NTNU – Trondheim
Norwegian University of
Science and Technology

Analyzing Motions of Unicycles and Car-Like Vehicles

Stian Aleksander Tvetmarken

Master of Science in Engineering Cybernetics
Submission date: June 2012
Supervisor: Anton Shiriaev, ITK
Co-supervisor: Uwe Mettin, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem text

Consider nonholonomic mechanical systems such as unicycle and car-like vehicles. Theoretical difficulties arise from non integrable constraints on their velocities so that trajectory generation and stabilization are challenging tasks. Analyze the system by deriving the standard kinematic control system and the corresponding Euler-Lagrange system in which the external forces are introduced. Given a smooth path (preferably composed of joined line and circular segments) and a feasible trajectory along the path, compute a transverse linearization of the Euler-Lagrange system parameterized for the path segments. Compare some properties of the system dynamics of both a four- and a five- degree of freedom nonholonomic system.

Abstract

Two configurations for a unicycle, a class of nonholonomic systems, is looked into. The first configuration is the case where the generalized coordinates consists of four parameters, while the other configurations consists of five. For both systems the equations of motion are calculated. Then a method for parameterizing a desired path for the systems using a synchronization function among all degrees of freedom is shown. A set of equations to find feasible trajectories keeping the desired path virtually constrained is calculated. It is then shown for the four degree of freedom system how to compute the transverse linearization, which can be used for orbital stabilization of a desired motion consisting of circular orbits or straight paths. A way to compute a periodic controller for a linear time varying system is derived, and a set of controllers are tried out on the four degree of freedom unicycle with negative result.

Sammendrag

To konfigurasjoner for en ett-hjuling, tilhørende de ikke-holonomiske systemer, er sett nærmere på. I den første konfigurasjonen består de generaliserte koordinatene av fire parametere, mens i den andre konfigurasjonen består koordinatene av fem parametere. For begge systemene blir bevegelseslikningene beregnet. Så blir en metode for å parametrisere en ønsket bane for systemene, ved hjelp av en synkroniseringsfunksjon blant alle frihetsgradene, vist. Deretter beregnes et sett av ligninger for å finne gjennomførbare baner som holder på ønsket bane ved hjelp av virtuelle begrensninger. Videre blir det vist for systemet bestående av fire frihetsgrader hvordan en transvers linearisering, som kan brukes for orbital stabilisering av en ønsket bane bestående av sirkulære og rette segmenter, kan beregnes. En måte å beregne en periodisk kontroller for et lineært tidsvarierende system er utledet, og et sett av kontrollere har blitt testet ut på etthjulingssystemet bestående av fire frihetsgrader med negativt resultat.

Contents

Chapter 1

Introduction

The last decades there has been an extensive development of different type of robots. Many of them are stationary factory workers doing painting jobs, welding jobs and other assembly line work, but lately there has also been developed more and more mobile robots with wheels, belts and other methods of locomotion. Some of them even have got legs, and move in a human-like way. Different type of robots raise different type of challenges to be solved and the main challenge of mobile robots with wheels and belts is the presence of nonholonomic motion constraints. The directions in which those systems can move are therefore constrained, which makes maneuvering complicated.

The reason why it is important to look at these types of robot locomotion is because their range of use is rapidly increasing. Today automated mobile robots do simple jobs at factories and work places carrying around things from one place to another. At printing plants they carry paper around, at hospitals they carry laundry and at big warehouses they drive around and pick and stack goods. All of these applications involve a supervisory central that always knows where the robots are and what they do. The paths they have to follow are often precomputed and the environment they work in is known, but this is about to change.

More and more robotics work involve autonomous mobile robots being able to find their own way through an environment that may or may not be known. This could be mobile carebots (helping robots at peoples home or at elder's centers), autonomous cars, exploring robots at other planets, surveillance robots and etc. All of these robots need to be able to navigate safely and efficiently to be able to work autonomous in a non-restricted environment.

This report will mainly focus on analysis of motion for a unicycle, which is a

nonholonomic system resembling a class of wheeled robots. The unicycle can be modeled with different complexity depending on consideration of center of mass being able to move out of the vertical plane. In the following analysis the focus will mainly be on the simplest case, namely the four degree of freedom system for which the potential energy is assumed constant. A brief comparison to the five-DOF case will be provided revealing interesting observations about principle motions that can be obtained.

Chapter 2

Problem Description

2.1 Definition of Nonholonomic Constraints

Nonholonomy was briefly mentioned in the introduction as something related to a certain type of constraints. These constraints are often related to vehicles and other mobile devices with steerable wheels or belts. The constraints are known for restricting the velocity vector of the vehicle, i.e. it cannot move sideways, as the wheels cannot roll that way unless the vehicle turns around first. The following definition is based on [1]. The smallest set of quantities defining the configuration of a mechanical system are called the generalized coordinates. They are denoted as $\boldsymbol{q} = [q_1,\ q_2,\ \cdots q_n]^T$ where n denotes the number of generalized coordinates. Sometimes some of the configuration variables are redundant, but they have constraints on possible configurations, such that their dynamics can be expressed by the generalized coordinates. If a given set of $k < n$ constraints on the configuration by smooth functions

$$h_i(q_1, ..., q_n) = 0, \quad i = 1, ..., k, \tag{2.1}$$

they are called holonomic. The motion of the system lies on an $m = (n-k)$-dimensional hypersurface (integral manifold) defined by $h_i(\boldsymbol{q}(t))$ for all $t > 0$. A set of $p < n$ constraints on the velocity

$$< \omega_i(\boldsymbol{q}), \dot{\boldsymbol{q}} >= 0, \quad i = 1, ..., p \tag{2.2}$$

is called Pfaffian, where $\omega_i(\boldsymbol{q})$ are covectors. All constraints on the form of Eq. (2.2) are holonomic if there exists p smooth functions $h_i(\boldsymbol{q})$ such that

$$\omega_i(\boldsymbol{q}) = dh_i(\boldsymbol{q}), \quad i = 1, ..., p. \tag{2.3}$$

Figure 2.1: The URANUS robot with omni directional wheels.

Otherwise they are nonholonomic, that is, no such functions, $h_i(\boldsymbol{q})$, exists. The reason why so many mobile robots have got nonholonomic constraints is because they are built with traditional wheels with steering, similar to how a car works. Some of the problems with nonholonomic constraints are maneuvering and path planning. Maneuvering a car can be tricky, especially parallel parking and navigation in small places. This is because of the differential constraints making it impossible to move sideways. A workaround for this problem is omni directional wheels, which are wheels with rollers which basically can roll in any direction by varying the speed at each wheel on the robot. Fig. 2.1 shows a robot with omni directional wheels. The downside with those kind of wheels is that they are more expensive and is hard to implement on larger vehicles as the diameter needs to be larger and they cause a phenomena called fish-tailing, which increases with the speed. Fish-tailing is basically when the traction gets too low and during a turn the rear end of the vehicle starts to skid. This is also related to another problem with the wheels, namely that because of the rollers the vehicle can be pushed around by external forces when standing still. Also this kind of wheels would cause vibrations and non smooth driving at greater speeds as they would not be perfectly balanced. All in all we see that we cannot throw away the old wheels yet, and we need good ways for planning a motion for such type of vehicles.

2.2 The Unicycle; a Nonholonimic System

This report will as mentioned in the introduction, look at a particular type of nonholonomic system called the unicycle. The unicycle is shown in two

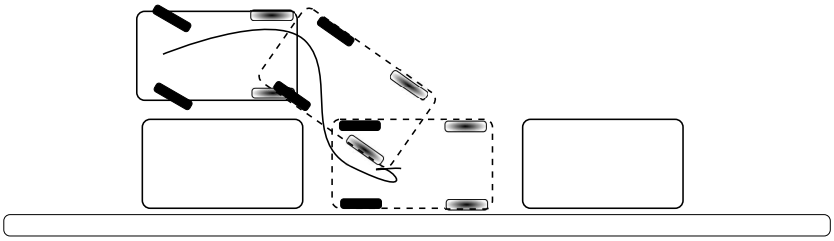
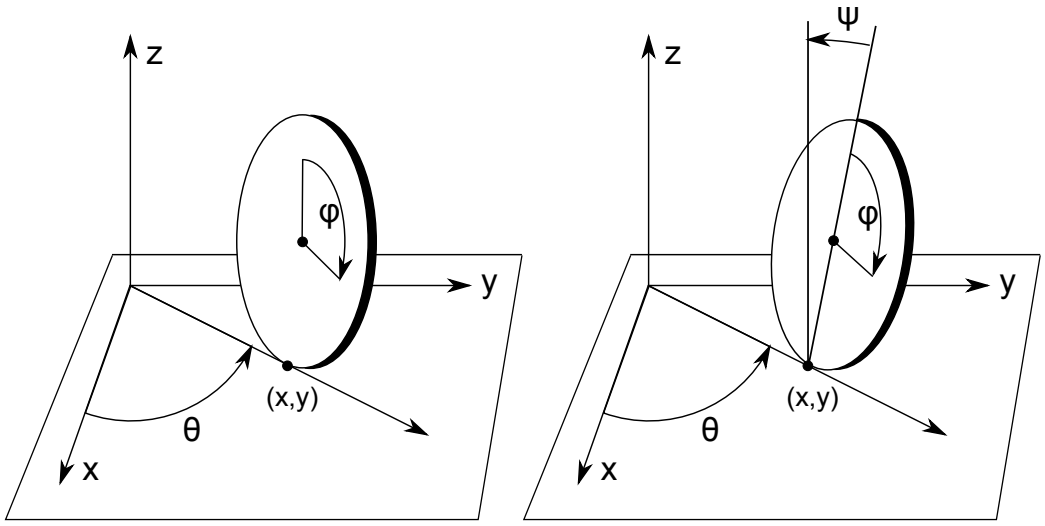


Figure 2.2: Parallel parking can sometimes be a challenge.

different ways in Fig. 2.3 where in Fig. 2.3a the unicycle is represented with four degrees of freedom (DOF) because it is assumed that gravity has no effect. In practice this means that the unicycle will always be in a stable vertical condition and will never fall over. This practical simplification makes the unicycle a system similar to a car-like system where nonholonomic constraints are present, but when standing still the system is stable. Fig. 2.3b introduces a new degree of freedom and gravity is now present. A small tilt will now change the center of gravity (COG) and unless external forces are applied the unicycle will fall over, just like a coin. The five DOF case of the unicycle is very closely related to a real unicycle, which is very agile, highly dynamic and not as representative for vehicles with significant roll dynamics about longitudinal axis.



(a) The 4 DOF unicycle case (b) The 5 DOF unicycle case

Figure 2.3: Comparison of the two different unicycle cases.

2.3 Path Planning and Control

As many people know how do drive a car, the task of making the car move cannot be said to be a difficult task. Parallel parking the car is somewhat more complicated and a survey made on behalf of Elephant car insurance in UK [2] shows that 21% of motorists can rarely or never parallel park while 58% will give up and park another place because they aren't able to manouvre their car into a space. This gives an idea of the restrictions nonholonomic constraints enforces a system, especially when it comes to path planning and manoeuvrability. This shows us that to manouvere a vehicle with nonholonomic constraints raises some challanges regarding path planning and vehicle movement in small and/or obstacle crowded spaces. The task of planning a path for nonholonomic systems was considered by the author during a project work reported in [3] , but the next step, making the vehicle follow the planned path, was not explored. This task is not straight forward, as the motion control can be quite complicated. Given a framework where a path planner gives a fesible, with respect to nonholonomic constraints, obstacle free path composed of straight lines and quadrants as described in Sec. 3.2.3. One of the challanges is also to find out whether the velocity along the path is constrained by the dynamics, also keeping in mind that the constraint forces must be satisfyed by the friction forces as the oposite will result in sliding. Also the nonholonomic constraints imply that there exists an uncontrollable subspace, with dimension of the nonholonomic constraints, for any control system kinematics or Lagrangian. This also complicates control tasks further. The same framework will be used to look at the system's closed loop solution and check if it would converge given an orbit. This is interesting because the nonholonomic constraints restricts the directions allowed to move in. Using transverse coordinates is a good way to analyze such a solution because they give a good indication of how far away from a desired orbit the system is.

Chapter 3

Theory and Previous Work

3.1 Equations of Motions for a Nonholonomic System

The calculation of the equations of motion (EOM) of a system can be done using the Euler-Lagrange [1]. To do this, the dynamics of the system first needs to be summarized to what is called the Lagrangian, $\mathcal{L} = T - V$, where T and V are the kinetic and the potential energy of a system, respectively. Further on the Euler-Lagrange equations are calculated using the following relation:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial q_i} = \Xi_i, \tag{3.1}$$

where $\boldsymbol{q} \in \mathbb{R}^n$ are the generalized coordinates, $\dot{\boldsymbol{q}} \in \mathbb{R}^n$ are the respective velocities and Ξ represents the non-conservative forces of the system such as damping and control inputs. Normally a general systems dynamics can be written on the form

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{u}, \tag{3.2}$$

where $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is a positive definite matrix of inertia, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ describes the Coriolis and the generalized centrifugal forces, $\boldsymbol{G}(\boldsymbol{q}) \in \mathbb{R}^n$ is where the forces due to gravitational terms are collected and $\boldsymbol{B}(\boldsymbol{q}) \in \mathbb{R}^{n\times m}$ describes the relationship between the states, $\boldsymbol{q}$, and the inputs, $\boldsymbol{u} \in \mathbb{R}^m$.

In [4] it is shown that introducing nonholonomic constraints yields the following system

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) &= \boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{\lambda} + \boldsymbol{B}(\boldsymbol{q})\boldsymbol{u}, \\ \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} &= \boldsymbol{0}, \end{aligned} \tag{3.3}$$

where $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{k\times n}$ is a constant full rank matrix, $k < n$, and the multiplier $\boldsymbol{\lambda}$ is a vector of amplitudes of the constraint forces needed to keep the k not integrable relations $\boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{0}$ invariant along the solutions. In other words constraint forces due to the nonholonomic constraints shows up as non-conservative forces and react to the system with a magnitude determined by a factor $\boldsymbol{\lambda}$.

3.2 Typical Control Methods of Nonholonomic Car-Like Systems

To be able to move an arbitrary system, a trajectory or path needs to be designed, and a controller needs to calculate some inputs for the system. In addition errors need to be calculated and corrected for. For car-like systems there have been designed some kinematic controllers, i.e controllers where the kinematics are directly controlled. As an example, for a car that would be the same as instantly changing the speed and ignore the fact that it would cause an extremely high acceleration. Some of these controllers were investigated in the project report [3] and shall be reviewed briefly in the following sections. Observations will show that the most popular trajectory generation strategies result from kinematic control laws, in which the path taken is unspecified. However, in general it is interesting to plan a feasible obstacle-free path and generate the trajectory including a stabilizing controller afterwards. One promising way of doing so is to parameterize the motion in terms of so-called virtual constraints that are thought of being enforced on the system. This approach allows to analytically introduce coordinates transversal to the flow of the dynamics and eventually analyze the resulting transverse dynamics for convergence to the prescribed orbit.

3.2.1 Trajectory Generation Using Kinematic Control

In Ch. 3 in [5] there is described a method using a feedback control to generate motion and a trajectory for a three-wheel differential type robot, which is represented in Fig. 3.1. The robot consists of two wheels at the rear end with

nonholonomic constraints, and a wheel in the front which can roll in every direction. The two inputs for the system are divided into one that controls the rotation around the back axis, ω, and another control that drives the robot at a straight line, i.e. constant speed at both back wheels, v.

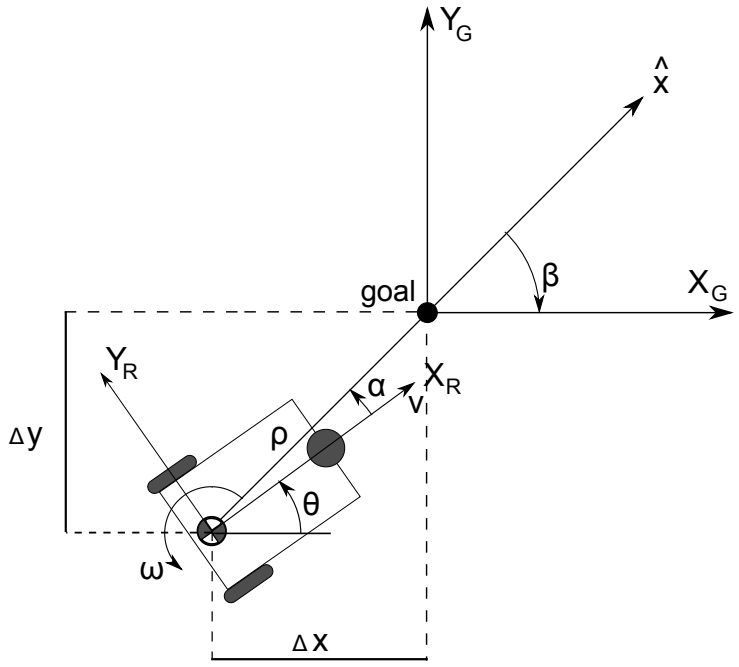


Figure 3.1: Robot reference frame compared to global frame with goal position for the robot as the origin.

The method consider a robot with an arbitrary position and orientation and a predefined goal position and orientation. Using trigonometry and polar coordinates the deviation from certain angle and position measurements are calculated and the inputs are found based on that. The equations used for calculating the errors and states are as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}^G = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \nu(t) \\ \omega(t) \end{bmatrix}, \tag{3.4}$$

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} \cos\alpha & 0 \\ \frac{1}{\rho}\sin\alpha & -1 \\ \frac{1}{\rho}\sin\alpha & 0 \end{bmatrix} \begin{bmatrix} \nu(t) \\ \omega(t) \end{bmatrix}. \tag{3.5}$$

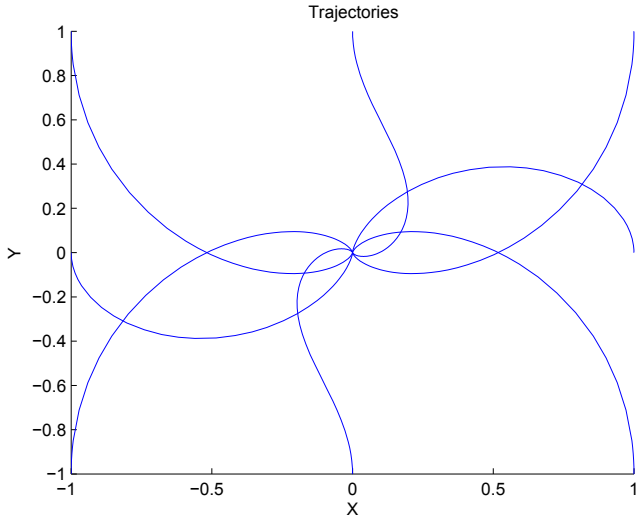


Figure 3.2: Trials with different initial positions and goal in origo.

Then a proportional controller is used to calculate the inputs:

$$\begin{aligned} \nu &= k_\rho \rho \\ \omega &= k_\alpha \alpha + k_\beta \beta \end{aligned} \tag{3.6}$$

Through some simulations the method was tested and the results are shown in Fig. 3.2. Here it is shown that the vehicle can be controlled to go from start to stop, but the controller still directly controls the speed which is only possible in simulations as inertia and forces are present in the real world. Another problem with the method is in the presence of obstacles which cannot be avoided in a simple way as the trajectory is planned automatically only based on position and goal coordinates. Due to these problems other methods were investigated.

3.2.2 Motion for a Kinematic Car Using One-Chained Form

The method described in Ch. 8 in [6] uses sine and cosine inputs to control the states of a car individually. By first controlling the directly controlled states to their goal position, integrally related sinusoids are used to control the rest of the states one by one. This method holds the property of not moving the

final position of the earlier controlled states because of the integrally related control inputs. In Fig. 3.3 a sketch of the car is shown, and the generalized coordinates are denoted $\boldsymbol{q} = [x,\ y,\ \theta,\ \phi]^T$. As mentioned the kinematic car is also a nonholonomic system and the similarities with the rolling disk can be seen.

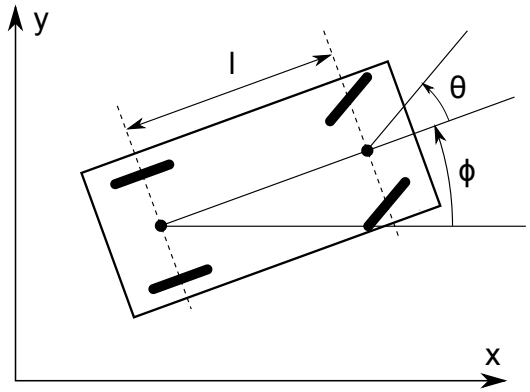


Figure 3.3: An overview of the kinematic car model.

The derivatives of the cars states, i.e the speed vectors and angular velocities, are shown to be

$$\dot{x} = \cos(\theta)u_1 \tag{3.7a}$$
$$\dot{y} = \sin(\theta)u_1 \tag{3.7b}$$
$$\dot{\theta} = \frac{1}{l}\tan(\phi)u_1 \tag{3.7c}$$
$$\dot{\phi} = u_2, \tag{3.7d}$$

where the x, y coordinate defines the position in the middle between the back wheels, ϕ is the angle between the x-axis and the car, and θ is describing the angle of the front wheels related to the cars heading. For modeling purposes the rear wheels are considered as one in the midpoint between them. The same applies for the front wheels. Chained form, which is used to design the controller in this case, is a way of representing a system such that it holds certain properties. This is explained in more detail in [7, 8, 6]. From these books it is also shown that many nonholonomical systems can be transformed

into a one-chained form that looks like this:

$$\begin{aligned}\dot{x}_1 &= v_1 \\ \dot{x}_2 &= v_2 \\ \dot{x}_3 &= x_2 v_1 \\ &\vdots \\ \dot{x}_n &= x_{n-1} v_1. \end{aligned} \tag{3.8}$$

The main characteristic of such a system is as mentioned above, it can be controlled one state at a time using sinusoids at integrally related frequencies, i.e $a\,sin(2\pi kt)$, where a and k are constants.

To get the system on a chained form, change of coordinates are done. The changes are done such that $z_1 = x, z_2 = \phi, z_3 = \sin(\theta), z_4 = y$ and the inputs $v_1 = \cos(\theta)u_1, v_2 = u_2$. This then yields

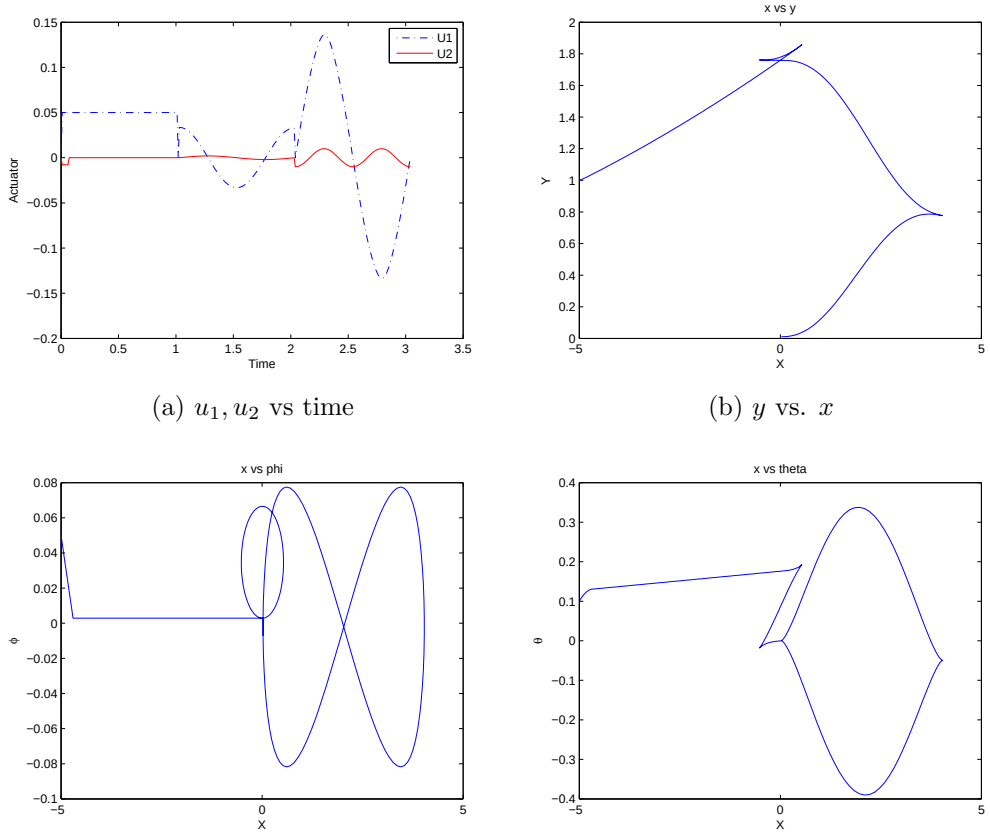
$$\dot{z}_1 = v_1 \tag{3.9a}$$
$$\dot{z}_2 = v_2 \tag{3.9b}$$
$$\dot{z}_3 = \frac{1}{l}\tan(z_2)v_1 \tag{3.9c}$$
$$\dot{z}_4 = \frac{z_3}{\sqrt{1 - z_3^2}}v_1. \tag{3.9d}$$

Several trials were done in the project [3], but only one is repeated here. The results are shown in Fig. 3.4.

From the results it is possible to see that the method works, but again the kinematics are not realistic. The inputs and thus the velocities are changed suddenly and would not be possible to implement on a real system. Second of all Fig. 3.4b shows how the transients are quite extensive and generally clumsy. It is also hard to calculate paths such that a vehicle can avoid and go around obstacles.

3.2.3 The RRT Approach

In addition to exploring some existing methods, another one was developed using path segments and a rapidly exploring random tree (RRT) approach. An RRT is a data structure and an algorithm developed by Steven M. Lavalle



(a) u_1, u_2 vs time

(b) y vs. x

(c) ϕ vs. x

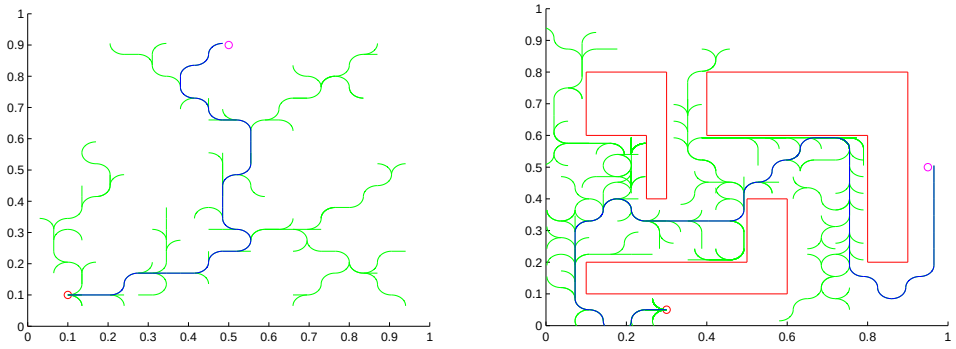
(d) θ vs. x

Figure 3.4: Run of trial 1 and the results.

[9] which uses random points in the configuration space to build itself. The RRT can efficiently explore a non-convex high dimensional space and by so making a connection from a given start point to a goal point. By randomly selecting points in the configuration space it searches through its vertices to find the one that minimizes the distance to the new one and connects it to the tree with a vertex. An advantage with RRT is that it does not tend to have a higher density around its starting point like many other algorithms, but instead it tends to reach for non discovered space. This makes it an efficient algorithm for searching through a space and find a path from an initial position to a goal. The algorithm for RRT can be found in [9].

Combining RRT with predefined path segments, typically straight lines and quadrants, a workspace can be explored with feasible motions for a vehicle. This way all paths found can be used to design a trajectory and implemented

for a vehicle to follow. The workspace was implemented with some walls acting as obstacles and the algorithms job was so find a feasible path that did not violate the obstacles. The path segments used was a predefined straight line, i.e drive forward, and predefined turns, 90^o turn left or right.



(a) No obstacles are present.

(b) With obstacles.

Figure 3.5: Two trials where the modified RRT algorithm finds feasible paths. No optimization was done.

Fig. 3.5 shows how the modified RRT algorithm manages to find feasible obstacle-free paths in an environment without and with the presence of obstacles. This is done using two classes of curves;

- line segments of different lengths
- quadrants of different radius

This method was tested more extensively in [3], and more results are shown there. Overall the algorithm performed well and the results led to this thesis which goal was to look into how to analyze and control a unicycle given a path.

3.3 Virtual Holonomic Constraints Approach

To use such a path as described in the previous chapter, the mechanical system needs to be analyzed and this will be done using some already developed methods described in [10, 4, 11]. The concept is in [11] called the virtual

holonomic constraints approach and refers to the use of virtual nonholonomic constraints to enforce a certain pre-calculated path.

3.3.1 Virtual Holonomic Constraints, VHC

Virtual holonomic constraints can be regarded as imaginary constraints, imposed through feedback control and not physically present in the system. The reason why they are called holonomic is because they can be described as a function of one of the states directly and they are not dependent on their derivatives as opposed to nonholonomic constraints, see Sec. 2.1. An example from [12] is reproduced in Fig. 3.6. This shows us how a 3 degrees of freedom (DOF) model, Fig. 3.6a, with help from two walls becomes a 1 DOF model, Fig. 3.6b. In this case it is an actual constraint, but if the virtual constraints are implemented correctly the effect is the same. From the figure it can be seen that the x-coordinate of the center of the

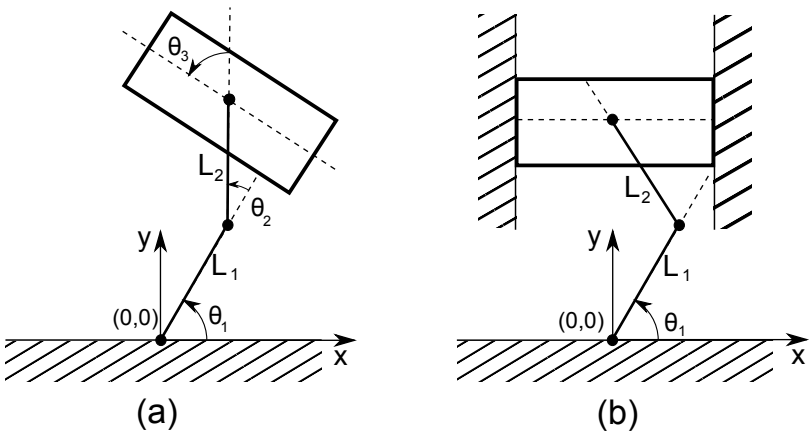


Figure 3.6: Example of how virtual holonomic constraints work.

load (i.e the black dot in the box) relative to the base can be described as $x_3 = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2)$. With the physical constraints it is easy to see that x_3 is forced to be zero. The virtual holonomic constraint in this case would be $L_1 \cos(\theta_1) = -L_2 \cos(\theta_1 + \theta_2)$. The constraints are also forcing the angle $\theta_{tot} = \theta_1 + \theta_2 + \theta_3$ to be a total of 180°, as the load is parallel with the horizontal axis. Assuming that there are controllable actuators in the joints(i.e the black dots in Fig. 3.6) it can by introducing control variables be

possible to set the angles, i.e a VHC where $\theta_1+\theta_2+\theta_3 = 180$. By saying that y_1 and y_2 are a controllers input and if it is further assumed that $y_1 = x_3$ and $y_2 = \theta_{tot} - 180°$ (the reference), it can be seen that the angles are controllable. This means that through feedback control one can asymptotically drive y_1 and y_2 to zero imposing the constraints virtually. Doing this the 3 DOF model from Fig. 3.6 will get the same kinematic behavior as the 1 DOF model.

It will later be shown how VHCs are used to impose certain behavior on the unicycle. It is also important that the constraints among the generalized coordinates are consistent with the nonholonomic constraints present in the system. But for a VHC to be enforced a reference needs to be set. In the papers listed in the intro to this section a synchronization function, called Φ is used as a reference. For each state of the system a reference is calculated based on a monotonically increasing parameter, $\theta(t)$ through this synchronization function.

Given a desired trajectory described by $\boldsymbol{q}_*(t) = \boldsymbol{\Phi}(\theta_*(t))$ it is now possible to use VHC and arrange the development of the states such that

$$\boldsymbol{q}_* = \begin{bmatrix} q_{1*} \\ q_{2*} \\ \vdots \\ q_{n*} \end{bmatrix} = \begin{bmatrix} \Phi_1(\theta) \\ \Phi_2(\theta) \\ \vdots \\ \Phi_n(\theta) \end{bmatrix} = \boldsymbol{\Phi}(\theta), \quad \theta = \theta_*(t), \quad t \in [0, T], \tag{3.10}$$

where the star represent the desired case. The desired trajectory will also have the property of being an orbit, i.e. it will finish at the point it starts at. This way stability of the motion can easily be checked as the time can evolve for infinite.

3.3.2 Transverse Linearization

Once the VHCs are defined it is easy to calculate an error based on the states and the desired motion. Following the method described in [11] the errors, $\boldsymbol{z}$, can be calculated as

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} - \begin{bmatrix} q_{*1} \\ q_{*2} \\ \vdots \\ q_{*n} \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} - \begin{bmatrix} \Phi_1(\theta_*(t)) \\ \Phi_2(\theta_*(t)) \\ \vdots \\ \Phi_n(\theta_*(t)) \end{bmatrix} \tag{3.11}$$

in addition to θ, which means that there are now $n+1$ coordinates. Given at least one excessive coordinate, one of those can be expressed as a smooth scalar function of the other ones, call the function $h(z,\theta)$, and without loss of generality it can also be assumed that this is z_n such that the new coordinate set becomes

$$\begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_{n-1} \\ q_n \end{bmatrix} = \begin{bmatrix} \Phi_1(\theta) \\ \Phi_2(\theta) \\ \vdots \\ \Phi_{n-1}(\theta) \\ \Phi_n(\theta) \end{bmatrix} + \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{n-1} \\ h(z,\theta) \end{bmatrix}. \tag{3.12}$$

The generalized velocities of the above mentioned system can be calculated as[1]

$$\dot{\boldsymbol{q}} = \boldsymbol{L}(\theta, \boldsymbol{z}) \begin{bmatrix} \dot{\boldsymbol{z}} \\ \dot{\theta} \end{bmatrix} \tag{3.13}$$

where

$$\boldsymbol{L}(\theta, \boldsymbol{z}) = \begin{bmatrix} \boldsymbol{I}_{(n-1)\times(n-1)} & \boldsymbol{0}_{(n-1)\times 1} \\ \frac{\partial h}{\partial z} & \frac{\partial h}{\partial \theta} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_{n\times(n-1)} & \begin{matrix} \Phi_1'(\theta) \\ \vdots \\ \Phi_n'(\theta) \end{matrix} \end{bmatrix}. \tag{3.14}$$

Eq. (3.12) and (3.13) now describe a coordinate transformation from $(\boldsymbol{q}, \dot{\boldsymbol{q}}) \rightarrow (\boldsymbol{z}, \dot{\boldsymbol{z}}, \theta, \dot{\theta})$ which is important in order to derive the $2n-1$ coordinates transversal to the prescribed orbit, that is transversal to the desired flow of the dynamics. Since $[\boldsymbol{z};\ \dot{\boldsymbol{z}}]^T$ already are good candidates, the only transformation that needs to be worked out is the one for $[\theta, \dot{\theta}]$ that yields the last transverse coordinate, merely called $I(\theta, \dot{\theta})$. This scalar quantity must measure the orthogonal distance to the desired curve in the phase plane $[\theta, \dot{\theta}]$. The coordinates transversal to the orbit is given as

$$\boldsymbol{x}_\perp = \begin{bmatrix} I \\ \boldsymbol{z} \\ \dot{\boldsymbol{z}} \end{bmatrix}, \tag{3.15}$$

and it can be seen that the vector $\boldsymbol{x}_\perp \in \mathbb{R}^{2n-1}$. The transverse coordinates can now be used to compute how far from a desired trajectory a current state is and the dynamics of the transverse coordinates can be used to calculate the system's behavior in the vicinity, and eventually to design a controller

[1] Note that z now has dimension n-1 and that $\mathbf{I}$ means the identity matrix

that makes the trajectories converge to the desired orbit. To be able to do that, the system dynamics in transverse coordinates needs to be linearized and obtain the following linear time-varying control system, the so-called transverse linearization [10]:

$$\frac{d}{d\tau}\begin{bmatrix}\Delta I\\ \Delta \boldsymbol{z}\\ \Delta \dot{\boldsymbol{z}}\end{bmatrix} = \mathcal{A}(\tau)\begin{bmatrix}\Delta I\\ \Delta \boldsymbol{z}\\ \Delta \dot{\boldsymbol{z}}\end{bmatrix} + \mathcal{B}(\tau)\Delta V(\tau). \tag{3.16}$$

Here the input variable $\Delta V(\tau)$ is the result of a transformation from u to $\Delta V(\tau)$ such that $\Delta V(\tau)$ is equal to zero along the desired orbit.

3.3.3 Orbital Stabilization of a Periodic Target Motion

Until now a motion for a system has been set, the error has been calculated in transverse coordinates and the dynamics of the error has been found using transverse linearization. To be able to stabilize a motion over time, it is for now assumed that the motion generated is periodic, i.e. $q_*(t) = q_*(t+T)$, where T is the time to complete one period of the motion. Then to stabilize a periodic motion, a periodic controller with the same time dependency needs to be designed. One way could be to use feedback control and design the input such that $\Delta V(\tau) = K(\tau)\begin{bmatrix}\Delta I\\ \Delta \boldsymbol{z}\\ \Delta \dot{\boldsymbol{z}}\end{bmatrix}$, where $K(\tau) = K(\tau + T)$

3.4 Controller Design Challenges

3.4.1 Controllability

Given the system

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{3.17}$$

where the matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and the matrix $\boldsymbol{B} \in \mathbb{R}^{n\times p}$ describes the system dynamics. It is often interesting to know if the system is controllable. According to [13] a system, is said to be controllable if for any initial state $\boldsymbol{x}_0$ and any final state $\boldsymbol{x}_1$, there exists an input, $\boldsymbol{u}$, that transfers $\boldsymbol{x}_0$ to $\boldsymbol{x}_1$ in finite time.

Controllability Matrix

For a linear time invariant (LTI) system the controllability of the system can be checked by looking at the rank of the controllability matrix

$$\mathcal{C} = \begin{bmatrix} B & AB & A^2B & \cdots & A^nB \end{bmatrix}. \tag{3.18}$$

If the rank is of dimension n, it is said to be controllable, as every state can be controlled. Otherwise it is said to be uncontrollable, and the uncontrollable states needs to be stable for the system to be stabilizable. Stability is checked by looking at the eigenvalues of the closed loop system, i.e $eig(\tilde{\boldsymbol{A}}) = eig(\boldsymbol{A} - \boldsymbol{BK})$ if the controlled input $\boldsymbol{u} = \boldsymbol{Kx}$. If the eigenvalues have negative or zero real parts, and all poles with zero real value are simple roots, the system is said to be marginally stable. It is said to be asymptotically stable if the real part of the poles all are negative. For a discrete-time case, $\boldsymbol{x}[k+1] = \tilde{\boldsymbol{A}}[k]$, the eigenvalues need to have a magnitude less than or equal to one to be marginally stable and less than one to be asymptotically stable. A marginally stable system is a system that, if given an finite impulse as input it will not blow up and give unbounded outputs. An asymptotically stable system will in the case of every finite initial state have a bounded output and approach 0 as time approaches infinity.

The Controllability Gramian

For a system that is not time invariant, the above mentioned method may not work, and other approaches exists, one of them called the controllability gramian. The gramian, $\boldsymbol{W_c}(t)$, consist of an integral over the state transition matrix, $\boldsymbol{\Phi}(t, t_0)$ [2], and the state input matrix, $\boldsymbol{B}(t)$ as stated at [14]. Given

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}(t)\boldsymbol{x}(t) + \boldsymbol{B}(t)\boldsymbol{u}(t), \tag{3.19}$$

$$\boldsymbol{W_c}(t) = \int_{t_0}^{t} \boldsymbol{\Phi}(t_0, \tau)\boldsymbol{B}(\tau)\boldsymbol{B}^T(\tau)\boldsymbol{\Phi}^T(t_0, \tau)\, d\tau \tag{3.20}$$

where the state transition matrix, $\boldsymbol{\Phi}(t, t_0)$, is calculated such that

$$\boldsymbol{\Phi}(t_0, t_0) = I \quad , \quad \frac{d\boldsymbol{\Phi}(t, t_0)}{dt} = \boldsymbol{A}(t)\Phi(t_0, t). \tag{3.21}$$

The most useful property of the state transition matrix is the relation

$$\boldsymbol{x}(t) = \boldsymbol{\Phi}(t_0, t)\boldsymbol{x}(t_0), \tag{3.22}$$

[2] Note that this is not the same as the synchronization function from Sec. 3.3.1

which means that the evolution of any initial condition can be calculated in an easy way. For a periodic system it is of interest to see if it is stable and controllable over a period as the subsequent periods also will be of the same condition. For the periodic system the controllability gramian needs to be non-singular over a period, i.e the eigenvalues are different from zero.

Chapter 4

System Analysis

This part of the report will focus on the unicycle system and analyze mainly the four degree of freedom (DOF) case, see Fig. 4.1, but a comparison to the five DOF case reveal interesting features at the end.

4.1 EOM for the Unicycle

Continuing with the unicycle example from Sec. 2.2 it can be seen that the general coordinates are $\boldsymbol{q} = [x,\ y,\ \phi,\ \theta]^T$, which represents position in the $x - y$ plane, the rotational angle of the unicycle with respect to a fixed point on the coin and the angular turning angle with respect to the x-axis respectively, see Fig. 4.1.

The kinetic and potential energy can be expressed like this:

$$T = \frac{1}{2}mv^2 + \frac{1}{2}I_\phi\dot{\phi}^2 + \frac{1}{2}J\dot{\theta}^2 = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) + \frac{1}{2}I_\phi\dot{\phi}^2 + \frac{1}{2}I_\theta\dot{\theta}^2\ , \quad V = 0. \tag{4.1}$$

The velocity in x and y direction will be directly related to the angular wheel velocity, $\dot{\phi}$, along the heading angle, θ, and the rolling unicycle will also be subject to the following non-holonomic constraints [15] :

$$\dot{x} - r_c\dot{\phi}\cos\theta = 0 \tag{4.2}$$

$$\dot{y} - r_c\dot{\phi}\sin\theta = 0, \tag{4.3}$$

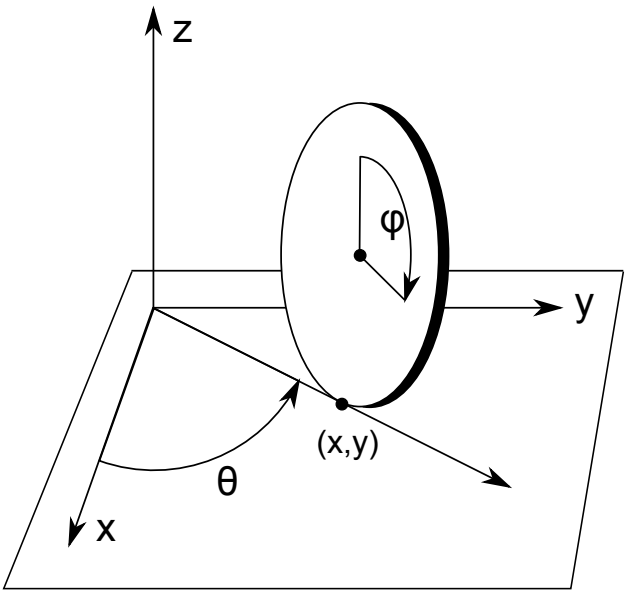


Figure 4.1: The coin described with four degrees of freedom

which can be written on a matrix form as

$$\begin{bmatrix} 1 & 0 & -r_c \cos\theta & 0 \\ 0 & 1 & -r_c \sin\theta & 0 \end{bmatrix} \dot{\boldsymbol{q}} = 0, \tag{4.4}$$

where the matrix can be recognized as J(q). Using the formula provided in Sec. 3.1 the Lagrangian is calculated

$$\mathcal{L} = T - V = \frac{1}{2} m(\dot{x}^2 + \dot{y}^2) + \frac{1}{2} I_\phi \dot{\phi}^2 + \frac{1}{2} I_\theta \dot{\theta}^2 - 0. \tag{4.5}$$

Applying Eq. (3.1) to the Lagrangian:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right) - \frac{\partial \mathcal{L}}{\partial x} = m\ddot{x} - 0. \tag{4.6a}$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{y}}\right) - \frac{\partial \mathcal{L}}{\partial y} = m\ddot{y} - 0. \tag{4.6b}$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\phi}}\right) - \frac{\partial \mathcal{L}}{\partial \phi} = I_\phi \ddot{\phi} - 0. \tag{4.6c}$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}}\right) - \frac{\partial \mathcal{L}}{\partial \theta} = I_\theta \ddot{\theta} - 0. \tag{4.6d}$$

Eq. (3.3) helps identify the following relations:

$$\underbrace{\begin{bmatrix} m & 0 & 0 & 0 \\ 0 & m & 0 & 0 \\ 0 & 0 & I_\phi & 0 \\ 0 & 0 & 0 & I_\theta \end{bmatrix}}_{\boldsymbol{M(q)}} \underbrace{\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\phi} \\ \ddot{\theta} \end{bmatrix}}_{\ddot{\boldsymbol{q}}} + \underbrace{0}_{\boldsymbol{C(q,\dot{q})}} \underbrace{\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \\ \dot{\theta} \end{bmatrix}}_{\dot{\boldsymbol{q}}} + \underbrace{0}_{\boldsymbol{G(q)}} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -r_c\cos\theta & -r_c\sin\theta \\ 0 & 0 \end{bmatrix}}_{\boldsymbol{J(q)}^T} \underbrace{\begin{bmatrix} \lambda_{c1} \\ \lambda_{c2} \end{bmatrix}}_{\boldsymbol{\lambda}} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\boldsymbol{B(q)}} \underbrace{\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}_{\boldsymbol{u}}, \tag{4.7}$$

which describes the full dynamics of the four DOF case of the unicycle. Here the inputs are given by $\boldsymbol{u}$ which controls the angular acceleration, $\ddot{\phi}$ and $\ddot{\theta}$. Further on it can be seen that the matrix containing the nonholonomic constraints is multiplied with a vector $\boldsymbol{\lambda}$ as explained in Sec. 3.1. Following method 11.4 in [4] it is possible to make the following relation:

$$\frac{d}{dt}\left[\boldsymbol{J(q)\dot{q}}\right] = \boldsymbol{\dot{J}(q)\dot{q} + J(q)\ddot{q}} = \boldsymbol{0}, \tag{4.8}$$

which in this case gives

$$\frac{d}{dt}\left[\boldsymbol{J(q)\dot{q}}\right] = \begin{bmatrix} 0 & 0 & r_c\sin(\theta)\,\dot{\theta} & 0 \\ 0 & 0 & -r_c\cos(\theta)\,\dot{\theta} & 0 \end{bmatrix} \boldsymbol{\dot{q}} + \begin{bmatrix} 1 & 0 & -r_c\cos\theta & 0 \\ 0 & 1 & -r_c\sin\theta & 0 \end{bmatrix} \boldsymbol{\ddot{q}} = \boldsymbol{0}. \tag{4.9}$$

For the convenience of the readers matrix notation will be simplified, i.e $\boldsymbol{C(q,\dot{q})\dot{q}} \rightarrow \boldsymbol{C\dot{q}}$. Following through with the example, since $\boldsymbol{M}$ is non-singular Eq. (4.7) can be rearranged such that

$$\boldsymbol{\ddot{q}} = -\boldsymbol{M}^{-1}(\boldsymbol{C\dot{q} + G - J^T\lambda - Bu}). \tag{4.10}$$

The idea is now to replace the reaction forces λ, and from Eq. (4.8) it can be seen that

$$\begin{aligned} \boldsymbol{\dot{J}\dot{q}} &= -\boldsymbol{J\ddot{q}} && (4.11)\\ &= \boldsymbol{JM}^{-1}(\boldsymbol{C\dot{q} + G - J^T\lambda - Bu}) && (4.12)\\ \boldsymbol{JM}^{-1}\boldsymbol{J}^T\lambda + \boldsymbol{\dot{J}\dot{q}} &= \boldsymbol{JM}^{-1}(\boldsymbol{C\dot{q} + G - Bu}) && (4.13)\\ \lambda &= \underbrace{\underbrace{(\boldsymbol{JM}^{-1}\boldsymbol{J}^T)^{-1}}_{\boldsymbol{J_M}}(\boldsymbol{JM}^{-1}(\boldsymbol{C\dot{q} + G - Bu}) - \boldsymbol{\dot{J}\dot{q}})}_{\boldsymbol{J_*}}. && (4.14) \end{aligned}$$

Please note that since $\boldsymbol{J}$ is not square and can not be inverted, the notation $\boldsymbol{J_M}$ will be used to simplify. Now, this can be combined with Eq. (4.7), hence

$$\boldsymbol{M\ddot{q} + C\dot{q} + G} = \boldsymbol{J}^T\boldsymbol{J}_*(\boldsymbol{C\dot{q} + G - Bu}) - \boldsymbol{J}^T\boldsymbol{J}_M\boldsymbol{\dot{J}\dot{q}} + \boldsymbol{Bu} \tag{4.15}$$

which yields

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \underbrace{(\boldsymbol{C} + \boldsymbol{J}^T\boldsymbol{J}_M\dot{\boldsymbol{J}} - \boldsymbol{J}^T\boldsymbol{J}_*\boldsymbol{C})}_{\boldsymbol{C_\lambda}}\dot{\boldsymbol{q}} + \underbrace{(\boldsymbol{I} - \boldsymbol{J}^T\boldsymbol{J}_*)\boldsymbol{G}}_{\boldsymbol{G_\lambda}} = \underbrace{(\boldsymbol{I} - \boldsymbol{J}^T\boldsymbol{J}_*)\boldsymbol{B}}_{\boldsymbol{B_\lambda}}\boldsymbol{u}, \tag{4.16}$$

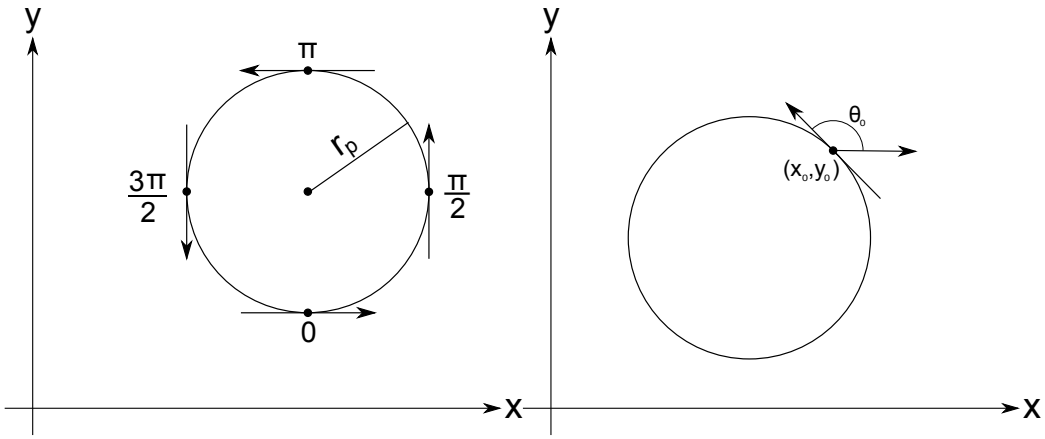
which is a rewritten form of Eq.(1) in [4] where λ is eliminated. This implies that any magnitude of constraint forces can be assumed feasible since this equation is true provided that the nonholonomic constraint is held.

4.2 Parametrization of Motion by VHC

To follow the approach described in Sec. 3.3, a synchronization function needs to be designed. To do this a monotonically increasing parameter needs to be chosen. From the modified RRT algorithm described in Sec. 3.2.3 it would be preferable if the trajectory described by the synchronization function would be a straight line or a quadrant. In Sec. 3.3.3 it was stated that the motion needed to be periodic in order to be stabilized over time. Therefore a full circle will be used as a synchronization function as a quadrant is insufficient to create a periodic motion. Given a circle as represented in Fig. 4.2a, the object will be to try to represent a segment given one parameter. Preferably this parameter is one of the generalized coordinates and given the motion being a circle and the parameter being monotonically increasing it is obvious that this parameter must be either ϕ or θ. As ϕ doesn't say anything about where on a circle an object is, θ seems like a good choice. In Fig. 4.2 four configurations for θ are shown, and given an arbitrary but known initial configuration, $q_0 = [x_0,\ y_0,\ \theta_0,\ \phi_0]$, one can track the configuration of the coin all over the path as long as one knows the radius of the unicycle, r_c, and the path, r_p, and θ.

Given a circle the synchronization function looks like:

$$\boldsymbol{q_*} = \begin{bmatrix} x_* \\ y_* \\ \phi_* \\ \theta_* \end{bmatrix} = \boldsymbol{\Phi}(\theta) = \begin{bmatrix} x_0 + r_p(-\sin(\theta_0) + \sin(\theta)) \\ y_0 + r_p(\cos(\theta_0) - \cos(\theta)) \\ \phi_0 + \frac{r_p}{r_c}(\theta - \theta_0) \\ \theta \end{bmatrix} \tag{4.17}$$



(a) Configuration with different θ (b) Configuration example

Figure 4.2: Configurations

which can be differentiated as

$$\dot{\boldsymbol{q}}_* = \dot{\boldsymbol{\Phi}}(\theta) = \boldsymbol{\Phi}'(\theta)\dot{\theta} = \begin{bmatrix} r_p\cos(\theta) \\ r_p\sin(\theta) \\ \frac{r_p}{r_c} \\ 1 \end{bmatrix} \dot{\theta} \tag{4.18}$$

and the double derivative becomes

$$\ddot{\boldsymbol{q}}_* = \ddot{\boldsymbol{\Phi}}(\theta) = \boldsymbol{\Phi}''(\theta)\dot{\theta}^2 + \boldsymbol{\Phi}'(\theta)\ddot{\theta} = \begin{bmatrix} -r_p\sin(\theta) \\ r_p\cos(\theta) \\ 0 \\ 0 \end{bmatrix} \dot{\theta}^2 + \begin{bmatrix} r_p\cos(\theta) \\ r_p\sin(\theta) \\ \frac{r_p}{r_c} \\ 1 \end{bmatrix} \ddot{\theta} \tag{4.19}$$

where $\theta = \theta_*(t), \quad \dot{\theta} = \dot{\theta}_*(t), \quad \ddot{\theta} = \ddot{\theta}_*(t), \quad t \in [0, T]$

4.2.1 Calculation of the α, β and γ Functions

To find a pair, θ and $\dot{\theta}$, [11, 10, 4] calculates the dynamics of the underactuated states of the system through a set of equations called the " alpha, beta gamma" pairs. To do this first of all a matrix, $\boldsymbol{B}^{\perp}$, needs to be found. The main property of $\boldsymbol{B}^{\perp}$ is that it annihilates $\boldsymbol{B}_{\boldsymbol{\lambda}}$, i.e $\boldsymbol{B}^{\perp}\boldsymbol{B}_{\boldsymbol{\lambda}} = \boldsymbol{0}$. Eq. (4.16) showed us that

$$\boldsymbol{B}_{\boldsymbol{\lambda}} = (\boldsymbol{I} - \boldsymbol{J}^{\boldsymbol{T}}\boldsymbol{J}_*)\boldsymbol{B}, \tag{4.20}$$

where

$$\boldsymbol{J}_* = (\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{J}^T)^{-1}\boldsymbol{J}\boldsymbol{M}^{-1} \tag{4.21}$$

The first part of $\boldsymbol{J}_*$ is derived as

$$\begin{aligned}\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{J}^T &= \begin{bmatrix}1 & 0 & -r_c\cos\theta & 0\\ 0 & 1 & -r_c\sin\theta & 0\end{bmatrix}\begin{bmatrix}\frac{1}{m} & 0 & 0 & 0\\ 0 & \frac{1}{m} & 0 & 0\\ 0 & 0 & \frac{1}{I_\phi} & 0\\ 0 & 0 & 0 & \frac{1}{I_\theta}\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 1\\ -r_c\cos\theta & -r_c\sin\theta\\ 0 & 0\end{bmatrix}\\ &= \begin{bmatrix}\frac{1}{m}+\frac{r_c^2}{I_\phi}\cos^2\theta & \frac{r_c^2}{I_\phi}\cos\theta\sin\theta\\ \frac{r_c^2}{I_\phi}\cos\theta\sin\theta & \frac{1}{m}+\frac{r_c^2}{I_\phi}\sin^2\theta\end{bmatrix}\end{aligned} \tag{4.22}$$

Following through with the derivations

$$\boldsymbol{B}_{\boldsymbol{\lambda}} = (\boldsymbol{I} - \boldsymbol{J}^T\boldsymbol{J}_*)\boldsymbol{B} \tag{4.23}$$

where

$$\boldsymbol{B}_{\boldsymbol{\lambda}} = \begin{bmatrix}B_{\lambda\ 1,1} & 0\\ B_{\lambda\ 2,1} & 0\\ B_{\lambda\ 3,1} & 0\\ 0 & 1\end{bmatrix} \tag{4.24}$$

$$B_{\lambda\ 1,1} = \frac{mr_c cos(\theta)}{mr_c^2 + I_\phi}, \quad B_{\lambda\ 2,1} = \frac{mr_c sin(\theta)}{mr_c^2 + I_\phi}, \quad B_{\lambda\ 2,1} = \frac{I_\phi}{mr_c^2 + I_\phi}$$

To be able to find an expression that looks like Eq.(11.10) in [4]

$$\alpha(\theta)\ddot{\theta} + \beta(\theta)\dot{\theta}^2 + \gamma(\theta) = 0, \tag{4.25}$$

the matrix $\boldsymbol{B}^\perp$ that annihilates $\boldsymbol{B}_{\boldsymbol{\lambda}}$ needs to be found. This is easily done in MATLAB using the command `null(B)` which gives

$$\boldsymbol{B}^\perp = \begin{bmatrix}B_{1,1}^\perp & 1 & 0 & 0\\ B_{2,1}^\perp & 0 & 1 & 0\end{bmatrix} \tag{4.26}$$

$$B_{1,1}^\perp = \frac{-B_{\lambda\ 2,1}}{B_{\lambda\ 1,1}} = \frac{-\sin(\theta)}{\cos(\theta)}, \quad B_{2,1}^\perp = \frac{-B_{\lambda\ 3,1}}{B_{\lambda\ 1,1}} = \frac{-I_\phi}{mr_c\cos(\theta)}$$

$$\boldsymbol{B}^\perp\boldsymbol{B}_{\boldsymbol{\lambda}} = \begin{bmatrix}-\frac{B_{\lambda\ 2,1}}{B_{\lambda\ 1,1}} & 1 & 0 & 0\\ -\frac{B_{\lambda\ 3,1}}{B_{\lambda\ 1,1}} & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}B_{\lambda\ 1,1} & 0\\ B_{\lambda\ 2,1} & 0\\ B_{\lambda\ 3,1} & 0\\ 0 & 1\end{bmatrix} = \boldsymbol{0}, \tag{4.27}$$

which seems to be a solution as the matrix product is $\mathbf{0}$.
Premultiplying the whole system with $\boldsymbol{B}^{\perp}$ gives

$$\boldsymbol{B}^{\perp}\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{B}^{\perp}\boldsymbol{C}_{\boldsymbol{\lambda}}\dot{\boldsymbol{q}} + \boldsymbol{B}^{\perp}\boldsymbol{G}_{\boldsymbol{\lambda}} = \underbrace{\boldsymbol{B}^{\perp}\boldsymbol{B}_{\boldsymbol{\lambda}}}_{=\mathbf{0}}\boldsymbol{u} \tag{4.28}$$

$$\boldsymbol{B}^{\perp}\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{B}^{\perp}(\underbrace{\boldsymbol{C}}_{=\mathbf{0}} + \boldsymbol{J}^{\boldsymbol{T}}\boldsymbol{J}_{\boldsymbol{M}}\dot{\boldsymbol{J}} - \boldsymbol{J}^{\boldsymbol{T}}\boldsymbol{J}_{*}\underbrace{\boldsymbol{C}}_{=\mathbf{0}})\dot{\boldsymbol{q}} + (\boldsymbol{I} - \boldsymbol{J}^{\boldsymbol{T}}\boldsymbol{J}_{*})\boldsymbol{B}^{\perp}\underbrace{\boldsymbol{G}}_{=\mathbf{0}} = \mathbf{0} \tag{4.29}$$

$$\boldsymbol{B}^{\perp}\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{B}^{\perp}\boldsymbol{J}^{\boldsymbol{T}}\boldsymbol{J}_{\boldsymbol{M}}\underbrace{\dot{\boldsymbol{J}}}_{\sum_i \frac{\partial \boldsymbol{J}}{\partial q_i}\dot{q}_i}\dot{\boldsymbol{q}} = 0. \tag{4.30}$$

Since $\frac{\partial \boldsymbol{J}}{\partial x} = \frac{\partial \boldsymbol{J}}{\partial y} = \frac{\partial \boldsymbol{J}}{\partial \phi} = 0$ only $\frac{\partial \boldsymbol{J}}{\partial \theta}$ makes a contribution. Then Eq. (4.17), (4.18) and (4.19) is used to get an expression like Eq. (4.25). To see the whole picture old notation is used:

$$\boldsymbol{B}^{\perp}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{B}^{\perp}(\boldsymbol{q})\boldsymbol{J}^{T}(\boldsymbol{q})\boldsymbol{J}_{M}(\boldsymbol{q})\frac{\partial \boldsymbol{J}(\boldsymbol{q})}{\partial \theta}\dot{\theta}\,\dot{\boldsymbol{q}} = \mathbf{0}. \tag{4.31}$$

In this case $\boldsymbol{B}^{\perp}(\boldsymbol{q})$, $\boldsymbol{B}_{\boldsymbol{\lambda}}(\boldsymbol{q})$, $\boldsymbol{J}(\boldsymbol{q})$, $\boldsymbol{J}_{\boldsymbol{M}}(\boldsymbol{q})$ are only dependent on the variable $q_4 = \theta$ and not the other ones. Meanwhile $\boldsymbol{M}(\boldsymbol{q})$ is only a matrix constructed by constants such that the equations can be continued as

$$\boldsymbol{B}^{\perp}(\boldsymbol{\theta})\boldsymbol{M}\left(\boldsymbol{\Phi}''(\theta)\dot{\theta}^2 + \boldsymbol{\Phi}'(\theta)\ddot{\theta}\right) + \boldsymbol{B}^{\perp}(\theta)\boldsymbol{J}^{T}(\theta)\boldsymbol{J}_{M}(\theta)\frac{\partial \boldsymbol{J}(\theta)}{\partial \theta}\dot{\theta}\,\boldsymbol{\Phi}'(\theta)\dot{\theta} = \mathbf{0}, \tag{4.32}$$

which can be rearranged and recognized as

$$\underbrace{\boldsymbol{B}^{\perp}(\theta)\boldsymbol{M}\,\boldsymbol{\Phi}'(\theta)}_{\boldsymbol{\alpha}(\theta)}\ddot{\theta} + \underbrace{\boldsymbol{B}^{\perp}(\theta)\Big(\boldsymbol{M}\,\boldsymbol{\Phi}''(\theta) + \boldsymbol{J}^{T}(\theta)\boldsymbol{J}_{M}(\theta)\frac{\partial \boldsymbol{J}(\theta)}{\partial \theta}\,\boldsymbol{\Phi}'(\theta)\Big)}_{\boldsymbol{\beta}(\theta)}\dot{\theta}^2 = \mathbf{0}. \tag{4.33}$$

It can be seen that there is no $\gamma(\theta)$ part of the equation as a result of the gravity term in Eq. (4.7) being zero.

Finding the explicit expression of Eq. (4.33):

$$\boldsymbol{\alpha}(\theta) = \boldsymbol{B}^{\perp}(\theta)\boldsymbol{M}\,\boldsymbol{\Phi}'(\theta) = \boldsymbol{B}^{\perp}(\theta)\begin{bmatrix} mr_p\cos(\theta) \\ mr_p\sin(\theta) \\ I_\phi\frac{r_p}{r_c} \\ I_\theta \end{bmatrix} \tag{4.34}$$

$$= \begin{bmatrix} -\frac{\sin(\theta)}{\cos(\theta)} & 1 & 0 & 0 \\ -\frac{I_\phi}{mr_c\cos(\theta)} & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} mr_p\cos(\theta) \\ mr_p\sin(\theta) \\ I_\phi\frac{r_p}{r_c} \\ I_\theta \end{bmatrix} = \mathbf{0} \tag{4.35}$$

$$\boldsymbol{\beta}(\theta) = \boldsymbol{B}^{\perp}(\theta)\Big(\boldsymbol{M}\,\boldsymbol{\Phi}''(\theta) + \boldsymbol{J}^T(\theta)\boldsymbol{J}_M(\theta)\frac{\partial \boldsymbol{J}(\theta)}{\partial\theta}\,\boldsymbol{\Phi}'(\theta)\Big)$$

$$= \boldsymbol{B}^{\perp}\left(\begin{bmatrix} -mr_p\sin(\theta) \\ mr_p\cos(\theta) \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} mr_p\sin(\theta) \\ -mr_p\cos(\theta) \\ 0 \\ 0 \end{bmatrix}\right) = \boldsymbol{B}^{\perp}\cdot\mathbf{0}, \tag{4.36}$$

from which it can be seen that both $\alpha(\theta)$ and $\beta(\theta)$ always will be equal to zero for a circular path. Other $\boldsymbol{B}^{\perp}(\theta)$ has also been tried but they all yields the same answer. This observation tells that it is no restrictions on how the pair $\theta, \dot{\theta}$ can be chosen, that is, along these circular paths trajectories can be arbitrarily but smoothly shaped.

4.3 Transverse Linearization and Orbital Stabilization

Following the method from Sec. 3.3 the following calculations needs to be done in order to analyze the motion of the system.

4.3.1 Introducing the New Coordinates

As the motion already has been designed, virtual holonomic constraints will be used as in Eq. (3.10) to introduce a new set of coordinates as:

$$x = z_1 + \Phi_1(\theta) = z_1 + (x_0 + r_p(-\sin(\theta_0) + \sin(\theta))) \tag{4.37a}$$
$$y = z_2 + \Phi_2(\theta) = z_2 + (y_0 + r_p(\cos(\theta_0) - \cos(\theta))) \tag{4.37b}$$
$$\phi = z_3 + \Phi_3(\theta) = z_3 + \left(\phi_0 + \frac{r_p}{r_c}(\theta - \theta_0)\right) \tag{4.37c}$$
$$\theta = z_4 + \Phi_4(\theta) = 0 + \theta = \theta \ , \tag{4.37d}$$

which is similar to Eq. (3.12). The goal is now to produce transverse coordinates $\boldsymbol{z}, \dot{\boldsymbol{z}}$ and $I(\theta, \dot{\theta})$ and linearize the resulting system dynamics as in Eq. (3.16) :

$$\frac{d}{d\tau}\begin{bmatrix}\Delta I \\ \Delta \boldsymbol{z} \\ \Delta \dot{\boldsymbol{z}}\end{bmatrix} = \mathcal{A}(\tau)\begin{bmatrix}\Delta I \\ \Delta \boldsymbol{z} \\ \Delta \dot{\boldsymbol{z}}\end{bmatrix} + \mathcal{B}(\tau)\Delta V(\tau).$$

First of all the dynamics of $\ddot{\boldsymbol{z}}$ needs to be found and to do that, the first step is to differentiate Eq. (4.37):

$$\dot{\boldsymbol{q}} = \begin{bmatrix}\dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4\end{bmatrix} = \begin{bmatrix}\dot{z_1} + \Phi_1'(\theta)\dot{\theta} \\ \dot{z_2} + \Phi_2'(\theta)\dot{\theta} \\ \dot{z_3} + \Phi_3'(\theta)\dot{\theta} \\ 0 + \Phi_4'(\theta)\dot{\theta}\end{bmatrix} = \boldsymbol{L}(\theta, \boldsymbol{z})\begin{bmatrix}\dot{\boldsymbol{z}} \\ \dot{\theta}\end{bmatrix}, \tag{4.38}$$

where

$$\boldsymbol{L}(\theta, \boldsymbol{z}) = \begin{bmatrix}\mathbf{I}_{3x3} & \mathbf{0}_{3x1} \\ \mathbf{0}_{1x4}\end{bmatrix} + \begin{bmatrix}\mathbf{0}_{4x3} & \begin{matrix}\Phi_1'(\theta) \\ \vdots \\ \Phi_4'(\theta)\end{matrix}\end{bmatrix}. \tag{4.39}$$

The double derivative is found:

$$\ddot{\boldsymbol{q}} = \dot{\boldsymbol{L}}(\theta, \mathbf{z})\begin{bmatrix}\dot{\boldsymbol{z}} \\ \dot{\theta}\end{bmatrix} + \boldsymbol{L}(\theta, \mathbf{z})\begin{bmatrix}\ddot{\boldsymbol{z}} \\ \ddot{\theta}\end{bmatrix}, \tag{4.40}$$

where

$$\dot{\boldsymbol{L}}(\theta, \mathbf{z}) = \begin{bmatrix}\mathbf{0}_{4x3} & \begin{matrix}\Phi_1''(\theta)\dot{\theta} \\ \vdots \\ \Phi_4''(\theta)\dot{\theta}\end{matrix}\end{bmatrix}. \tag{4.41}$$

The vectors $\boldsymbol{q}, \dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ can now be substituted into Eq. (4.16)

$$\boldsymbol{M}\left(\dot{\boldsymbol{L}}(\theta,\mathbf{z})\begin{bmatrix}\dot{\boldsymbol{z}}\\ \dot{\theta}\end{bmatrix}+\boldsymbol{L}(\theta,\mathbf{z})\begin{bmatrix}\ddot{\boldsymbol{z}}\\ \ddot{\theta}\end{bmatrix}\right)+\boldsymbol{C}_{\boldsymbol{\lambda}}\,\boldsymbol{L}(\theta,\boldsymbol{z})\begin{bmatrix}\dot{\boldsymbol{z}}\\ \dot{\theta}\end{bmatrix}+\underbrace{\boldsymbol{G}_{\lambda}}_{=0}=\boldsymbol{B}_{\boldsymbol{\lambda}}\,\boldsymbol{u}. \tag{4.42}$$

Rearranging the expression:

$$\boldsymbol{M}\,\boldsymbol{L}(\theta,\mathbf{z})\begin{bmatrix}\ddot{\boldsymbol{z}}\\ \ddot{\theta}\end{bmatrix}=\boldsymbol{B}_{\boldsymbol{\lambda}}\,\boldsymbol{u}-\boldsymbol{C}_{\boldsymbol{\lambda}}\,\boldsymbol{L}(\theta,\boldsymbol{z})\begin{bmatrix}\dot{\boldsymbol{z}}\\ \dot{\theta}\end{bmatrix}-\boldsymbol{M}\,\dot{\boldsymbol{L}}(\theta,\mathbf{z})\begin{bmatrix}\dot{\boldsymbol{z}}\\ \dot{\theta}\end{bmatrix} \tag{4.43}$$

$$\begin{bmatrix}\ddot{\boldsymbol{z}}\\ \ddot{\theta}\end{bmatrix}=-\boldsymbol{L}^{-1}(\theta,\mathbf{z})\left[\boldsymbol{M}^{-1}\boldsymbol{C}_{\boldsymbol{\lambda}}\,\boldsymbol{L}(\theta,\boldsymbol{z})+\dot{\boldsymbol{L}}(\theta,\mathbf{z})\right]\begin{bmatrix}\dot{\boldsymbol{z}}\\ \dot{\theta}\end{bmatrix}+\boldsymbol{L}^{-1}(\theta,\mathbf{z})\,\boldsymbol{M}^{-1}\boldsymbol{B}_{\boldsymbol{\lambda}}\,\boldsymbol{u}, \tag{4.44}$$

where the following relation can be found

$$\ddot{\boldsymbol{z}}=\boldsymbol{R}(\theta,\dot{\theta},\boldsymbol{z},\dot{\boldsymbol{z}})+\boldsymbol{N}(\theta,\boldsymbol{z})\boldsymbol{u}, \tag{4.45}$$

with

$$\boldsymbol{R}(\theta,\dot{\theta},\boldsymbol{z},\dot{\boldsymbol{z}})=\begin{bmatrix}\boldsymbol{I}_3 & \mathbf{0}_{3x1}\end{bmatrix}\boldsymbol{L}^{-1}(\theta,\mathbf{z})\left[-\boldsymbol{M}^{-1}\boldsymbol{C}_{\boldsymbol{\lambda}}\,\boldsymbol{L}(\theta,\boldsymbol{z})-\dot{\boldsymbol{L}}(\theta,\mathbf{z})\right]\begin{bmatrix}\dot{\boldsymbol{z}}\\ \dot{\theta}\end{bmatrix} \tag{4.46}$$

$$\boldsymbol{N}(\theta,\boldsymbol{z})=\begin{bmatrix}\boldsymbol{I}_3 & \mathbf{0}_{3x1}\end{bmatrix}\boldsymbol{L}^{-1}(\theta,\mathbf{z})\,\boldsymbol{M}^{-1}\boldsymbol{B}_{\boldsymbol{\lambda}}. \tag{4.47}$$

Doing the calculations Eq. (4.45) can be written as:

$$\ddot{z}_1(\theta,\dot{\theta},\boldsymbol{z},\dot{\boldsymbol{z}},\boldsymbol{u})=-r_c\sin(\theta)\dot{\theta}\dot{z}_3+\frac{r_c\cos(\theta)}{mr_c^2+I_\phi}u_1-\frac{r_p\cos(\theta)}{I_\theta}u_2 \tag{4.48a}$$

$$\ddot{z}_2(\theta,\dot{\theta},\boldsymbol{z},\dot{\boldsymbol{z}},\boldsymbol{u})=r_c\cos(\theta)\dot{\theta}\dot{z}_3+\frac{r_c\sin(\theta)}{mr_c^2+I_\phi}u_1-\frac{r_p\sin(\theta)}{I_\theta}u_2 \tag{4.48b}$$

$$\ddot{z}_3(\theta,\dot{\theta},\boldsymbol{z},\dot{\boldsymbol{z}},\boldsymbol{u})=\frac{1}{mr_c^2+I_\phi}u_1-\frac{r_p}{I_\theta r_c}u_2, \tag{4.48c}$$

and in addition

$$\ddot{\theta}=\frac{1}{I_\theta}u_2. \tag{4.49}$$

The system dynamics with the change of coordinates from $[\boldsymbol{q},\dot{\boldsymbol{q}}]\rightarrow[\boldsymbol{z},\dot{\boldsymbol{z}},\theta,\dot{\theta}]$ has now been done. A desired trajectory $[\boldsymbol{q}_*(t);\ \dot{\boldsymbol{q}}_*(t)]$ then takes the form $\Big[\boldsymbol{z}_*(t)=0,\ \dot{\boldsymbol{z}}_*(t)=0,\ \theta_*(t),\ \dot{\theta}_*(t)\Big]$, meaning that the motions are now described by the virtual holonomic constraints given in Eq. (3.10) and the desired curve in the phase plane of the path coordinate $[\theta,\dot{\theta}]$.

4.3.2 Calculating the Control Transformation

In order to analyze the motion of the system in the vicinity of a desired orbit traced out by $[\boldsymbol{z}_*(t) = 0,\ \dot{\boldsymbol{z}}_*(t) = 0,\ \theta_*(t),\ \dot{\theta}_*(t)]$ in new coordinates, consider a control transformation in the form of

$$\boldsymbol{u} = \boldsymbol{U}(\theta, \dot{\theta}, \boldsymbol{z}, \dot{\boldsymbol{z}}) + \boldsymbol{v} \tag{4.50}$$

in which $\boldsymbol{U}(\cdot)$ corresponds to the nominal control at any point on the orbit such that the interpolation condition $\boldsymbol{u}_*(t) = \boldsymbol{U}(\theta_*(t), \dot{\theta}_*(t), 0, 0)$ holds. The new control variable $\boldsymbol{v}$ is zero on the desired orbit $\boldsymbol{v}_*(t) = \boldsymbol{0}$ and shall be used to force the system towards the orbit by control action. To compute $\boldsymbol{U}(\cdot)$, Eq. (4.37),(4.38) and (4.40) are inserted into Eq. (4.16) such that:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}_{\boldsymbol{\lambda}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \boldsymbol{B}_{\boldsymbol{\lambda}}(\boldsymbol{q})\boldsymbol{U} \Bigg| \begin{aligned} \boldsymbol{q} &= \boldsymbol{\Phi}(\theta) + \boldsymbol{z} \\ \dot{\boldsymbol{q}} &= \boldsymbol{L}(\theta, \boldsymbol{z}) \begin{bmatrix} \dot{\boldsymbol{z}} \\ \dot{\theta} \end{bmatrix} \\ \ddot{\boldsymbol{q}} &= \dot{\boldsymbol{L}}(\theta, \mathbf{z}) \begin{bmatrix} \dot{\boldsymbol{z}} \\ \dot{\theta} \end{bmatrix} + \boldsymbol{L}(\theta, \mathbf{z}) \begin{bmatrix} \ddot{\boldsymbol{z}} \\ \ddot{\theta} \end{bmatrix} \end{aligned} \tag{4.51}$$

which yields

$$\begin{bmatrix} m(\ddot{z}_1 + r_p\cos(\theta)\ddot{\theta} - r_p\sin(\theta)\dot{\theta}^2) + \dot{\theta}mr_c\sin(\theta)(\dot{z}_3 + \frac{\dot{\theta}r_p}{r_c}) \\ m(\ddot{z}_2 + r_p\sin(\theta)\ddot{\theta} + r_p\cos(\theta)\dot{\theta}^2) - \dot{\theta}mr_c\cos(\theta)(\dot{z}_3 + \frac{\dot{\theta}r_p}{r_c}) \\ I_\phi(\ddot{z}_3 + \frac{\ddot{\theta}r_p}{r_c}) \\ I_\theta\ddot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{mr_c\cos(\theta)}{mr_c^2+I_\phi}U_1 + v_1 \\ \frac{mr_c\sin(\theta)}{mr_c^2+I_\phi}U_1 + v_1 \\ \frac{I_\phi}{mr_c^2+I_\phi}U_1 + v_1 \\ U_2 + v_2 \end{bmatrix}. \tag{4.52}$$

Concentrating on the two bottom rows and setting $\boldsymbol{z}, \dot{\boldsymbol{z}}, \ddot{\boldsymbol{z}}, \boldsymbol{v} = \boldsymbol{0}$, the expression for U_1 and U_2 are found to be

$$\begin{aligned} U_1(\ddot{\theta}) &= \frac{r_p}{r_c}\ddot{\theta}(mr_c^2 + I_\phi) \\ U_2(\ddot{\theta}) &= I_\theta\ddot{\theta}. \end{aligned} \tag{4.53}$$

4.3.3 Finding $I(\theta, \dot{\theta})$ and its Derivatives

In [11] $I(\theta, \dot{\theta})$ is found using the "alpha-beta-gamma" equations, but in this case they turned out to be zero. First off all lets check if the nonholonomic

constraints are invariant along the motion as expected. This fact clearly disqualifies those quantities as transverse coordinates. Eq. (4.2) is given as

$$I(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \dot{x} - r_c\dot{\phi}\cos(\theta)$$

and its derivative can be calculated as

$$\dot{I}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \ddot{x} - r_c\ddot{\phi}\cos(\theta) + r_c\dot{\phi}\dot{\theta}\sin(\theta). \tag{4.54}$$

Performing a coordinate transformation by replacing $(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ gives

$$\dot{I}(\theta, \dot{\theta}, \ddot{\theta}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \ddot{\boldsymbol{z}}) = \ddot{z}_1 + \ddot{\Phi}_1(\theta) - r_c\cos(\theta)(\ddot{z}_3 + \ddot{\Phi}_3(\theta)) + r_c\dot{\theta}\sin(\theta)(\dot{z}_3 + \dot{\Phi}_3(\theta)). \tag{4.55}$$

By substituting $\dot{\boldsymbol{\Phi}}(\theta)$ and $\ddot{\boldsymbol{\Phi}}(\theta)$

$$\begin{aligned}\dot{I}(\theta, \dot{\theta}, \ddot{\theta}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \ddot{\boldsymbol{z}}) &= \ddot{z}_1 + (-r_p\sin(\theta)\dot{\theta}^2 + r_p\cos(\theta)\ddot{\theta}) \\ &\quad - r_c\cos(\theta)(\ddot{z}_3 + \frac{r_p}{r_c}\ddot{\theta}) + r_c\dot{\theta}\sin(\theta)(\dot{z}_3 + \frac{r_p}{r_c}\dot{\theta}) \\ &= \ddot{z}_1 - \cancel{r_p\sin(\theta)\dot{\theta}^2} + \cancel{r_p\cos(\theta)\ddot{\theta}} - r_c\cos(\theta)\ddot{z}_3 \\ &\quad - \cancel{r_p\cos(\theta)\ddot{\theta}} + r_c\dot{\theta}\sin(\theta)\dot{z}_3 + \cancel{r_p\sin(\theta)\dot{\theta}^2}.\end{aligned} \tag{4.56}$$

Continuing using Eq. (4.48) as a substitution for $\ddot{\boldsymbol{z}}$:

$$\begin{aligned}\dot{I}(\theta, \dot{\theta}, \ddot{\theta}, \dot{\boldsymbol{z}}, \boldsymbol{v}) =& \ddot{z}_1 - \ddot{z}_3 r_c\cos(\theta) + r_c\sin(\theta)\dot{\theta}\dot{z}_3 \\ =& - r_c\sin(\theta)\dot{\theta}\dot{z}_3 + \frac{r_c\cos(\theta)}{mr_c^2 + I_\phi}u_1 - \frac{r_p\cos(\theta)}{I_\theta}u_2 \\ & - \left(\frac{1}{mr_c^2 + I_\phi}u_1 - \frac{r_p}{I_\theta r_c}u_2\right) r_c\cos(\theta) + \dot{z}_3\dot{\theta}r_c\sin(\theta) \\ =& \frac{r_c\cos(\theta)}{mr_c^2 + I_\phi}u_1 - \frac{r_p\cos(\theta)}{I_\theta}u_2 - \frac{r_c\cos(\theta)}{mr_c^2 + I_\phi}u_1 + \frac{r_p r_c\cos(\theta)}{I_\theta r_c}u_2 \\ =& 0,\end{aligned} \tag{4.57}$$

which is as expected. The same can also be observed in the y-direction, but the calculations are very similar so it will be left out. An expression for I still needs to be found and by implementing a constant angular velocity, ω =const, I can be constructed such that

$$\begin{aligned}I(\dot{\theta}) &= \dot{\theta}^2 - \omega^2 = 0 \\ \dot{I}(\dot{\theta}, \ddot{\theta}) &= 2\dot{\theta}\ddot{\theta} - 0\end{aligned} \tag{4.58}$$

With Eq. (4.49) in mind, $\ddot{\theta}$ can be substituted

$$\dot{I}(\dot{\theta}, \boldsymbol{u}) = \frac{2\dot{\theta}}{I_\theta} u_2 \tag{4.59}$$

It is known from Eq. (4.50) that $\boldsymbol{u}$ will be divided into two parts; the nominal input $\boldsymbol{U}(\ddot{\theta})$ and the control input $\boldsymbol{v}$ such that

$$\dot{I}(\dot{\theta}, \boldsymbol{u}) = \frac{2\dot{\theta}}{I_\theta}(U_2 + v_2) = 2\dot{\theta}\ddot{\theta} + \frac{2\dot{\theta}}{I_\theta} v_2 \tag{4.60}$$

$$\Delta\dot{I}(\dot{\theta}, \boldsymbol{v}) = \frac{2\dot{\theta}}{I_\theta} v_2 \tag{4.61}$$

4.3.4 State Space Representation

Until now a trajectory has been designed and the dynamics for the transverse coordinates has been calculated as a function of $\ddot{\boldsymbol{z}}$ and I. The next step in analysis and control design is to compute the transverse linearization in the form of Eq. (3.16).

$$\frac{d}{d\tau}\begin{bmatrix}\Delta I \\ \Delta \boldsymbol{z} \\ \Delta \dot{\boldsymbol{z}}\end{bmatrix} = \mathcal{A}(\tau)\begin{bmatrix}\Delta I \\ \Delta \boldsymbol{z} \\ \Delta \dot{\boldsymbol{z}}\end{bmatrix} + \mathcal{B}(\tau)\Delta V(\tau).$$

Rewriting Eq. (4.48) into matrix form it looks like

$$\begin{bmatrix}\ddot{z}_1 \\ \ddot{z}_2 \\ \ddot{z}_3\end{bmatrix} = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & -r_c\sin(\theta)\dot{\theta} \\ 0 & 0 & 0 & 0 & 0 & 0 & r_c\cos(\theta)\dot{\theta} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3\end{bmatrix} + \begin{bmatrix}\frac{r_c\cos(\theta)}{mr_c^2+I_\phi} & -\frac{r_p\cos(\theta)}{I_\theta} \\ \frac{r_c\sin(\theta)}{mr_c^2+I_\phi} & -\frac{r_p\sin(\theta)}{I_\theta} \\ \frac{1}{mr_c^2+I_\phi} & -\frac{r_p}{r_c I_\theta}\end{bmatrix}\begin{bmatrix}u_1 \\ u_2\end{bmatrix}. \tag{4.62}$$

From this it can be seen that the system is already linear in the transverse coordinates, $\boldsymbol{z}, \dot{\boldsymbol{z}}$, and no further linearization is necessary. The only thing left is to switch the input to the controller input and use the linear system notation. As for ΔI it can be seen from Eq. (4.61). Changing notation and putting it into a system it can be seen that

$$\begin{bmatrix} \Delta\dot{I} \\ \Delta\dot{z}_1 \\ \Delta\dot{z}_2 \\ \Delta\dot{z}_3 \\ \Delta\ddot{z}_1 \\ \Delta\ddot{z}_2 \\ \Delta\ddot{z}_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -r_c\sin(\theta)\dot{\theta} \\ 0 & 0 & 0 & 0 & 0 & 0 & r_c\cos(\theta)\dot{\theta} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathcal{A}(\tau)} \begin{bmatrix} \Delta I \\ \Delta z_1 \\ \Delta z_2 \\ \Delta z_3 \\ \Delta\dot{z}_1 \\ \Delta\dot{z}_2 \\ \Delta\dot{z}_3 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & \frac{2\dot{\theta}}{I_\theta} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{r_c\cos(\theta)}{mr_c^2+I_\phi} & -\frac{r_p\cos(\theta)}{I_\theta} \\ \frac{r_c\sin(\theta)}{mr_c^2+I_\phi} & -\frac{r_p\sin(\theta)}{I_\theta} \\ \frac{1}{mr_c^2+I_\phi} & -\frac{r_p}{r_cI_\theta} \end{bmatrix}}_{\mathcal{B}(\tau)} \begin{bmatrix} \Delta V_1 \\ \Delta V_2 \end{bmatrix} \tag{4.63}$$

4.3.5 The Control Law

Making a controller for a nonlinear time varying system is an active research field on itself. In the lack of good controllers simplifications are many times made, such as linearization, which is what is done in the previous chapters. For linear time varying (LTV) system, different control strategies does exist, but are known to be difficult to implement and use. The system to be controlled now looks like Eq. (3.19)

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}(t)\boldsymbol{x}(t) + \boldsymbol{B}(t)\boldsymbol{u}(t).$$

Controllability

First of all the controllability of the system needs to be checked. One method that will be tested is to use the controllability matrix described in Sec. 3.4.1. Because that method is designed for LTI systems, it is not possible to check whether or not the system is controllable over a whole period. Instead it will be used to check a selection of time instances spread out over the period and give an indication of how many of the states that are controllable. By using the `ctrb(A,B)` command in MATLAB for some evenly distributed samples of time over the period for the orbit, the results yielded a rank of 4 on every pair of $\mathcal{A}(\tau)$ and $\mathcal{B}(\tau)$. An example of how the controllability matrix look

like is given here:

$$\begin{bmatrix} 0 & \frac{3927}{625} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1923}{2500} & -5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{4833}{5000} & -\frac{3927}{625} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{30769}{10000} & -20 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1923}{2500} & -5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{4833}{5000} & -\frac{3927}{625} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{30769}{10000} & -20 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{4.64}$$

where the values are examples at a time instant. The rest of the time the structure is the same, only that the values of the non-zero elements alter. All the zero elements stays at zero.
A method making use of the controllability gramian for LTV systems is described in the same section as the controllability matrix mentioned above. The controllability gramian method needed to be implemented as there is no commonly known controllability gramian solvers for LTV systems similar to Eq. (3.19). For the convenience of the reader the controllability gramian equation for LTV systems given in Eq. (3.20) is reproduced here

$$\boldsymbol{W}_c(t) = \int_{t_0}^{t} \boldsymbol{\Phi}(t_0, \tau)\boldsymbol{B}(\tau)\boldsymbol{B}^T(\tau)\boldsymbol{\Phi}^T(t_0, \tau)\ d\tau.$$

Calculating the controllability gramian on the system at time $t = T$, i.e. after a full period the eigenvalues look like:

$$[19934;\ 1109;\ 662.1;\ 195.9;\ 4.182;\ 7.514;\ 0.5148] \tag{4.65}$$

which yields a rank of 7, which is the same as the number of states in the system. Because of the nonholonomic constraints a uncontrollable subspace was expected, as stated in Sec. 2.3. With a rank of 7 and the gramian over a period being non-singular it is expected for the system to be controllable for all states. A periodic controller then needs to be designed.

The Bézier approach

The controller to be designed will be a periodic, time dependent feedback controller where

$$\boldsymbol{v}(\tau) = \boldsymbol{K}(\tau)\Delta\boldsymbol{x}_\perp, \tag{4.66}$$

such that

$$\Delta\dot{\boldsymbol{x}}_\perp = \mathcal{A}(\tau)\Delta\boldsymbol{x}_\perp(\tau) + \mathcal{B}(\tau)\boldsymbol{K}(\tau)\Delta\boldsymbol{x}_\perp \tag{4.67}$$

$$= \Big(\underbrace{\mathcal{A}(\tau) + \mathcal{B}(\tau)\boldsymbol{K}(\tau)}_{\widetilde{\mathcal{A}}}\Big)\Delta\boldsymbol{x}_\perp. \tag{4.68}$$

This is as said earlier no trivial task as the periodic and time varying part of the controller can be hard to achieve. One method that does exist is one where an algebraic Riccati equation is solved, but the task of making it periodic is very difficult and it is hard to make qualified initial guesses for the different parameters. This led to another method where Bézier curves laid the foundation for the time varying controller as they are cheap to compute and has some favourable properties. The idea and the framework for this method was made by Stian Askeland in [16]. The method was then adapted to the problem given in this thesis.

Bézier Controller

A Bézier curve is a curve constructed in a specific way using a number of points to define a curve. With reference to [17] and [18] the following describes the properties and how to construct such a curve. Given a set of $n+1$ points $P_0, P_1, P_2, \cdots, P_n$ the curve is associated with the advantageous properties

- It starts at P_0 and ends at P_n.
- When it starts from P_0 it heads directly towards P_1, and when it arrives at P_n it is coming from the direction of P_{n-1}.

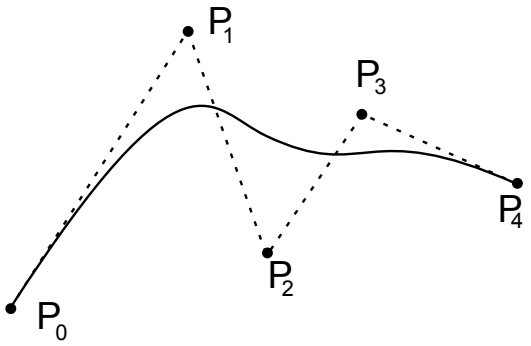


Figure 4.3: A Bézier curve with 5 control points

Given the set of control points mentioned above, the corresponding curve is given by

$$\boldsymbol{C}(t) = \sum_{i=0}^{n} P_i B_{i,n}(t), \tag{4.69}$$

where $t \in [0, 1]$ and $B_{i,n}(t)$ is a Bernstein polynomial defined by

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i} \tag{4.70}$$

Given the controller in Eq. (4.67), the trick is to find a $\boldsymbol{K}(t)$ such that the system becomes stable. By using Bézier curves to design $\boldsymbol{K}(t)$ it is possible to optimize on the controller with a cost function based on the closed loop system over a period and with the Bézier points as the variable. By also making certain restrictions like for instance that the first two points match the last two points, i.e. the derivative and the state at the start and end are equal, such that the controller also becomes periodic. The control design was done specifying a set of initial points, and use those as input to the bézier algorithm. Then each of the 14 (states x inputs) control gain variables in $\boldsymbol{K}(t)$ would be represented as a bézier curve through time. This would as mentioned above make them smooth and periodic if wanted. For each $\boldsymbol{K}(t)$ found, the state transition matrix for $\widetilde{\boldsymbol{A}}(t) = (\boldsymbol{A}(t) - \boldsymbol{B}(t)\boldsymbol{K}(t))$ would be calculated. The norm of that will then tell if the system is stable or not. If the norm is less that one, it means that given an initial condition with an error in all states, the system will decrease the error in one period.

4.4 A Straight Line Path Segment Instead of Circular

As mentioned in Sec. 3.2.3 the path provided by the modified RRT method would consist of quadrants and straight line segments. The dynamics where the path was orbital and circular was looked into in the previous chapter, and now the dynamics where the path is given by straight line segments will be looked into.

Given a straight line segment it can be assumed that θ will be kept constant, but ϕ will increase along the path. The two other coordinates x and y will depend on some initial configuration, ϕ and its derivatives. Given a line segment the synchronization function can look like:

$$\boldsymbol{q}_* = \begin{bmatrix} x_* \\ y_* \\ \phi_* \\ \theta_* \end{bmatrix} = \boldsymbol{\Phi}(\phi) = \begin{bmatrix} x_0 + r_c\phi\cos(\theta_0) \\ y_0 + r_c\phi\sin(\theta_0) \\ \phi \\ \theta_0 \end{bmatrix}, \tag{4.71}$$

which can be differentiated as

$$\dot{\boldsymbol{q}}_* = \dot{\boldsymbol{\Phi}}(\phi) = \boldsymbol{\Phi}'(\phi)\dot{\phi} = \begin{bmatrix} r_c\cos(\theta_0) \\ r_c\sin(\theta_0) \\ 1 \\ 0 \end{bmatrix} \dot{\phi}, \tag{4.72}$$

and

$$\ddot{\boldsymbol{q}}_* = \ddot{\boldsymbol{\Phi}}(\phi) = \boldsymbol{\Phi}''(\phi)\dot{\phi}^2 + \boldsymbol{\Phi}'(\phi)\ddot{\phi} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \dot{\phi}^2 + \begin{bmatrix} r_c\cos(\theta_0) \\ r_c\sin(\theta_0) \\ 1 \\ 0 \end{bmatrix} \ddot{\phi} \tag{4.73}$$

where $\phi = \phi_*(t), \quad \dot{\phi} = \dot{\phi}_*(t), \quad \ddot{\phi} = \ddot{\phi}_*(t), \quad t \in [0, T]$

4.4.1 Calculation of the α, β and γ Functions

Following the same procedure as in Sec. 4.2.1:

$$\alpha = \begin{bmatrix} \frac{mr_c\sin(\theta_0-\theta)}{cos(\theta)} \\ I_\phi - \frac{I_\phi\cos(\theta_0)}{\cos(\theta)} \end{bmatrix} \tag{4.74}$$

$$\beta = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{4.75}$$

were, in the ideal case, $\theta = \theta_0$ such that

$$\alpha = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4.76}$$

$$\beta = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4.77}$$

in which they both are zero. Again it can be seen that along the path the trajectory can be arbitrarily but smoothly shaped.

4.4.2 Transverse Linearization and Orbital Stabilization

Introducing the New Coordinates

In this case $q_3 = \Phi_4(\phi) = \phi$, such that

$$x = z_1 + \Phi_1(\phi) = z_1 + (x_0 + r_c\phi\cos(\theta_0)) \tag{4.78a}$$
$$y = z_2 + \Phi_2(\phi) = z_2 + (y_0 + r_c\phi\sin(\theta_0)) \tag{4.78b}$$
$$\phi = 0 + \Phi_3(\phi) = \phi \tag{4.78c}$$
$$\theta = z_3 + \Phi_4(\phi) = z_3 + \theta_0 \tag{4.78d}$$

The main difference from the circle case, except from the shape of the path, is that z_3 now is associated with $\Phi_4(\phi)$. Continuing with the calculations, the derivatives can be shown to be

$$\dot{\boldsymbol{q}} = \begin{bmatrix} \dot{q_1} \\ \dot{q_2} \\ \dot{q_3} \\ \dot{q_4} \end{bmatrix} = \begin{bmatrix} \dot{z_1} + \Phi_1'(\phi)\dot{\phi} \\ \dot{z_2} + \Phi_2'(\phi)\dot{\phi} \\ 0 + \Phi_3'(\phi)\dot{\phi} \\ \dot{z_3} + \Phi_4'(\phi)\dot{\phi} \end{bmatrix} = L(\phi, \boldsymbol{z}) \begin{bmatrix} \dot{\boldsymbol{z}} \\ \dot{\phi} \end{bmatrix} \tag{4.79}$$

where

$$L(\phi, \boldsymbol{z}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} & \Phi_1'(\phi) \\ \mathbf{0}_{4x3} & \vdots \\ & \Phi_4'(\phi) \end{bmatrix} \tag{4.80}$$

Furter on the double derivative is

$$\ddot{\boldsymbol{q}} = \dot{L}(\phi, \mathbf{z}) \begin{bmatrix} \dot{\boldsymbol{z}} \\ \dot{\phi} \end{bmatrix} + L(\phi, \mathbf{z}) \begin{bmatrix} \ddot{\boldsymbol{z}} \\ \ddot{\phi} \end{bmatrix} \tag{4.81}$$

where

$$\dot{L}(\phi, \mathbf{z}) = \begin{bmatrix} & \Phi_1''(\phi)\dot{\phi} \\ \mathbf{0}_{4x3} & \vdots \\ & \Phi_4''(\phi)\dot{\phi} \end{bmatrix} = \mathbf{0} \tag{4.82}$$

Following the same method as for the circle, calculating the double derivative for the error

$$\ddot{\boldsymbol{z}} = \boldsymbol{R}(\phi, \dot{\phi}, \boldsymbol{z}, \dot{\boldsymbol{z}}) + \boldsymbol{N}(\phi, \boldsymbol{z})\boldsymbol{u}, \tag{4.83}$$
$$\tag{4.84}$$

where

$$\ddot{z}_1(\phi, \dot{\phi}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \boldsymbol{u}) = -\dot{\phi} r_c \sin(\theta_0 + z_3)\dot{z}_3 - \frac{r_c(\cos(\theta_0) - \cos(\theta_0 + z_3))}{mr_c^2 + I_\phi} u_1 \tag{4.85a}$$

$$\ddot{z}_2(\phi, \dot{\phi}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \boldsymbol{u}) = \dot{\phi} r_c \cos(\theta_0 + z_3)\dot{z}_3 - \frac{r_c(\sin(\theta_0) - \sin(\theta_0 + z_3))}{mr_c^2 + I_\phi} u_1 \tag{4.85b}$$

$$\ddot{z}_3(\phi, \dot{\phi}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \boldsymbol{u}) = \frac{1}{I_\theta} u_2 \tag{4.85c}$$

and in addition

$$\ddot{\phi} = \frac{1}{mr_c^2 + I_\phi} u_1 \tag{4.86}$$

Calculating the Control Transformation

The control calculations can be done as in Sec. 4.3.2, inserting Eq. (4.78),(4.79) and (4.81) into Eq. (4.16) and yield

$$\begin{bmatrix} m(\ddot{z}_1 + \ddot{\phi} r_c \cos(\theta_0)) + \dot{\phi}\dot{z}_3 m r_c \sin(\theta_0 + z_3) \\ m(\ddot{z}_2 + \ddot{\phi} r_c \sin(\theta_0)) - \dot{\phi}\dot{z}_3 m r_c \cos(\theta_0 + z_3) \\ I_\phi \ddot{\phi} \\ I_\theta \ddot{z}_3 \end{bmatrix} = \begin{bmatrix} \frac{mr_c\cos(\theta)}{mr_c^2 + I_\phi} U_1 + v_1 \\ \frac{mr_c\sin(\theta)}{mr_c^2 + I_\phi} U_1 + v_1 \\ \frac{I_\phi}{mr_c^2 + I_\phi} U_1 + v_1 \\ U_2 + v_2 \end{bmatrix}. \tag{4.87}$$

And again concentrating on the two bottom rows and setting $\boldsymbol{z}, \dot{\boldsymbol{z}}, \ddot{\boldsymbol{z}}, \boldsymbol{v} = \boldsymbol{0}$, the expression for U_1 and U_2 are found to be

$$\begin{aligned} U_1 &= \ddot{\phi}(mr_c^2 + I_\phi), \\ U_2 &= 0 \end{aligned} \tag{4.88}$$

Linearizing $\ddot{\boldsymbol{z}}$ Using Taylor Series Expansion

As seen earlier in the text, $\ddot{\boldsymbol{z}}$ for the straight line segment is not linear in $\boldsymbol{z}, \dot{\boldsymbol{z}}$ and the expressions needs to be linearized. This is done using Taylor series expansion which will be described shortly.

According to [19] a Taylor series is a series expansion of a function, $f(x)$, about a point, $x = a$,

$$f(x) = f(a) + f'(a)(x - a) + O(x, a), \tag{4.89}$$

where $O(x, a)$ denotes higher order terms. To achieve a model as linear as possible we can stick to the first two orders and ignore the rest. In the case where the system is described using matrices we get:

$$\boldsymbol{f}(\boldsymbol{x}) \approx \boldsymbol{f}(\boldsymbol{a}) + \boldsymbol{\nabla}\, \boldsymbol{f}(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{a}}\, (\boldsymbol{x} - \boldsymbol{a}) \tag{4.90}$$

$$\nabla \boldsymbol{f} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \ddots & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \tag{4.91}$$

Using Taylor series expansion on $\ddot{\boldsymbol{z}}$ described in Eq. (4.85):

$$\ddot{\boldsymbol{z}}(\boldsymbol{x}) \approx \ddot{\boldsymbol{z}}(\boldsymbol{a}) + \left.\frac{\partial \ddot{\boldsymbol{z}}(\boldsymbol{x})}{\partial x_1}\right|_{\boldsymbol{x}=\boldsymbol{a}} (x_1 - a_1) + \cdots + \left.\frac{\partial \ddot{\boldsymbol{z}}(\boldsymbol{x})}{\partial x_n}\right|_{\boldsymbol{x}=\boldsymbol{a}} (x_n - a_n) \tag{4.92}$$

where $\boldsymbol{x} = \left[\phi, \dot{\phi}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \boldsymbol{u}\right]^T$, $\boldsymbol{a} = \left[\phi_*, \dot{\phi}_*, \boldsymbol{0}, \boldsymbol{0}, \boldsymbol{U}\right]^T$ and $\boldsymbol{x} - \boldsymbol{a} = [0, 0, \boldsymbol{z}, \dot{\boldsymbol{z}}, \boldsymbol{v}]^T$ and $n = dim(\boldsymbol{q})$. Making use of the relationship from Eq. (4.50), where $\boldsymbol{U}(\cdot)$ is calculated in Eq. (4.88) the formula yields

$$\begin{aligned} \ddot{\boldsymbol{z}}(\phi, \dot{\phi}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \boldsymbol{u}) \approx & \ddot{\boldsymbol{z}}(\phi_*, \dot{\phi}_*, \boldsymbol{0}, \boldsymbol{0}, \boldsymbol{U}) + \nabla_{\boldsymbol{z}} \ddot{\boldsymbol{z}}(\phi, \dot{\phi}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \boldsymbol{u})\Big|_{\boldsymbol{x}=\boldsymbol{a}} \boldsymbol{z} \\ & + \nabla_{\dot{\boldsymbol{z}}} \ddot{\boldsymbol{z}}(\phi, \dot{\phi}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \boldsymbol{u})\Big|_{\boldsymbol{x}=\boldsymbol{a}} \dot{\boldsymbol{z}} + \nabla_{\boldsymbol{u}} \ddot{\boldsymbol{z}}(\phi, \dot{\phi}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \boldsymbol{u})\Big|_{\boldsymbol{x}=\boldsymbol{a}} \boldsymbol{v}, \end{aligned} \tag{4.93}$$

since $\phi - \phi_* = \dot{\phi} - \dot{\phi}_* = 0$. Continuing with the calculations

$$\ddot{\boldsymbol{z}}(\phi_*, \dot{\phi}_*, \mathbf{0}, \mathbf{0}, \boldsymbol{U}) = 0 \tag{4.94}$$

$$\nabla_{\boldsymbol{z}} \ddot{\boldsymbol{z}}(\phi, \dot{\phi}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \boldsymbol{u})\Big|_{\boldsymbol{x}=\boldsymbol{a}} \boldsymbol{z} = \begin{bmatrix} 0 & 0 & -\frac{r_c \sin(\theta_0)}{m r_c^2 + I_\phi} U_1 \\ 0 & 0 & \frac{r_c \cos(\theta_0)}{m r_c^2 + I_\phi} U_1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \tag{4.95}$$

$$\nabla_{\dot{\boldsymbol{z}}} \ddot{\boldsymbol{z}}(\phi, \dot{\phi}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \boldsymbol{u})\Big|_{\boldsymbol{x}=\boldsymbol{a}} \dot{\boldsymbol{z}} = \begin{bmatrix} 0 & 0 & -\dot{\phi} r_c \sin(\theta_0) \\ 0 & 0 & \dot{\phi} r_c \cos(\theta_0) \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} \tag{4.96}$$

$$\nabla_{\boldsymbol{u}} \ddot{\boldsymbol{z}}(\phi, \dot{\phi}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \boldsymbol{u})\Big|_{\boldsymbol{x}=\boldsymbol{a}} \boldsymbol{v} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{I_\theta} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \tag{4.97}$$

where canceling terms and the terms containing zero are removed. Collecting and rearranging the terms:

$$\begin{aligned} \ddot{z}_1 &= -\frac{r_c \sin(\theta_0)}{m r_c^2 + I_\phi} U_1 z_3 - \dot{\phi} r_c \sin(\theta_0) \dot{z}_3 \\ &= -r_c \sin(\theta_0) \ddot{\phi} z_3 - \dot{\phi} r_c \sin(\theta_0) \dot{z}_3 \qquad (4.98a) \\ \ddot{z}_2 &= \frac{r_c \cos(\theta_0)}{m r_c^2 + I_\phi} U_1 z_3 + \dot{\phi} r_c \cos(\theta_0) \dot{z}_3 \\ &= r_c \cos(\theta_0) \ddot{\phi} z_3 + \dot{\phi} r_c \cos(\theta_0) \dot{z}_3 \qquad (4.98b) \\ \ddot{z}_3 &= \frac{1}{I_\theta} v_2. \qquad (4.98c) \end{aligned}$$

Finding I

Again, for the straight line segment the nonholonomic constraints needs to be checked if they are invariant along the desired motion. By using the relationship provided in Eq. (4.86) and replacing $\boldsymbol{q}, \dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ the following can be found

$$\begin{aligned} \dot{I}_x(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) &= \ddot{x} - r_c \ddot{\phi} \cos(\theta) + r_c \dot{\phi} \dot{\theta} \sin(\theta) \\ \dot{I}_x(\phi, \dot{\phi}, \ddot{\phi}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \ddot{\boldsymbol{z}}) &= \ddot{z}_1 + \ddot{\Phi}_1(\phi) - r_c \cos(\Phi_3(\phi) + z_3) \ddot{\phi} + r_c \dot{\phi} \sin(\Phi_3(\phi) + z_3)(\dot{z}_3 + \dot{\Phi}_3(\phi)) \\ \dot{I}_x(\phi, \dot{\phi}, \ddot{\phi}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \ddot{\boldsymbol{z}}, \boldsymbol{u}) &= \ddot{z}_1 + \ddot{\Phi}_1(\phi) - \frac{r_c \cos(\theta_0 + z_3)}{m r_c^2 + I_\phi} u_1 + r_c \dot{\phi} \dot{z}_3 \sin(\theta_0 + z_3) \\ \dot{I}_x(\phi, \dot{\phi}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \ddot{\boldsymbol{z}}, \boldsymbol{u}) &= \ddot{z}_1 + r_c \dot{\phi} \dot{z}_3 \sin(\theta_0 + z_3) - \frac{r_c(\cos(\theta_0 + z_3) - \cos(\theta_0))}{m r_c^2 + I_\phi} u_1 \end{aligned} \tag{4.99}$$

Remembering that $\phi - \phi_* = \dot{\phi} - \dot{\phi}_* = 0$, performing the Taylor series expansion gives

$$\dot{I}_x(\phi_*, \dot{\phi}_*, \mathbf{0}, \mathbf{0}, \mathbf{0}, \boldsymbol{U}) = 0 \tag{4.100}$$

$$\nabla_{\boldsymbol{\phi}} \dot{I}_x(\phi, \dot{\phi}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \ddot{\boldsymbol{z}}, \boldsymbol{u})\Big|_{\boldsymbol{x}=\boldsymbol{a}} (\phi - \phi_*) = 0 \tag{4.101}$$

$$\nabla_{\dot{\boldsymbol{\phi}}} \dot{I}_x(\phi, \dot{\phi}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \ddot{\boldsymbol{z}}, \boldsymbol{u})\Big|_{\boldsymbol{x}=\boldsymbol{a}} (\dot{\phi} - \dot{\phi}_*) = 0 \tag{4.102}$$

$$\nabla_{\boldsymbol{z}} \dot{I}_x(\phi, \dot{\phi}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \ddot{\boldsymbol{z}}, \boldsymbol{u})\Big|_{\boldsymbol{x}=\boldsymbol{a}} \boldsymbol{z} = \left[0, 0, \frac{r_c \sin(\theta_0)}{m r_c^2 + I_\phi} U_1\right] \boldsymbol{z} \tag{4.103}$$

$$\nabla_{\dot{\boldsymbol{z}}} \dot{I}_x(\phi, \dot{\phi}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \ddot{\boldsymbol{z}}, \boldsymbol{u})\Big|_{\boldsymbol{x}=\boldsymbol{a}} \dot{\boldsymbol{z}} = \left[0, 0, \dot{\phi} r_c \sin(\theta_0)\right] \dot{\boldsymbol{z}} \tag{4.104}$$

$$\nabla_{\ddot{\boldsymbol{z}}} \dot{I}_x(\phi, \dot{\phi}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \ddot{\boldsymbol{z}}, \boldsymbol{u})\Big|_{\boldsymbol{x}=\boldsymbol{a}} \ddot{\boldsymbol{z}} = [1, 0, 0]\, \ddot{\boldsymbol{z}} \tag{4.105}$$

$$\nabla_{\boldsymbol{u}} \dot{I}_x(\phi, \dot{\phi}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \ddot{\boldsymbol{z}}, \boldsymbol{u})\Big|_{\boldsymbol{x}=\boldsymbol{a}} \boldsymbol{v} = [0, 0]\, \boldsymbol{v} \tag{4.106}$$

hence

$$\begin{aligned}
\dot{I}_x(\phi, \dot{\phi}, \boldsymbol{U}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \ddot{\boldsymbol{z}}) =& \frac{r_c \sin(\theta_0)}{m r_c^2 + I_\phi} U_1 z_3 + \dot{\phi} r_c \sin(\theta_0) \dot{z}_3 + \ddot{z}_1 \\
\dot{I}_x(\phi, \dot{\phi}, \boldsymbol{U}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \boldsymbol{v}) =& \frac{r_c \sin(\theta_0)}{m r_c^2 + I_\phi} U_1 z_3 + \dot{\phi} r_c \sin(\theta_0) \dot{z}_3 \\
& - \frac{r_c \sin(\theta_0)}{m r_c^2 + I_\phi} U_1 z_3 - \dot{\phi} r_c \sin(\theta_0) \dot{z}_3 = 0
\end{aligned} \tag{4.107}$$

where $\ddot{z}_1$ is replaced. The same is also observed in $y-$direction, but again the calculations are very similar so they will be left out. This proves that the constraints are held given the desired trajectory.
The same idea for I as for the circular path is tried in this case, namely

$$I(\dot{\phi}) = \dot{\phi}^2 - \omega^2 = 0 \tag{4.108}$$

$$\dot{I}(\dot{\phi}, \ddot{\phi}) = 2\dot{\phi}\ddot{\phi} - 0, \tag{4.109}$$

where ω is representing a constant angular velocity for the wheel. This seems to be a good candidate and will be used. Making use of the same transformation as for the circle, Eq. (4.86) is used to transform I

$$\dot{I}(\dot{\phi}, \boldsymbol{u}) = 2\dot{\phi}\ddot{\phi} = \frac{2\dot{\phi}}{m r_c^2 + I_\phi} u_1, \tag{4.110}$$

where

$$\Delta I(\dot{\phi}, \boldsymbol{v}) = \frac{2\dot{\phi}}{mr_c^2 + I_\phi} v_1 \tag{4.111}$$

State Space Representation

With basis in Eq. (4.98) and (4.107) the state space representation can be expressed as

$$\begin{bmatrix} \Delta\dot{I} \\ \Delta\dot{z}_1 \\ \Delta\dot{z}_2 \\ \Delta\dot{z}_3 \\ \Delta\ddot{z}_1 \\ \Delta\ddot{z}_2 \\ \Delta\ddot{z}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -r_c\sin(\theta_0)\ddot{\phi} & 0 & 0 & -\dot{\phi}r_c\sin(\theta_0) \\ 0 & 0 & 0 & r_c\cos(\theta_0)\ddot{\phi} & 0 & 0 & \dot{\phi}r_c\cos(\theta_0) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta I \\ \Delta z_1 \\ \Delta z_2 \\ \Delta z_3 \\ \Delta\dot{z}_1 \\ \Delta\dot{z}_2 \\ \Delta\dot{z}_3 \end{bmatrix} + \begin{bmatrix} \frac{2\dot{\phi}}{mr_c^2+I_\phi} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{I_\theta} \end{bmatrix} \begin{bmatrix} \Delta V_1 \\ \Delta V_2 \end{bmatrix} \tag{4.112}$$

4.5 Short Review of the Five DOF Case

4.5.1 Initializing

As mentioned in Sec. 2.2, there are several ways to describe the unicycle. The simple one contained only 4 DOF, as the unicycle was said to always maintain a stable upright position vertically. The other option is a 5 DOF case where gravity can affect the vertical position of the unicycle. This is achieved by introducing a new coordinate ψ which represent the angle at which the unicycle is positioned vertically (i.e relative to a z-axis) as shown in Fig. 4.4.

The new generalized coordinates can be described as $\boldsymbol{q} = [x,\ y,\ \phi,\ \psi,\ \theta]^T$, where the four of them are recognized from before and a fifth one is added.

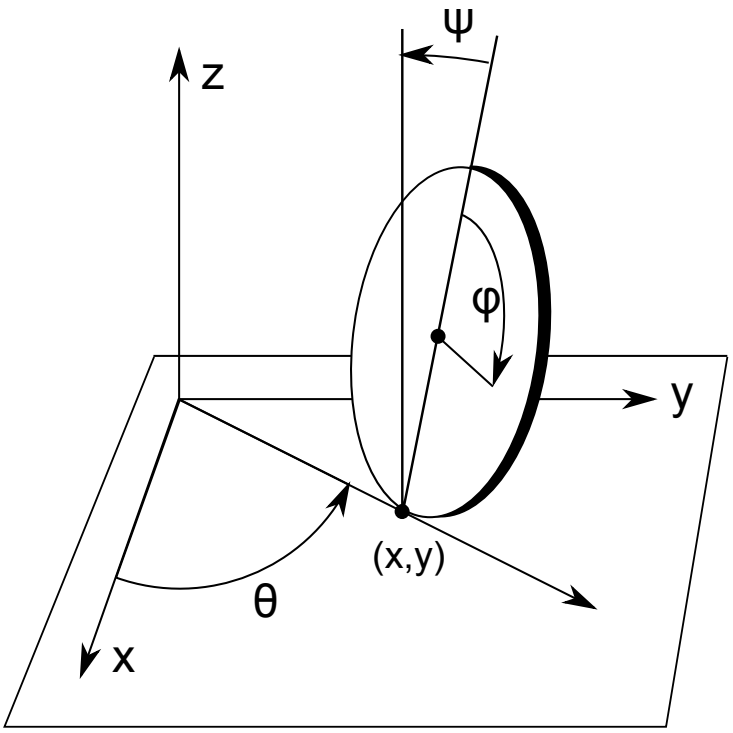


Figure 4.4: Configuration 5 DOF

4.5.2 The new EOM

In this section two different approaches for deriving the EOM for the 5 DOF system will be showed. The first one is described in a book, while the other one is derived by the author.

Method One

From [15] the Lagrangian for the system can be found as [1]

$$\mathcal{L} = \frac{m}{2}\left[\left(\xi - r_c\left(\dot{\theta}\sin(\psi) + \dot{\phi}\right)\right)^2 + \eta^2\sin^2(\psi) + \left(\eta\cos(\psi) + r_c\dot{\psi}\right)^2\right] \quad (4.113)$$

$$+ \frac{1}{2}\left[I_\theta\left(\dot{\psi}^2 + \dot{\theta}^2\cos^2(\psi)\right) + I_\phi\left(\dot{\theta}\sin(\psi) + \dot{\phi}\right)^2\right] - mgr_c\cos(\psi), \quad (4.114)$$

where [2]

$$\xi = \dot{x}\cos(\theta) + \dot{y}\sin(\theta) + r_c\dot{\phi} = 0 \quad (4.115)$$

$$\eta = -\dot{x}\sin(\theta) + \dot{y}\cos(\theta) = 0 \quad (4.116)$$

As always the nonholonomic constraints are

$$\dot{x} = -r_c\cos(\theta)\dot{\phi} \quad (4.117)$$

$$\dot{y} = -r_c\sin(\theta)\dot{\phi}. \quad (4.118)$$

Leaving out η and ξ and rewriting the Lagrangian it looks like

$$\mathcal{L} = \frac{m}{2}\left[r_c^2\dot{\theta}^2\sin^2(\psi) + 2r_c^2\dot{\theta}\sin(\psi)\dot{\phi} + \dot{\phi}^2 + r_c^2\dot{\psi}^2\right] \quad (4.119)$$

$$+ \frac{1}{2}\left[I_\theta\left(\dot{\psi}^2 + \dot{\theta}^2\cos^2(\psi)\right) + I_\phi\left(\dot{\theta}\sin(\psi) + \dot{\phi}\right)^2\right] - mgr_c\cos(\psi), \quad (4.120)$$

[1] Changing the notation from the one in the book to using the variables already defined here

[2] When the constraints are held

Applying Eq. (3.1) on the 5 DOF system:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right) - \frac{\partial \mathcal{L}}{\partial x} = 0. \tag{4.121a}$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{y}}\right) - \frac{\partial \mathcal{L}}{\partial y} = 0. \tag{4.121b}$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\phi}}\right) - \frac{\partial \mathcal{L}}{\partial \phi} = \left(mr_c^2 + I_\phi\right)\left(\ddot{\phi} + \ddot{\theta}\sin(\psi) + \dot{\psi}\dot{\theta}\cos(\psi)\right) - 0. \tag{4.121c}$$

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\psi}}\right) - \frac{\partial \mathcal{L}}{\partial \psi} = {} & \left(2r_c^2 + I_\theta\right)\ddot{\psi} - I_\phi\dot{\theta}\cos(\psi)\left(\dot{\phi} + \dot{\theta}\sin(\psi)\right) \\ & - I_\theta\dot{\theta}^2\cos(\psi)\sin(\psi) + gmr_c\sin(\psi) \\ & + \dot{\theta}mr_c^2\cos(\psi)\left(\dot{\phi} + \dot{\theta}\sin(\psi)\right)\end{aligned} \tag{4.121d}$$

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}}\right) - \frac{\partial \mathcal{L}}{\partial \theta} = {} & I_\theta\ddot{\theta}\cos(\psi)^2 + I_\phi\ddot{\theta}\sin(\psi)^2 + I_\phi\ddot{\phi}\sin(\psi) \\ & + I_\phi\dot{\phi}\dot{\psi}\cos(\psi) + \ddot{\theta}mr_c^2\sin(\psi)^2 \\ & + \ddot{\phi}mr_c^2\sin(\psi) + 2I_\phi\dot{\psi}\dot{\theta}\cos(\psi)\sin(\psi) \\ & - 2I_\theta\dot{\psi}\dot{\theta}\cos(\psi)\sin(\psi) + \dot{\phi}\dot{\psi}mr_c^2\cos(\psi) \\ & + 2\dot{\psi}\dot{\theta}mr_c^2\cos(\psi)\sin(\psi) - 0.\end{aligned} \tag{4.121e}$$

In matrix form we get the matrices [3]

$$\underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & mr_c^2 + I_\phi & 0 & (mr_c^2 + I_\phi)\sin(\psi) \\ 0 & 0 & 0 & 2r_c^2 + I_\theta & 0 \\ 0 & 0 & \sin(\psi)(I_\phi + mr_c^2) & 0 & I_\theta \cos(\psi)^2 + \sin(\psi)^2(I_\phi + mr_c^2) \end{bmatrix}}_{\boldsymbol{M(q)}} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\phi} \\ \ddot{\psi} \\ \ddot{\theta} \end{bmatrix}$$

$$+ \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & (mr_c^2 + I_\phi)\cos(\psi)\dot{\psi} \\ 0 & 0 & 0 & 0 & \sin(2\psi)\dot{\theta}\,(I_\theta - I_\phi - mr_c^2) - \cos(\psi)\dot{\phi}\,(I_\phi + mr_c^2) \\ 0 & 0 & 0 & \cos(\psi)\dot{\phi}\,(I_\phi + mr_c^2) & \sin(2\psi)\dot{\psi}\,(I_\phi - I_\theta + mr_c^2) \end{bmatrix}}_{\boldsymbol{C(q,\dot{q})}} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \\ \dot{\psi} \\ \dot{\theta} \end{bmatrix}$$

$$+ \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ gmr_c \sin(\psi) \\ 0 \end{bmatrix}}_{\boldsymbol{G(q)}}$$

As this representation does not contain all the dynamics due to shortenings and simplifications the full dynamics will be calculated in the next section.

Method Two

In this section the full information, including $\dot{x}$ and $\dot{y}$ dynamics will be used in the calculations Decomposing the generalized coordinates into a wheel center frame:

$$x_c = x - r_c \sin(\psi)\sin(\theta), \tag{4.122a}$$
$$y_c = y + r_c \sin(\psi)\cos(\theta), \tag{4.122b}$$
$$z_c = r_c \cos(\psi) \tag{4.122c}$$

which can be differentiated

$$v_x = \dot{x} - \dot{\psi} r_c \cos(\psi)\sin(\theta) - \dot{\theta} r_c \cos(\theta)\sin(\psi), \tag{4.123a}$$
$$v_y = \dot{y} - \dot{\theta} r_c \sin(\psi)\sin(\theta) + \dot{\psi} r_c \cos(\psi)\cos(\theta), \tag{4.123b}$$
$$v_z = -\dot{\psi} r_c \sin(\psi) \tag{4.123c}$$

[3] $\sin(2\alpha) = \sin(\alpha)\cos(\alpha)$

Using the decomposed velocity vectors we find the velocity $v^2 = v_x^2 + v_y^2 + v_z^2$, and the rotational speed is the same as in [15].

$$\begin{aligned}\mathcal{L} = \frac{m}{2}\Bigg[&\Big(\dot{x} - \dot{\psi} r_c \cos(\psi)\sin(\theta) - \dot{\theta} r_c \cos(\theta)\sin(\psi)\Big)^2 + \\ &\Big(\dot{y} - \dot{\theta} r_c \sin(\psi)\sin(\theta) + \dot{\psi} r_c \cos(\psi)\cos(\theta)\Big)^2 + \Big(-\dot{\psi} r_c \sin(\psi)\Big)^2\Bigg] \\ &+ \frac{1}{2}\Big[I_\theta\left(\dot{\psi}^2 + \dot{\theta}^2\cos^2(\psi)\right) + I_\phi\left(\dot{\theta}\sin(\psi) + \dot{\phi}\right)^2\Big] - m g r_c \cos(\psi), \quad (4.124)\end{aligned}$$

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right) - \frac{\partial \mathcal{L}}{\partial x} &= \ddot{x} m + (\dot{\psi} m(2\dot{\psi} r_c \sin(\psi)\sin(\theta) - 2\dot{\theta} r_c \cos(\psi)\cos(\theta)))/2 \\ &+ (\dot{\theta} m(2\dot{\theta} r_c \sin(\psi)\sin(\theta) - 2\dot{\psi} r_c \cos(\psi)\cos(\theta)))/2 \\ &- \ddot{\theta} m r_c \cos(\theta)\sin(\psi) - \ddot{\psi} m r_c \cos(\psi)\sin(\theta). \\ &= m\Big(r_c \sin(\psi)\sin(\theta)\dot{\psi}^2 - 2 r_c \cos(\psi)\cos(\theta)\dot{\psi}\dot{\theta} + r_c \sin(\psi)\sin(\theta)\dot{\theta}^2 \\ &+ \ddot{x} - \ddot{\theta} r_c \cos(\theta)\sin(\psi) - \ddot{\psi} r_c \cos(\psi)\sin(\theta)\Big) \\ \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{y}}\right) - \frac{\partial \mathcal{L}}{\partial y} &= -m\Big(r_c \cos(\theta)\sin(\psi)\dot{\psi}^2 + 2 r_c \cos(\psi)\sin(\theta)\dot{\psi}\dot{\theta} + r_c \cos(\theta)\sin(\psi)\dot{\theta}^2 \\ &- \ddot{y} + \ddot{\theta} r_c \sin(\psi)\sin(\theta) - \ddot{\psi} r_c \cos(\psi)\cos(\theta)\Big). \\ \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\phi}}\right) - \frac{\partial \mathcal{L}}{\partial \phi} &= I_\phi\left(\ddot{\phi} + \ddot{\theta}\sin(\psi) + \dot{\psi}\dot{\theta}\cos(\psi)\right). \\ \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\psi}}\right) - \frac{\partial \mathcal{L}}{\partial \psi} &= I_\theta \ddot{\psi} + \ddot{\psi} m r_c^2 - g m \sin(\psi) + \ddot{y} m r_c \cos(\psi)\cos(\theta) - \ddot{x} m r_c \cos(\psi)\sin(\theta) \\ &- 2\dot{\theta}^2 m r_c \cos(\psi)\sin(\psi) - 2\dot{\phi}\dot{\theta} m r_c \cos(\psi) - \dot{\theta}\dot{x} m r_c \cos(\psi)\cos(\theta) \\ &- \dot{\psi}\dot{y} m r_c \cos(\theta)\sin(\psi) - \dot{\theta}\dot{y} m r_c \cos(\psi)\sin(\theta) + \dot{\psi}\dot{x} m r_c \sin(\psi)\sin(\theta) \\ \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}}\right) - \frac{\partial \mathcal{L}}{\partial \theta} &= I_\phi \ddot{\theta} + \ddot{\theta} m r_c^2 - I_\phi \ddot{\theta}\cos(\psi)^2 + I_\theta \ddot{\theta}\cos(\psi)^2 + I_\phi \ddot{\phi}\sin(\psi) \\ &- \ddot{\theta} m r_c^2 \cos(\psi)^2 + I_\phi \dot{\phi}\dot{\psi}\cos(\psi) + 2 I_\phi \dot{\psi}\dot{\theta}\cos(\psi)\sin(\psi) \\ &- 2 I_\theta \dot{\psi}\dot{\theta}\cos(\psi)\sin(\psi) - \ddot{x} m r_c \cos(\theta)\sin(\psi) - \ddot{y} m r_c \sin(\psi)\sin(\theta) \\ &+ 2\dot{\psi}\dot{\theta} m r_c^2 \cos(\psi)\sin(\psi),\end{aligned}$$

where on a shape as in Eq. (4.7) it looks like

$$
\underbrace{\begin{bmatrix} m & 0 & 0 & -mr_c\cos(\psi)\sin(\theta) & -mr_c\cos(\theta)\sin(\psi) \\ 0 & m & 0 & mr_c\cos(\psi)\cos(\theta) & -mr_c\sin(\theta)\sin(\psi) \\ 0 & 0 & I_\phi & 0 & I_\phi\sin(\psi) \\ -mr_c\cos(\psi)\sin(\theta) & mr_c\cos(\psi)\cos(\theta) & 0 & mr_c^2+I_\theta & 0 \\ -mr_c\sin(\psi)\cos(\theta) & -mr_c\sin(\psi)\sin(\theta) & I_\phi\sin(\psi) & 0 & I_\phi+mr_c^2-I_\phi\cos(\psi)^2+I_\theta\cos(\psi)^2-mr_c^2\cos(\psi)^2 \end{bmatrix}}_{\boldsymbol{M(q)}} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\phi} \\ \ddot{\psi} \\ \ddot{\theta} \end{bmatrix}
$$

$$
+ \underbrace{\begin{bmatrix} 0 & 0 & 0 & mr_c(\dot{\psi}\sin(\psi)\sin(\theta)-2\dot{\theta}\cos(\psi)\cos(\theta)) & mr_c\dot{\theta}\sin(\psi)\sin(\theta) \\ 0 & 0 & 0 & -mr_c(\dot{\psi}\cos(\theta)\sin(\psi)+2\dot{\theta}\cos(\psi)\sin(\theta)) & -mr_c\dot{\theta}\cos(\theta)\sin(\psi)) \\ 0 & 0 & 0 & I_\phi\dot{\theta}\cos(\psi) & 0 \\ 0 & 0 & -I_\phi\dot{\theta}\cos(\psi) & 0 & -\frac{1}{2}\left(\dot{\theta}\sin(2\psi)\left(mr_c^2+I_\phi-I_\theta\right)\right) \\ 0 & 0 & 0 & \sin(2\psi)\dot{\theta}(mr_c^2+I_\phi-I_\theta)+I_\phi\dot{\phi}\cos(\psi) & 0 \end{bmatrix}}_{\boldsymbol{C(q,\dot{q})}} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \\ \dot{\psi} \\ \dot{\theta} \end{bmatrix}
$$

$$
+ \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ -gmr_c\sin(\psi) \\ 0 \end{bmatrix}}_{\boldsymbol{G(q)}}
$$

By including the nonconservative forces and input forces at the right hand side of the equation;

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -r_c\cos(\theta) & -r_c\sin(\theta) \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{\boldsymbol{J}^T(\boldsymbol{q})} \underbrace{\begin{bmatrix} \lambda_{c1} \\ \lambda_{c2} \end{bmatrix}}_{\boldsymbol{\lambda}} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}}_{\boldsymbol{B}} \underbrace{\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}_{\boldsymbol{u}} \tag{4.126}$$

To get the system in a format that is favorable for calculating the $\boldsymbol{\alpha}(\phi)$, $\boldsymbol{\beta}(\phi)$, $\boldsymbol{\gamma}(\phi)$ equations the method from Sec. 4.2.1 is used.

$$\begin{aligned} &\boldsymbol{B}_\perp(\boldsymbol{q})\Big[\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \Big(\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{J}_M(\boldsymbol{q})\dot{\boldsymbol{J}}(\boldsymbol{q}) - \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{J}_*\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\Big)\dot{\boldsymbol{q}} + \\ &\Big(\boldsymbol{I} - \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{J}_*(\boldsymbol{q})\Big)\boldsymbol{G}(\boldsymbol{q})\Big] = \boldsymbol{B}_\perp(\boldsymbol{q}) \underbrace{\Big(\boldsymbol{I} - \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{J}_*(\boldsymbol{q})\Big)\boldsymbol{B}}_{\boldsymbol{B}_\lambda}\boldsymbol{u}, \end{aligned} \tag{4.127}$$

where $\boldsymbol{B}_\perp$ is a matrix that annihilates $\boldsymbol{B}_\lambda$. Substituting such that $\begin{bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \\ \ddot{\boldsymbol{q}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}(s) \\ \boldsymbol{\Phi}'(s)\dot{s} \\ \boldsymbol{\Phi}''(s)\dot{s}^2 + \boldsymbol{\Phi}'(s)\ddot{s} \end{bmatrix}$;

$$\begin{aligned} &\boldsymbol{B}_\perp(\boldsymbol{\Phi}(s))\Big[\boldsymbol{M}(\boldsymbol{\Phi}(s))(\boldsymbol{\Phi}''(s)\dot{s}^2 + \boldsymbol{\Phi}'(s)\ddot{s}) + \Big(\boldsymbol{C}(\boldsymbol{\Phi}(s),\boldsymbol{\Phi}'(s)\dot{s}) \\ &+ \boldsymbol{J}^T(\boldsymbol{\Phi}(s))\boldsymbol{J}_M(\boldsymbol{\Phi}(s))\dot{\boldsymbol{J}}(\boldsymbol{\Phi}(s)) \\ &- \boldsymbol{J}^T(\boldsymbol{\Phi}(s))\boldsymbol{J}_*(\boldsymbol{\Phi}(s))\boldsymbol{C}(\boldsymbol{\Phi}(s),\boldsymbol{\Phi}'(s)\dot{s})\Big)\boldsymbol{\Phi}'(s)\dot{s} - \\ &\Big(\boldsymbol{I} - \boldsymbol{J}^T(\boldsymbol{\Phi}(s))\boldsymbol{J}_*(\boldsymbol{\Phi}(s))\Big)\boldsymbol{G}(\boldsymbol{\Phi}(s))\Big] = \boldsymbol{0}, \end{aligned} \tag{4.128}$$

where s is a general parameter for the synchronization function $\boldsymbol{\Phi}(s)$ which for now also is general and not path specific. Because $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ is linear in $\dot{\boldsymbol{q}}$, $\boldsymbol{C}(\boldsymbol{\Phi}(s),\boldsymbol{\Phi}'(s)\dot{s})$ can be written as $\boldsymbol{C}(\boldsymbol{\Phi}(s),\boldsymbol{\Phi}'(s))\dot{s}$. Eq. (4.30) shows that $\dot{\boldsymbol{J}}(\boldsymbol{q}) = \sum_i \frac{\partial \boldsymbol{J}}{\partial q_i}\dot{q}_i$, which can be rewritten with the substitution as $\dot{\boldsymbol{J}}(\boldsymbol{\Phi}(s)) =$

$\sum_i \frac{\partial \boldsymbol{J}}{\partial \Phi_i(s)} \Phi_i'(s)\dot{s}$. That then yields

$$\begin{aligned}
&\boldsymbol{B}_\perp(\boldsymbol{\Phi}(s)) \Big[\boldsymbol{M}(\boldsymbol{\Phi}(s))\boldsymbol{\Phi}'(s)\ddot{s} + \Big(\boldsymbol{M}(\boldsymbol{\Phi}(s))\boldsymbol{\Phi}''(s) + \Big(\boldsymbol{C}(\boldsymbol{\Phi}(s), \boldsymbol{\Phi}'(s)) \\
&+ \boldsymbol{J}^T(\boldsymbol{\Phi}(s))\boldsymbol{J}_M(\boldsymbol{\Phi}(s)) \left(\sum_{i=1}^{n} \frac{\partial \boldsymbol{J}}{\partial \Phi_i(s)} \Phi_i'(s) \right) \\
&- \boldsymbol{J}^T(\boldsymbol{\Phi}(s))\boldsymbol{J}_*(\boldsymbol{\Phi}(s))\boldsymbol{C}(\boldsymbol{\Phi}(s), \boldsymbol{\Phi}'(s)) \Big) \boldsymbol{\Phi}'(s) \Big) \dot{s}^2 \\
&- \Big(\boldsymbol{I} - \boldsymbol{J}^T(\boldsymbol{\Phi}(s))\boldsymbol{J}_*(\boldsymbol{\Phi}(s)) \Big) \boldsymbol{G}(\boldsymbol{\Phi}(s)) \Big] = \boldsymbol{0}, \qquad (4.129)
\end{aligned}$$

where $\boldsymbol{J}_*$ and $\boldsymbol{J}_M$ are constructed as in Eq. (4.14).

The following relation can then be made

$$\boldsymbol{\alpha}(s)\ddot{s} + \boldsymbol{\beta}(s)\dot{s}^2 + \boldsymbol{\gamma}(s) = \boldsymbol{0}, \qquad (4.130)$$

where

$$\boldsymbol{\alpha}(s) = \boldsymbol{B}_\perp(s)\boldsymbol{M}(s)\boldsymbol{\Phi}'(s) \qquad (4.131)$$

$$\begin{aligned}
\boldsymbol{\beta}(s) = &\boldsymbol{B}_\perp(s) \Bigg[\boldsymbol{M}(s)\boldsymbol{\Phi}''(s) + \Big[\boldsymbol{C}(s) + \boldsymbol{J}^T(s)\boldsymbol{J}_M(s) \Big(\sum_{i=1}^{n} \frac{\partial \boldsymbol{J}}{\partial \Phi_i(s)} \Phi_i'(s) \Big) \\
&- \boldsymbol{J}^T(s)\boldsymbol{J}_*(s)\boldsymbol{C}(s) \Big] \boldsymbol{\Phi}'(s) \Bigg] \qquad (4.132)
\end{aligned}$$

$$\boldsymbol{\gamma}(s) = -\boldsymbol{B}_\perp(s) \Big(\boldsymbol{I} - \boldsymbol{J}^T(s)\boldsymbol{J}_*(s) \Big) \boldsymbol{G}(s). \qquad (4.133)$$

As can be seen from the equations above, the 5 DOF system is quite more challenging to describe and it is even more challenging to control. As the $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$ equations are non-zero it is expected that the parameter $s, \dot{s}, \ddot{s}$ must be chosen in accordance to certain configurations of the system. This can be shown by picking $\Phi(s)$ as in Sec. 4.2, where θ is chosen the parameter for the synchronization function and the orbital circle is the path. The initial parameters are

$$\begin{aligned}
I_\phi = 0.2, \quad I_\theta = 0.4, \quad m = 2, \quad g = 9.81, \\
r_c = 0.25, \quad r_p = 2 \quad \psi_0 = \frac{20\pi}{180} \qquad (4.134)
\end{aligned}$$

Calculating $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$ it was seen that for a circular path, α were constant $= \boldsymbol{0}$. This means that

$$\boldsymbol{\beta}(\theta)\dot{\theta}^2 + \boldsymbol{\gamma}(\theta) = \boldsymbol{0} \rightarrow \dot{\theta} = \sqrt{-\frac{\boldsymbol{\gamma}(\theta)}{\boldsymbol{\beta}(\theta)}} \qquad (4.135)$$

For the calculations θ was chosen to be a vector representing a full orbit, i.e starting at $\theta = 0$ and ending at $\theta = 2\pi$. Using $\boldsymbol{\beta}(\theta)$ and $\boldsymbol{\gamma}(\theta)$, $\dot{\theta}$ was calculated for different values of θ over the orbit. The results are shown in Fig. 4.5, 4.6 and 4.7. Here it can be seen that $\alpha_1(\theta) = \alpha_3(\theta) = 0$, which also is the case for $\alpha_2(\theta)$. Fig. 4.5 and 4.7 shows how β and γ evolves and how the product $\beta_i\dot{\theta}^2 + \gamma_i = 0$, as expected. Fig. 4.6 shows that the velocity profile for θ is constant for the circular orbit and a stationary $\psi = \psi_0$. This led to Fig. 4.8 where the same calculations have been done, but with different initial values for ψ. It can be seen that as the initial tilt increases, the velocity, $\dot{\theta}$ also increases in order to keep the tilt constant. This correspond good with the real world, as a coin rolling in a circle and falling further and further down increase its speed.

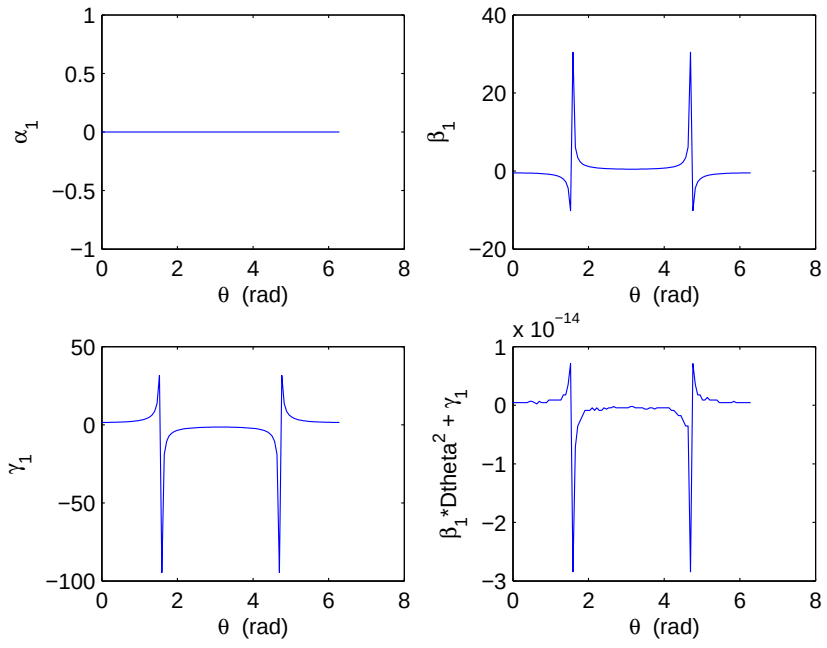


Figure 4.5: $\alpha_1, \beta_1, \gamma_1$ values.

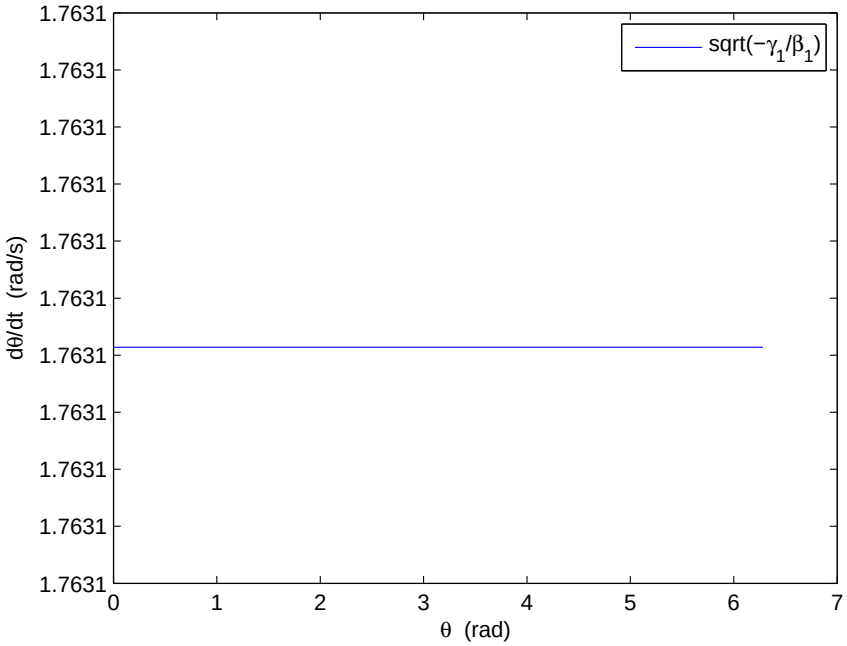


Figure 4.6: Turning velocity profile, $\dot{\theta}(\theta)$, for unicycle.

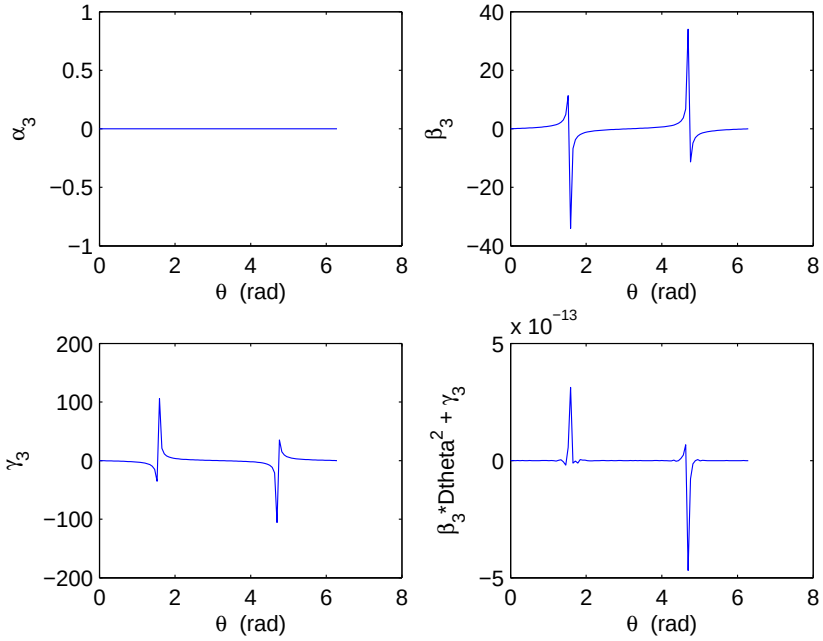


Figure 4.7: $\alpha_3, \beta_3, \gamma_3$ values.

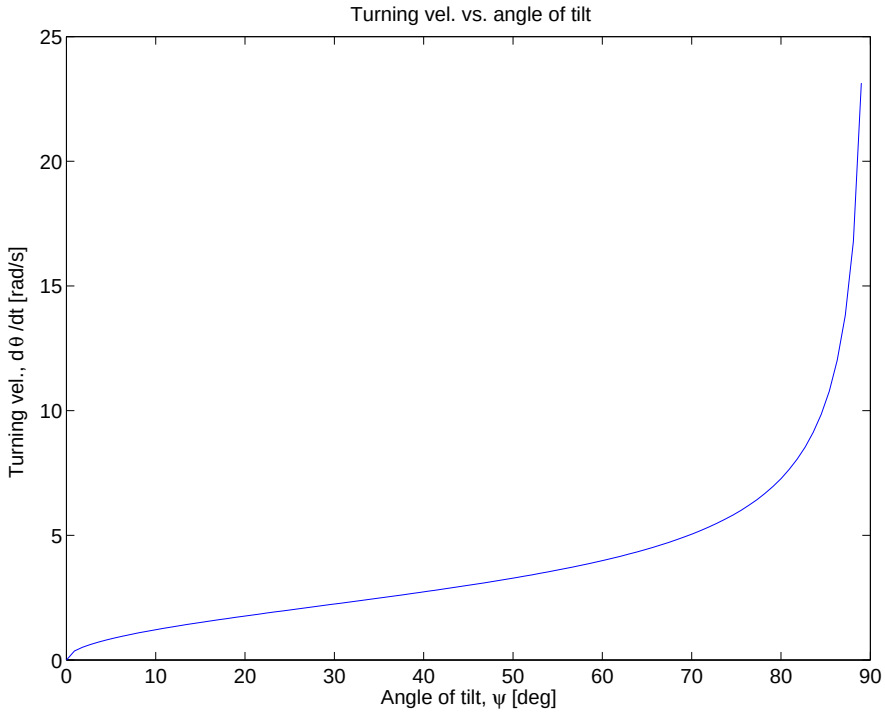


Figure 4.8: Turning velocity vs. tilt angle ψ for circular orbit.

Chapter 5

Implementation and Simulation of the Four DOF system

5.1 Implementation in SIMULINK

A testbed for the system was implemented in MATLAB. The function of this testbed was to run simulations of the system with various controllers and initial conditions.

5.1.1 System

Fig. 5.1 shows the overall system implemented in SIMULINK. The system consist of two subsystems; a part where the dynamics are calculated, see Fig. 5.2, and a part where the nominal and the controller inputs are calculated, see Fig. 5.3. In Fig. 5.1 it can be seen that the dynamics are calculated using the input u in addition to the states. The controllers input is sthe states, and uses those to calculate the input.

In Fig. 5.2 it can be seen how the states are calculated. The function `d2q(q,dq,u)` takes in the states $\boldsymbol{q}, \dot{\boldsymbol{q}}$ and the input $\boldsymbol{u}$ and then it use Eq. (4.16) to calculate $\ddot{\boldsymbol{q}}$ based on the inputs. The other states, $\boldsymbol{q}, \dot{\boldsymbol{q}}$ are then simply integrals of $\ddot{\boldsymbol{q}}$.

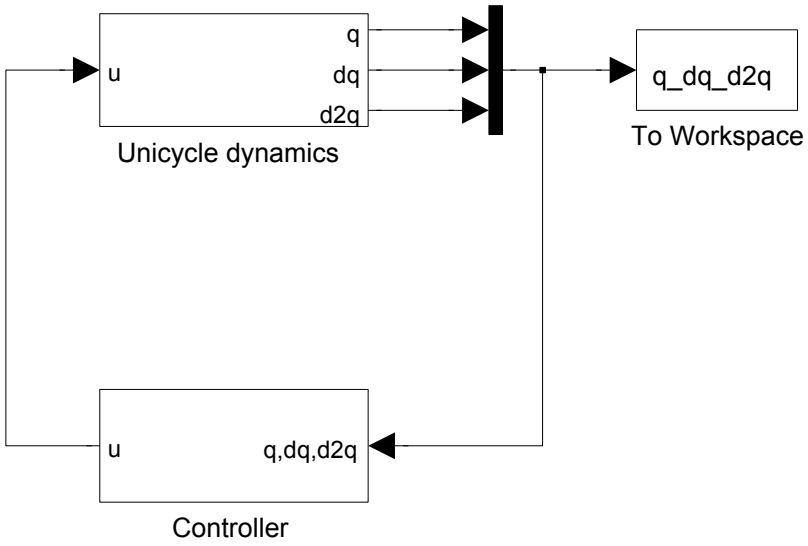


Figure 5.1: The overall SIMULINK system.

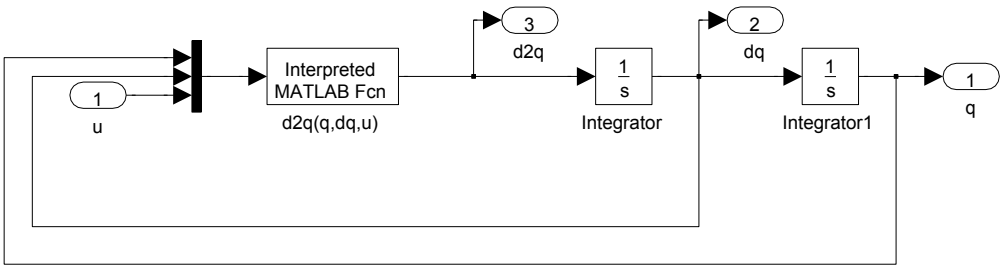


Figure 5.2: The dynamics of the system.

5.1.2 Controller

The controller part of the system is somewhat more extensive. The input to the controller is the states and the output is the inputs to the system, $\boldsymbol{u} = \boldsymbol{U} + \boldsymbol{v}$. The function `calc_tau(q,dq,d2q)` uses the parameter for the synchronization function and interpolates to find the time reference, τ, that corresponds best with the current state. This is then used to calculate the optimal input at that specific state in `get_U(q,dq,u)` and also to find the optimal controller gain from the time varying controller using the function `controller_from_opt(t)`. The function `x_perp_calc(q,dq,d2q)` is used to calculate the vector $\Delta\boldsymbol{x}_\perp = [\Delta I;\ \Delta\boldsymbol{z};\ \Delta\dot{\boldsymbol{z}}]$. The controller gains, $\boldsymbol{K}$, are then multiplied with $\Delta\boldsymbol{x}_\perp$ to form the control input $\boldsymbol{v}$. Then $\boldsymbol{U}$ and $\boldsymbol{v}$ are summed and used by the dynamics subsystem.

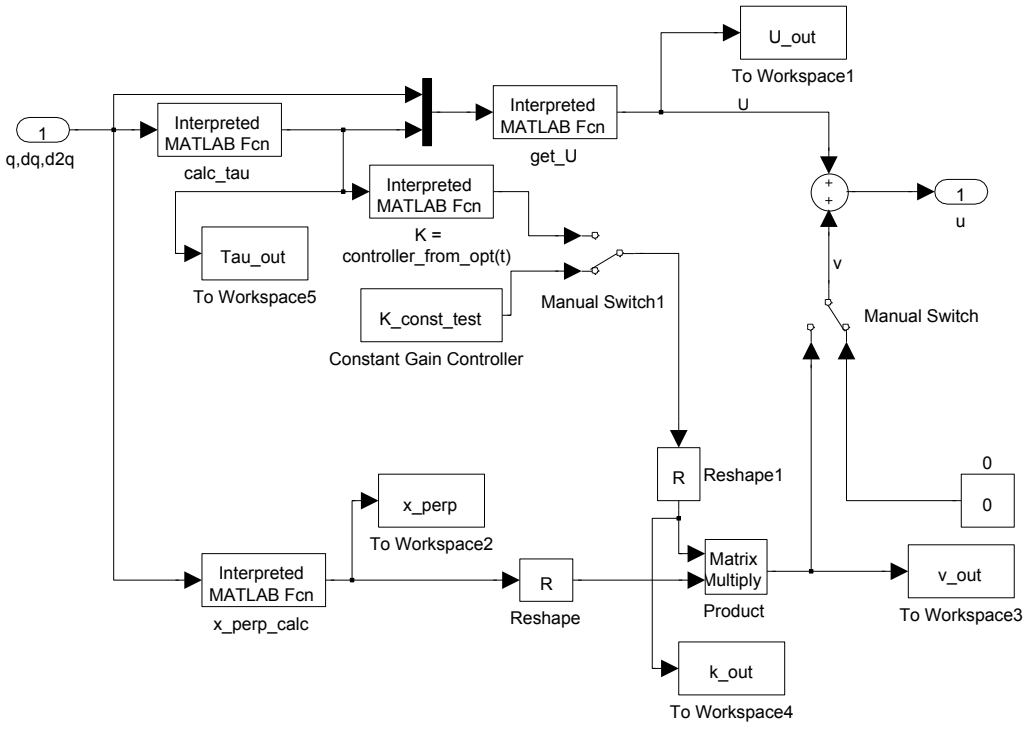


Figure 5.3: The system controller.

5.2 Running Simulations

This section will look into some simulations done with different controllers and different initial conditions. This to better get an overview of how the system reacts to different inputs and how the system evolves. All the simulations are done with the four DOF system and with $\Phi(\theta)$ representing a circular orbit as described in Sec. 4.2. The parameters used for the system are

$$I_\phi = 0.2, \quad I_\theta = 0.4, \quad m = 2, \quad g = 9.81 \quad r_c = 0.25, \quad r_p = 2.$$

All the simulations are done over a period of $3T$, which is 15 seconds with 5 seconds periods.

5.2.1 No Controller

The natural dynamics of the system can be shown by simulating using no controller and let the dynamics run for itself. This way makes it easier to measure the performance of a controller, as this will act as a reference for the rest.

No Initial Error

A simulation with no initial error was made. The initial path parameters were set to be

$$\begin{gathered} x_0 = 0, \quad y_0 = 0, \quad \phi_0 = 0, \quad \theta_0 = 0, \\ \dot{\theta}_0 = \frac{2\pi}{T}\bigg|_{T=5s} = 1.2566, \quad \dot{\phi}_0 = \dot{\theta}_0 \frac{r_p}{r_c}, \quad \dot{x}_0 = r_c \dot{\phi}_0 \cos(\theta_0), \quad \dot{y}_0 = r_c \dot{\phi}_0 \sin(\theta_0), \end{gathered} \tag{5.1}$$

where the velocities are set such that they does not violate the nonholonomic constraints. As there are no initial errors,

$$I = \boldsymbol{z} = \dot{\boldsymbol{z}} = \mathbf{0}. \tag{5.2}$$

Parameter	Initial Error
I	0
z_1	0.2
z_2	-0.2
z_3	0
$\dot{z}_1$	0
$\dot{z}_2$	0
$\dot{z}_3$	0.2

Table 5.1: Initial errors, simulating with no controller.

With Initial Error

Initial error for the system are introduced. The path parameters are set as in the previous chapter, only now there is introduced error as shown in Tab. 5.1.

5.2.2 Constant Gain Controller With Initial Error

The controller for the system is given by $\boldsymbol{v} = \boldsymbol{K}\boldsymbol{x}_\perp$, $K \in \mathbb{R}^{2\times 7}$ which gives 14 different control parameters. All of the control parameters will have an impact on the resulting gain, i.e it is not a trivial way to select some parameters to be weighted more than others. That is why the constant gain controller matrix will consist of 14 equal gains.

5.2.3 Time Varying Controller With Initial Error

The controller found with the Bézier method from Sec. 4.3.5 is now used. Many hours are spent trying to find an optimal time varying controller. Different cost functions for the controller are used and an extensive set of different initial controller points are tried, but no optimal controller were found. The best one found will be used in this simulation and the initial conditions are the same as before.

Chapter 6

Simulation Results

6.1 Simulating with no controller

6.1.1 No initial error

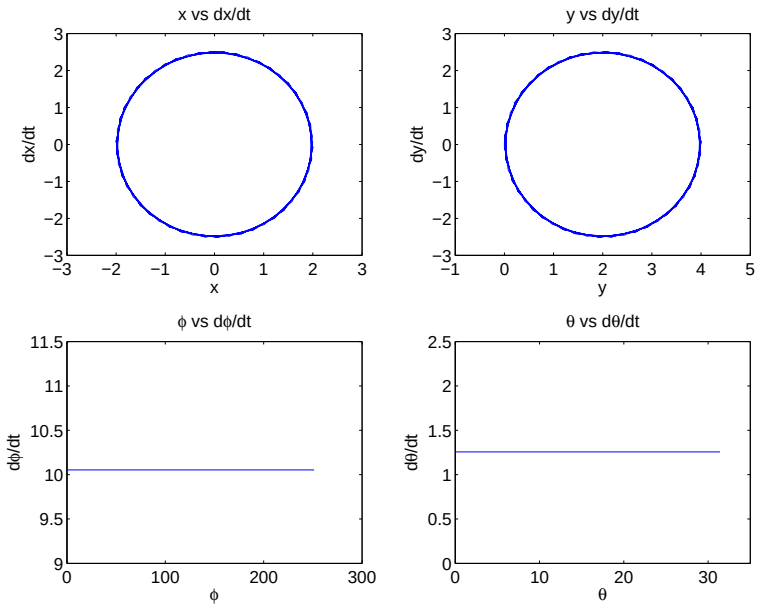


Figure 6.1: Simulation with no controller, no initial error, showing $\boldsymbol{q}$ vs. $\dot{\boldsymbol{q}}$.

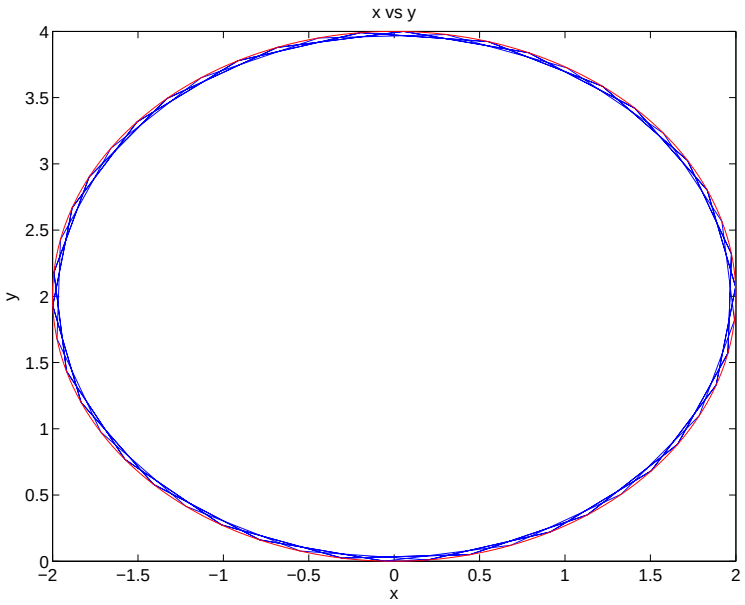


Figure 6.2: Simulation with no controller, no initial error, showing x vs. y.

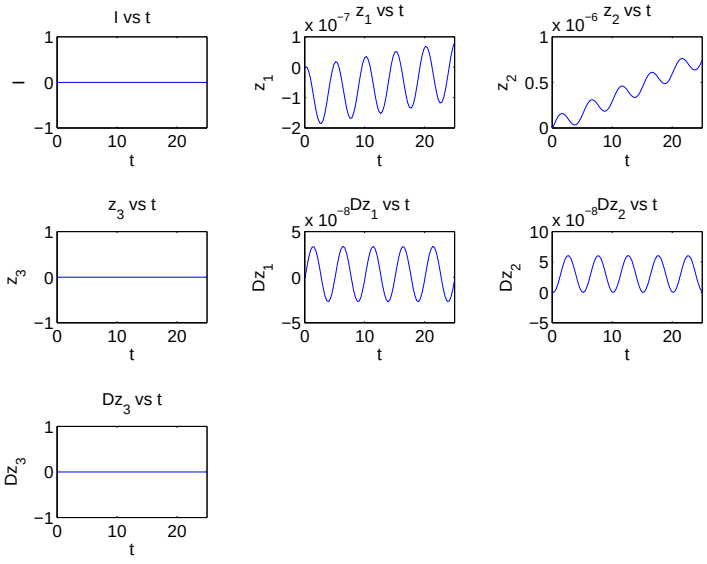


Figure 6.3: Simulation with no controller, no initial error, showing $\Delta \boldsymbol{x}_{\perp}$.

Fig. 6.1, 6.2 and 6.3 shows the simulation results when no initial errors are present and no controllers are influencing the system.. It can be seen from Fig. 6.3 that the error in I, z_3 and $\dot{z}_3$ start and remain at zero the whole time. $\dot{z}_1$ and $\dot{z}_2$ start at zero but oscillates, with no trend to drift. z_1 and z_2 also starts at zero and oscillates, but have a slight drift. The drift is measured as seen in Fig. 6.4 and the drift for z_2 is measured to be $0.8 \cdot 10^{-6}$ and for z_1 it is $0.5 \cdot 10^{-7}$. This can be interpreted as $\dot{z}_1$ and $\dot{z}_2$ not oscillating around zero but some positive offset which yield a drift in z_1 and z_2. The phase plots in Fig. 6.1 looks ok, but some of the errors can be seen in Fig. 6.2 where the red line represents the desired path and the blue is the actual one.

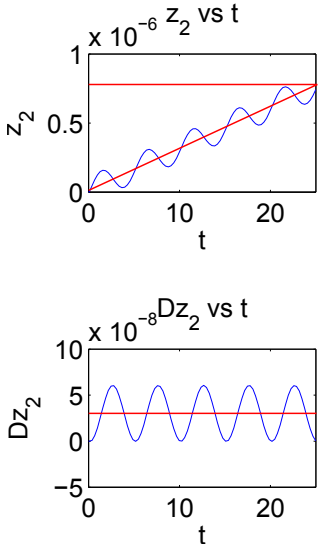


Figure 6.4: How the drift is measured and the offset can be seen.

6.1.2 With initial error

An initial error, seen in Tab. 5.1, is now present. The results can be seen in Fig. 6.5, 6.6 and 6.7. Fig. 6.5 shows that still I starts and stays at zero, as expected, $\dot{z}_3$ starts and stays at its initial error causing z_3 to drift away from its starting point at 0. $\dot{z}_1$ starts at zero but oscillates around -0.05 causing z_1 to drift in a negative direction. It can also be seen that $\dot{z}_2$ starts in and oscillates around 0, causing no drift in z_2 which only has got an offset. The offsets in and drifts can be seen in Fig. 6.6 as a mismatch between the blue and the red circle. As all the plots take a lot of space and can be hard to read these plots will show examples of the link between the plots. The plots

with the most information is the ones showing the error in $I, \boldsymbol{z}, \dot{\boldsymbol{z}}$, and those will be used in the next sections.

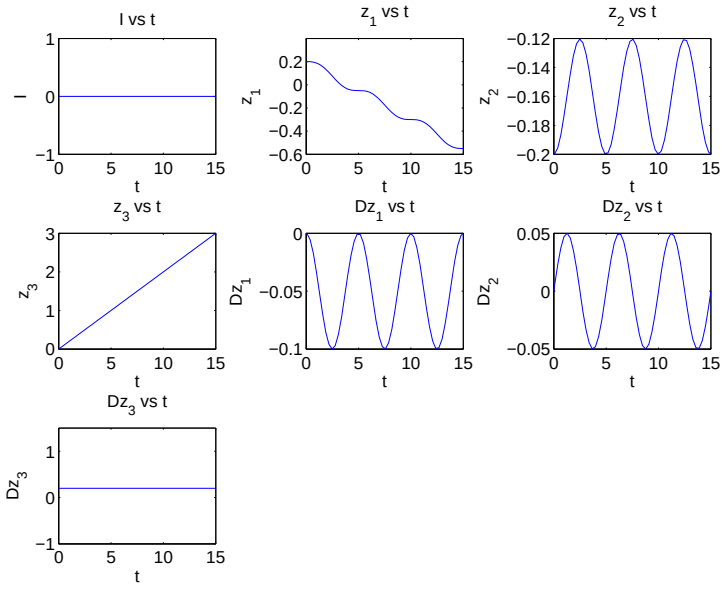


Figure 6.5: Simulation with no controller, initial error from Tab. 5.1, showing $\Delta \boldsymbol{x}_{\perp}$.

6.2 Simulating Using Constant Gain Controller With Initial Error

There was done trials with four different gain controllers. The results using the different controllers were very much alike and therefore this section will only present two of them. Fig. 6.8 shows the results using a constant gain on all control variables equal to one, while Fig. 6.9 shows the result using a gain equal to 100. As seen in the two figures, the errors are very similar, but in Fig. 6.9 the result of a more aggressive controller can be seen. The error in I now starts at 0, but quickly rises and ends up oscillating around 0.05. $\dot{z}_3$ starts at zero, but also develops an oscillation around 0.08. This causes a drift in z_3 which starts at zero but drifts ~ 0.8 in 3 periods. $\dot{z}_1$ has an initial

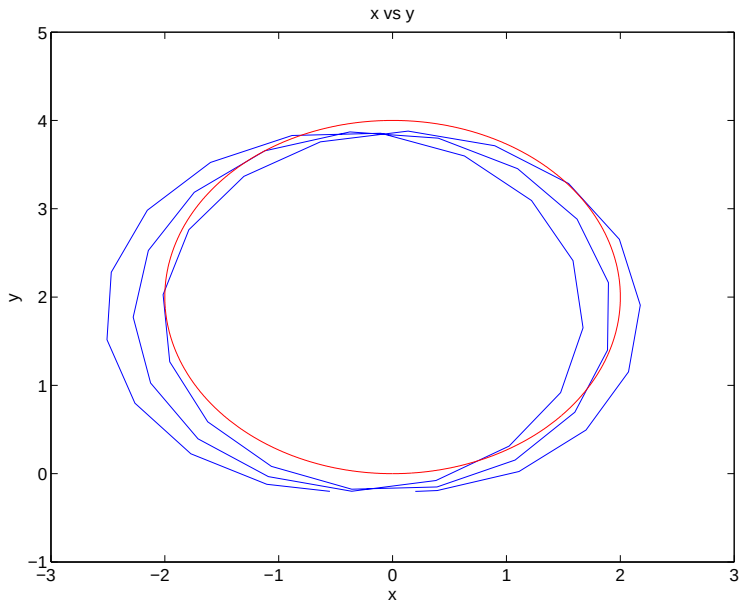


Figure 6.6: Simulation with no controller, initial error from Tab. 5.1, showing x vs. y.

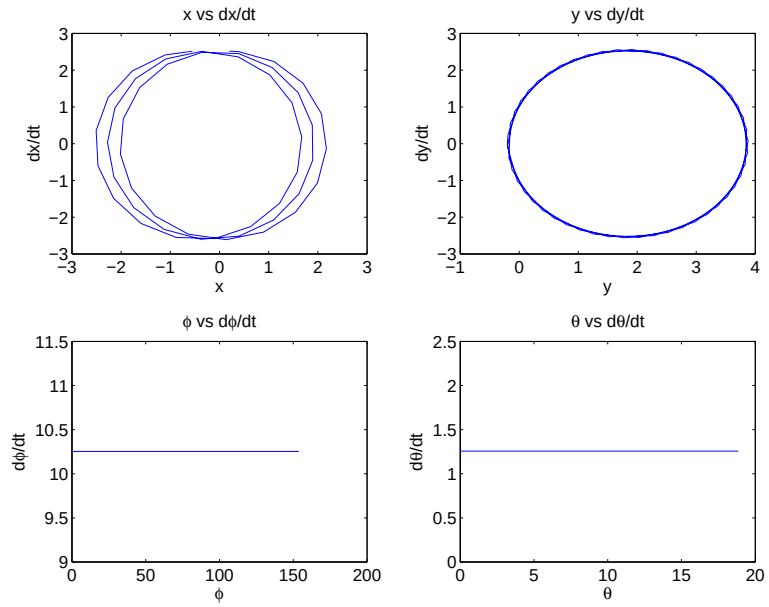


Figure 6.7: Simulation with no controller, initial error from Tab. 5.1, showing $\boldsymbol{q}$ vs. $\dot{\boldsymbol{q}}$.

value at 0, but quickly gets a stable oscillation around -0.05. This causes z_1 to drift ~ -0.8 in 3 periods from initial value as given in Tab. 5.1. $\dot{z}_2$ starts in zero and oscillates somewhere right below zero, which causes a drift in z_2 at ~ -0.03 in 3 periods.

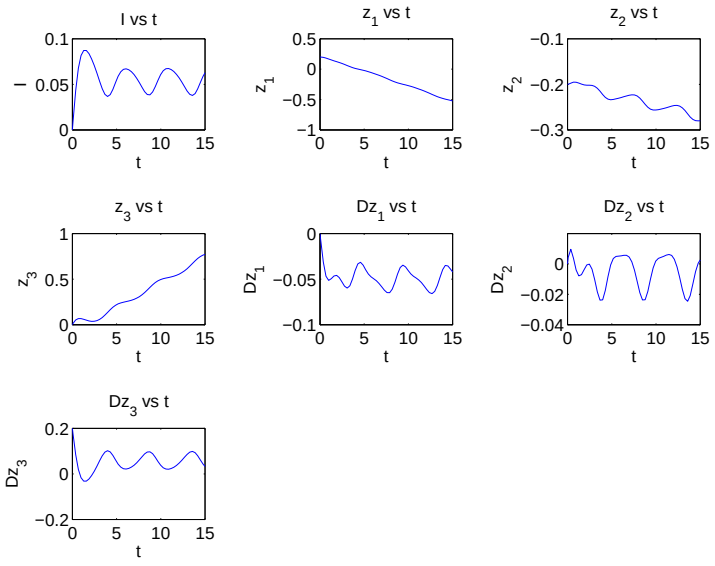


Figure 6.8: Simulation with controller gains = 1, initial error from Tab. 5.1, showing $\Delta \boldsymbol{x}_\perp$.

A small negative gain was also tried out, but caused the states to blow up in under 0.1 seconds, and the results were therefore discarded.

6.3 Simulating Using Time Dependent Controller With Initial Error

The best controller found in a lot of trials were used and the result can be viewed in Fig. 6.10. There it can be seen how the error, despite some ugly peaks, seems to converge. Neither of the states seems to blow up, they actually seems to slowly decrease in magnitude. This led to another test being done, using the same controller, but with no initial error present. The

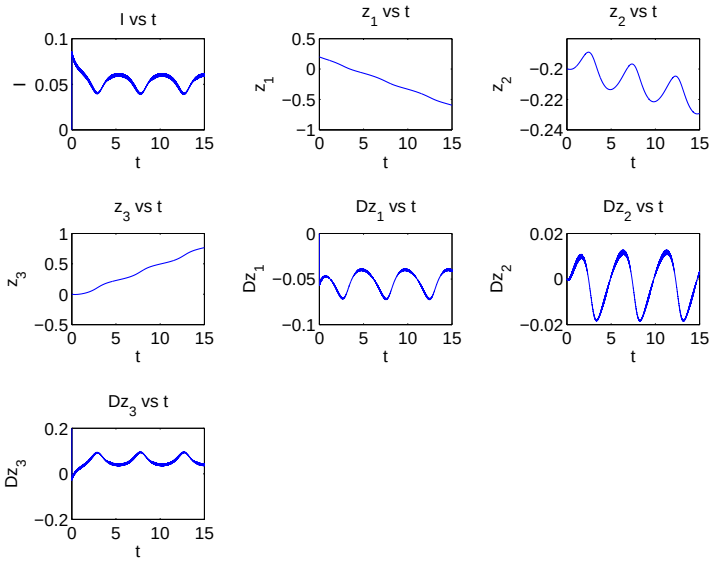


Figure 6.9: Simulation with controller gains = 100, initial error from Tab. 5.1, showing $\Delta \boldsymbol{x}_{\perp}$.

simulation results for 20 periods can be seen in Fig. 6.11. This clearly shows how the states go from zero initial error, to a growing error for every period.

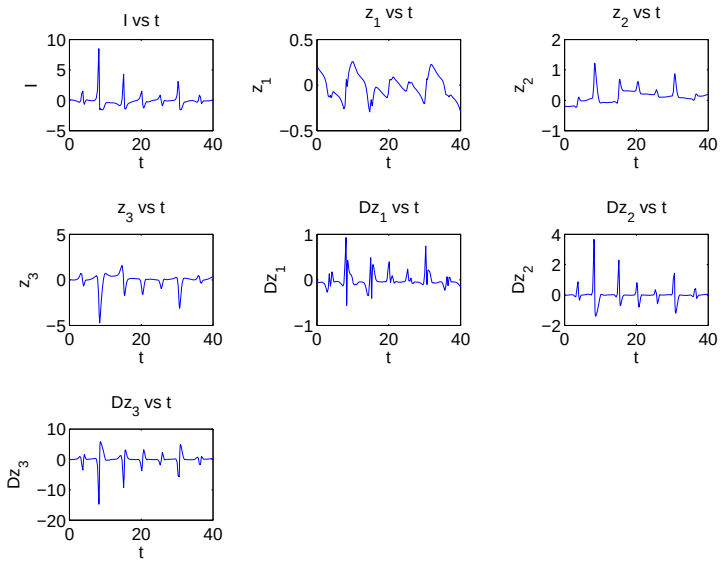


Figure 6.10: Simulation with time dep. controller, initial error from Tab. 5.1, showing $\Delta \boldsymbol{x}_{\perp}$.

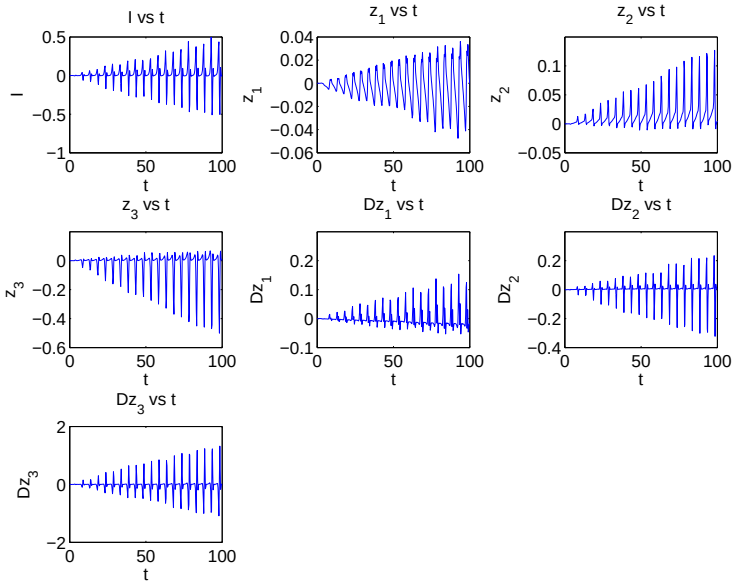


Figure 6.11: Simulation with time dep. controller, no initial error, showing $\Delta \boldsymbol{x}_{\perp}$.

Chapter 7

Discussion

7.1 Analysis of the nonholonomic system

It was shown in Ch. 4 that there are at least two different ways to calculate the equations of motion for a unicycle. The one using four generalized coordinates yield a much simpler dynamic that with the one using five. For the four DOF system it was shown that both the circular and the straight path could be parametrized as wanted. No constraints on the pair $(\theta, \dot{\theta})$ and $(\phi, \dot{\phi})$ were given. This is probably because the potential energy due to gravity not is present and the coin will never "fall" no mater how fast it turns. On the other would friction forces show up, causing the unicycle to skid if the velocity got too high in for instance a turn. This problem was looked into in the project [3], and the path planner from there would generate feasible paths when taking skidding into consideration.
For the five DOF system it was shown that for different tilt angles, the velocity of the unicycle had to increase with the tilt in a non linear way. This is due to the center of gravity that moves with the tilt. An increase in speed would increase the forces due to centripetal acceleration, and thus keep the unicycle from falling.

7.2 Simulations of the four DOF system

The results of the simulations performed in Sec. 5.2 was shown in Ch. 6.1. When simulating with no initial errors it was seen in Fig. 6.3 that there was a drift in some of the states and even though initialized in zero, some of the

states ended up oscillating around non-zero values. Probably this is due to simulation inaccuracies as a result of to big steps in the simulation, even though they were regulated to not be so big (max step size were 0.2). The drift was present in two states, z_1 and z_2, representing the error in x and y. The error seemed to be increasing, but at a very small rate. Ideally the trial where no controller and no initial errors were present, the system would have been stable. But it can be seen that as time goes the states increases, which is characterized as unstable.

When the system is initialized with errors and no controllers are present it was seen in Fig. 6.5 that some of the states grew with time. As $\dot{z}_3$ was initialized with an error it was expected that z_3 grew unbounded. A slightly less but still a significant error growth is the one in z_1. Its derivative, $\dot{z}_1$ has a negative offset in its oscillations, which explain why z_1 is drifting. The reason to the offset in $\dot{z}_1$ has probably something to do with the initial error in $\dot{z}_3$ which has an impact on $\dot{z}_1$ and $\dot{z}_2$. Still, it does not explain why there is no offset in $\dot{z}_2$. It can be seen that the error of several states will grow with time and the system will probably get unstable with time.

The simulation with the constant gain controllers was shown in Fig. 6.8 and 6.9. It could be seen, compared to the system with no controller, Fig. 6.5, that all the velocity errors $\boldsymbol{z}$ had lower magnitude and looked stable. The lower amplitude led to a slower drift in z_1 and z_3, but also led to drift in z_2 and an offset in I. The introduction of an error in I is because of the controller varying the speed of $\dot{\theta}$, which wasn't possible without any controller. Even though a small drift was introduced in z_2, the drift in z_1 and z_3 was decreased more, and the oscillations in $\dot{z}_2$ and $\dot{z}_1$ was damped more than the introduction of oscillations in $\dot{z}_3$ added. All together the controller with constant gains performed slightly better then no controller, but the system is still unstable.

When introducing the time varying controller described in Sec. 4.3.5 the errors were in Fig. 6.10 shown to be decreasing or not increasing over time. Such a behavior would yield a marginally stable system, as the output were bounded. When the same controller was tested without any initial errors, it failed and the errors were oscillating and increasing over time, see Fig. 6.11. This indicates that the controller not stabilizes the system and that it is very little robust. The fact that it was the best time dependent periodic controller found imply that finding such a controller is not an easy task. Many hours were used trying to find to find a good controller. Different techniques were tried including different initial values for the bézier curve, different number of control points for the curves, different cost functions varying from none to ones where the norm of the transition matrix at the end was measured and other ones where all the norms at each point in time were weighted dif-

ferently and more. As also mentioned in the controller design section the state transition of the closed loop system was measured in the making of the controller. Observations made showed that no controller managed to bring the last three eigenvalues of $(\boldsymbol{A}(t) - \boldsymbol{B}(t)\boldsymbol{K}(t))$ under 1 (nor over). This either indicates that the three last states are uncontrollable, even though the controllability gramian in Sec. 4.3.5 told otherwise, or maybe they are very hard to control. Initially, with the two nonholonomic constraints it would be expected an uncontrollable subspace of dimension 2. The gramian showed something else but experiments sort of confirmed parts of the initial guess. Either way the task of designing a controller stabilizing the system seems to be a master thesis on its own.

Chapter 8

Conclusion and Further Work

8.1 Conclusion

The analysis of a four DOF and a somewhat smaller analysis of a five DOF nonholonomic system was done in Ch. 4. First the equations of motions were found for the nonholonomic system which was done in a similar way as the traditional Euler-Lagrange method, except for some extra constraints that showed up due to the nonholonomy. After deriving the EOM, virtual holonomic constraints were used to parametrize a desired motion for the system. This could then be used to show the relation between the monotonically increasing path coordinate, θ, and its derivative, $\dot{\theta}$ with the help of $\alpha(\theta), \beta(\theta)$ and $\gamma(\theta)$, which are second order differential equation parameters for θ. A method for calculating the transverse linearization was shown for the nonholonomic systems. It was done by calculating a set of transverse coordinates and linearizing them. This could then be used to design a controller to stabilize a desired orbital motion for the system.
It also was shown that the difference in the dynamics of the four DOF and five DOF system is quite extensive because the potential energy not is constant. The four DOF system showed, both with the circular path and the straight line segment, no restrictions on the parametrization of the trajectories, $(\theta, \dot{\theta})$ and $(\phi, \dot{\phi})$. For the five DOF system, on the other hand, it was showed that for a simple circular orbit a constant speed as a function of the tilt was necessary to maintain the initial tilt angle. It was also shown two different approaches to calculate the EOM for the five DOF system, where the first one shortened out some of the dynamics of the system due to simplifications by incorporating nonholonomic constraints in the Lagrangian. The

other one described the full dynamics and was used further on.
Simulations of the four DOF system in Ch. 5.2 yielded instability even with no initial errors due to simulation inaccuracies and maybe too big step size in the solver. An attempt to design a control law for the system was done, but with no good results. A constant gain controller was tried out making the system less instable, but still not stable. A periodic and smooth controller was also tried, and gave some promising results. These results were then shown to be incidental as a trial with no initial errors resulted in an unstable system. As much time and effort has been put into the task of finding such a controller, it proves that controller deign for LTV systems is quite extensive and not straight forward.

8.2 Further Work

Even though some controller strategies for the four DOF system were tried out, no good ones were found. This report lays the foundation for the task to design such a controller, both for the four DOF and the five DOF nonholonomic system. The first step would naturally be to find a controller for the circular orbit for a four DOF system and then a controller for the straight line segments afterwards (which probably is easier). When a controller is found it would be interesting to combine the system with the path planner found in [3], and in that way found a base for an autonomous vehicle with nonholonomic constraints present, navigating in an obstacle crowded workspace. Combined with some sensors (camera based, ultra sound etc.) an algorithm detecting obstacles and free paths could be implemented making the vehicle fully automatic and autonomous. Such vehicles could be used in a wide range of applications for example in hazardous environments, remotely operated places and more.
An even more extensive work would be to find a controller for the 5 DOF nonholonomic system and maybe use it to control a unicycle or something similar. Some kind of an Euler's Disk could be made with a built in motor and by varying the angular speed of the disc, an endless spin mechanism could be made. That is, initiate the spin at some tilt angle, and the built in controller regulates the speed such that it spins "forever".

Bibliography

[1] M.W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley & Sons, 2006.

[2] Elephant.co.uk. A car insurance agency. `http://www.elephant.co.uk/pressReleases/05042012/Parallel/parking/proves/problematic` Checked May 2011.

[3] Stian Tvetmarken. Motion planning of unicycles and car-like vehicles. Project report, December 2011.

[4] LeonidB. Freidovich and AntonS. Shiriaev. Transverse linearization for underactuated nonholonomic mechanical systems with application to orbital stabilization. In Rolf Johansson and Anders Rantzer, editors, *Distributed Decision Making and Control*, volume 417 of *Lecture Notes in Control and Information Sciences*, pages 245–258. Springer London, 2012.

[5] R. Siegwart and I.R. Nourbakhsh. *Introduction to autonomous mobile robots*. Intelligent robotics and autonomous agents. MIT Press, 2004.

[6] R.M. Murray, Z. Li, and S.S. Sastry. *A mathematical introduction to robotic manipulation*. CRC Press, 1994.

[7] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Available at http://planning.cs.uiuc.edu/.

[8] J.P. Laumond. *Robot motion planning and control*. Lecture notes in control and information sciences. Springer, 1998.

[9] Steven M. Lavalle. Rapidly-exploring random trees: A new tool for path planning. 1998.

[10] A.S. Shiriaev, L.B. Freidovich, and S.V. Gusev. Transverse linearization for controlled mechanical systems with several passive degrees of free-

dom. *Automatic Control, IEEE Transactions on*, 55(4):893 –906, april 2010.

[11] Uwe Mettin. *Principles for Planning and Analyzing Motions of Underactuated Mechanical Systems and Redundant Manipulators*. PhD thesis, Umeå University, 2009.

[12] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E.R. Westervelt, C. Canudas-De-Wit, and J.W. Grizzle. Rabbit: a testbed for advanced control theory. *Control Systems, IEEE*, 23(5):57–79, oct. 2003.

[13] C.T. Chen. *Linear System Theory and Design*. Oxford series in electrical and computer engineering. Oxford University Press, 2009.

[14] `http://en.wikipedia.org/wiki/Controllability_Gramian` Checked May 2011.

[15] A. Bloch. *Nonholonomic mechanics and control*. Interdisciplinary applied mathematics: Systems and control. Springer, 2003.

[16] Stian H. Askeland. Stabilization of brachiation locomotion in a monkey robot. Master's thesis, Norwegian University of Science and Technology, 2012.

[17] Eric W. Weisstein. Bézier curves. From MathWorld - A Wolfram Web Resource. `http://mathworld.wolfram.com/BezierCurve.html` Checked May 2011.

[18] D. Salomon. *Curves And Surfaces for Computer Graphics*. Springer Science+Business Media, 2006.

[19] Eric W. Weisstein. Taylor series. From MathWorld - A Wolfram Web Resource. `http://mathworld.wolfram.com/TaylorSeries.html` Checked May 2011.