create any propulsion when the robot lies in a straight line 5. The proposed justification is a geometric study. The contact forces here lie parallel to each other. In figure 4.1c a snake lying on an arc braces against three contact points. The force vectors intersect in a single point, the center of the circle on which the snake is lying. It is then easy to see that the sum of the forces must be zero, as otherwise the robot would be collapsing inward or outward to or from the circle center. Note that if the force vectors weren't intersecting in one single point, there would be no possible way to achieve zero net force in any direction, and it would thereby



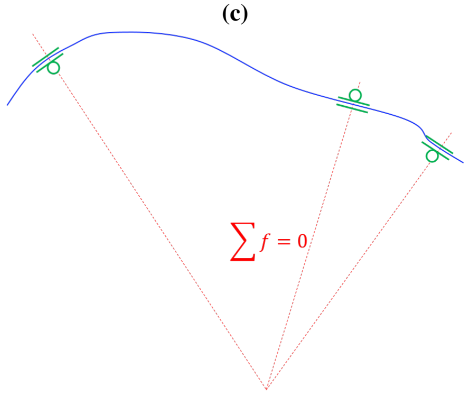

**(a)**




**(b)**




**(c)**




**(d)**

**Figure 4.1:** Series of figures to represent static cases for snake robot in cluttered environment: (a) Snake robot with three points of contact with the environment; (b) simplified version of (a); (c) contact points and force sum on for snake robot on an arc; (d) contact points and force sum for snake robot on arbitrary path

be acceleration in some direction. In figure 4.1d the same argument can be applied. The forces intersect in a single point, the only difference is that the contact points don't lie on the same circle centered in the intersection, but rather on circles centered in the intersection with different radii.
Then, let the radius of the circle approach infinity. Any lines now intersecting in the center will be parallel, and hence the sum of the forces has to be zero in order to have a static case.

This identity helps simplify the static cases to any case where the robot is lying straight and in contact with at least three points. It is now possible to apply the theory of West and Asada (1985) on an example.

## 4.2 Example: 2 Links, 3 Constraints

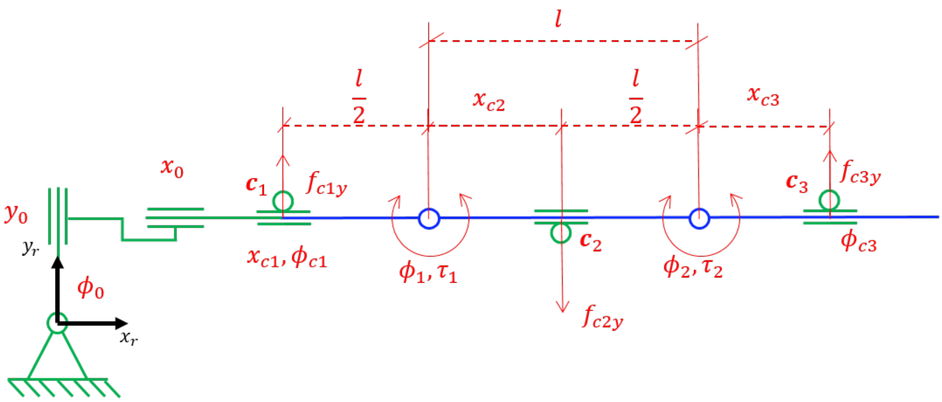



**Figure 4.2:** The constrained snake robot discussed in section 4.2

One exceedingly simple example of a snake robot in a static case is given in figure 4.2. The advantage of studying static examples is that the results are easy to verify. Knowing that we are studying a static case, finding the allowable force directions at the contact points is simply force and torque balance.
For now, let $l = 2$, $\phi_0 = x_0 = y_0 = x_{c1} = 0$, and $x_{c2} = x_{c3} = \frac{l}{2} = 1$. Force balance

gives that

$$f_{c1y} + f_{c3y} = f_{c2y} \tag{4.1}$$

and moment balance around the point $c_2$ gives

$$2f_{c1y} = 2f_{c3y} \implies f_{c1y} = f_{c3y} \tag{4.2}$$

from equations (4.1), (4.2) it is clear that

$$f_y = \begin{bmatrix} f_{c1y} \\ 2f_{c1y} \\ f_{c1y} \end{bmatrix} = k \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \tag{4.3}$$

where $k$ is an arbitrary constant, and $f_y$ is the vector of forces at the contact points required for the system to be static. It is trivial that the actuator torques must be equal for the system to remain in a static state, hence the allowable robot force space is

$$c\begin{bmatrix}1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}^T,$$

where $c$ is an arbitrary constant. Clearly there is one degree of freedom for this system in the "force" direction. The control setpoint would here be the constant $k$, which follows directly from $c$.

With the basis for verification above, what does West and Asada's method produce? With the given configuration, the forward kinematics are derived using the equations in section 3.3, as well as the contact Jacobians. The interest lies currently only in forces in the joints. Equation (2.7) gives the projections onto the allowable force space for the contacts, ${}_j\mathbf{P}_{af1}, {}_j\mathbf{P}_{af2}, {}_j\mathbf{P}_{af3}$. The projection ${}_j\mathbf{P}_{afc}$ projects onto the union of the spaces spanned by the contact projections. Finally, ${}_j\mathbf{P}_{af}$ is the projection onto the intersection of the space spanned by ${}_j\mathbf{P}_{afc}$ and $\mathbf{A}$, where

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

With the initial values for **q**:

$$\mathbf{q} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix},$$

the projection becomes

$$ {}_j\mathbf{P}_{af} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.4}$$

which has basis

$$\begin{bmatrix}1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}^T,$$

which means $\tau_1 = \tau_2$ is the only allowable force direction according to West and Asada's method!

## 4.3 Example: 3 Links, 4 Constraints

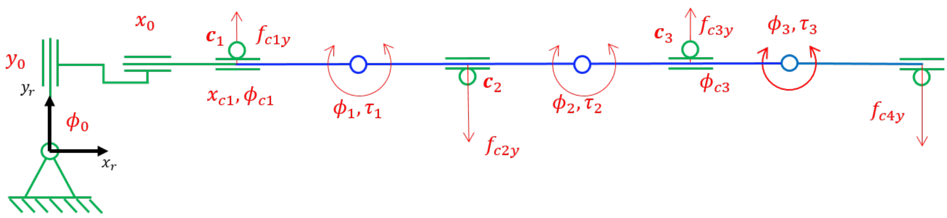



**Figure 4.3:** The constrained robot discussed in section 4.3. Note all distances and lengths are equal to the ones in figure 4.2.

The example in figure 4.3 is a bit more complex, with one extra link and constraint. Here the force balance isn't unique, as it is possible to push off against different combinations of contact points and stay still: $(c_1,c_2,c_3)$, $(c_2,c_3,c_4)$, and $(c_1,c_2,c_3,c_4)$. However, the torque balance around all contact points must still be zero, and works as verification.

In matrix form, the torque balance around all the contact points is

$$\begin{bmatrix} 0 & -1 & 2 & -3 \\ -1 & 0 & 1 & -2 \\ -2 & 1 & 0 & -1 \\ -3 & 2 & -1 & 0 \end{bmatrix} \begin{bmatrix} f_{c1y} \\ f_{c2y} \\ f_{c3y} \\ f_{c4y} \end{bmatrix} = 0 \tag{4.5}$$

which reduces to

$$\begin{bmatrix} 0 & 1 & -2 & 3 \\ 1 & 0 & -1 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} f_{c1y} \\ f_{c2y} \\ f_{c3y} \\ f_{c4y} \end{bmatrix} = 0 \tag{4.6}$$

so clearly there are two DOFs of allowable forces. All allowable force directions are linear combinations of the vectors $\begin{bmatrix} 1 & 2 & 1 & 0 \end{bmatrix}^T$, $\begin{bmatrix} -2 & -3 & 0 & 1 \end{bmatrix}^T$ (here $f_{c3y}$ and $f_{c4y}$ are chosen as free variables). The directions of the forces are in a way taken into account, in that $f_{c1y}$,$f_{c3y}$ are defined positive upwards and $f_{c2y}$,$f_{c4y}$ are defined positive downwards. These allowed contact force vectors correspond directly to allowed joint torque directions. Clearly the first vector is the same as in the previous example, which means that the corresponding allowed torque vector is $\begin{bmatrix} 1 & 1 & 0 \end{bmatrix}^T$. By trivial inspection the other torque vector is $\begin{bmatrix} -2 & -1 & 1 \end{bmatrix}^T$. The passive joint elements are omitted for simplicity.

The approach for finding allowable torques through the West and Asada-method is almost identical to the one described in the previous example, and will not be repeated here. Reassuringly, the method provides the exact same results as the above static analysis, which further solidifies the method as applicable.

## 4.4 Remarks

Further examples serve no distinct purpose. The method has been shown to be applicable to simple, idealized scenarios. The geometrical argument in section 4.1 serves as proof that any static scenario for the snake robot can be reduced to a straight example.

One observation made testing different scenarios is that the number of DOFs for allowable torque/force is equal to two less than the number of contact points. This correlation was verified for $N$(number of contact points) from 3 up to 20, although no mathematical proof has been derived. The correlation was found when observing that the matrix for torque balance around the contact points (see equation (4.5)) always had rank 2. The proof is most likely found by inspection of the near-symmetry of that matrix.

One final remark is that increasing the amount of joints in between the contact points does nothing to the number of allowable force directions, so extra joints will only add to the allowable directions for position control.

Chapter 5

# Discussion

## 5.1 Results

The method of hybrid position/force control seems applicable for a snake robot using OAL. The modeling of the contact points as virtual joints and applying the theory of West and Asada (1985) produced the same results as the static analysis, showing the applicability of the theory.

## 5.2 Further Research

This report set out to be a comprehensive study of hybrid position/force control and its applicability to snake robots. However, that goal was quickly discovered to be unreachable for a part-time, half year research undertaking for one student. Several facets of hybrid position/force control for OAL in snake robots have been touched upon in this report. Nearly all of them, however, need to be improved upon and further researched before an actual physical robot can be constructed adhering to those principles.

There are several important facets of this theory that need further research. Locomotion needs to be introduced in the system, which means dynamics of a such robot are necessary. Finding task descriptions, both for force and position control, is extremely important. Further analysis of different scenarios is needed to find a method for finding optimal paths and optimal force directions at the contact points. Mass and friction must also be introduced at some point. Choice of controllers is a point for further research as well. The same goes for stability analysis.

In summary, a mountain of work dwarfing the small hill this report represents still remains for the future development and application of this theory.

## 5.3 Conclusion

Hybrid position/force control is possibly a new paradigm shift in the design of snake robots. An analytical approach to solving a locomotive problem is certainly desirable, and hybrid position/force control seems to provide exactly that; a method based on known physics, mathematics, and robot manipulator theory. The results presented are certainly promising, and feel intuitively "correct".

However, too much work remains undone for this report to reach a definite conclusion. Further research may disprove the applicability of this theory in real-life scenarios, although this seems highly unlikely. As this project will not be continued into a master's thesis, it is up to someone else to pick up where this report ended for continued work. Hopefully this report serves as a stepping stone into further research.

# Bibliography

P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl. *Snake Robots : Modelling, Mechatronics, and Control*. Springer, 2012. ISBN 978-1-4471-2995-0.

R. Piziak et al. Snake robot obstacle-aided locomotion: Modeling, simulations, and experiments. *Computers and Mathematics with Applications*, 37:67–74, 1999.

M. Raibert and J. Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 102:126–133, 6 1981.

F. Sanfilippo et al. Perception-driven obstacle-aided locomotion for snake robots: The state of the art, challenges and possibilities. *MDPI Applied Sciences*, 4, 3 2017.

Ø. Stavdahl. Working note: Hybrid position/force control of snake robots for perception driven obstacle aided locomotion. 2018.

A. A. Transeth et al. Snake robot obstacle-aided locomotion: Modeling, simulations, and experiments. *IEEE Transactions on Robotics*, 24:88–104, 2 2008.

H. West and H. Asada. A method for the design of hybrid position/force controllers for manipulators constrained by contact with the environment. *1985 IEEE International Conference on Robotics and Automation*, pages 251–259, 1985.

Appendix A

# Projections onto Intersect and Union of Spaces

West and Asada West and Asada (1985) provide some equations for projecting onto the intersect and union of the spaces spanned by two or more projectors. Given projection matrices $\mathbf{P_L}, \mathbf{P_M}$ that project onto the spaces $\mathbf{L}$ and $\mathbf{M}$, respectively, the projection onto the union of the spaces is given by

$$\mathbf{P_{L \cup M}} = (\mathbf{P_L} + \mathbf{P_M})(\mathbf{P_L} + \mathbf{P_M})^{+} \tag{A.1}$$

where the "$+$"-superscript denotes the pseudoinverse. The projection onto the intersection of the two spaces is given by

$$\mathbf{P_{L \cap M}} = 2\mathbf{P_L}(\mathbf{P_L} + \mathbf{P_M})^{+}\mathbf{P_M}. \tag{A.2}$$

Another form of these projectors are

$$\mathbf{P_{L \cup M}} = \begin{bmatrix} \mathbf{P_L} & \mathbf{P_M} \end{bmatrix} \begin{bmatrix} \mathbf{P_L} & \mathbf{P_M} \end{bmatrix}^{+} \tag{A.3}$$

and

$$\mathbf{P}_{\mathbf{L}\cap\mathbf{M}} = 2(\mathbf{P}_{\mathbf{L}} - \mathbf{P}_{\mathbf{L}}(\mathbf{P}_{\mathbf{L}} + \mathbf{P}_{\mathbf{M}})^{+}\mathbf{P}_{\mathbf{L}}, \tag{A.4}$$

which are collected from Piziak et. al. Piziak et al. (1999).