Master's thesis

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

Gustav Omberg Often

Planning and Performing Dynamic Manipulations: A Case Study of the Butterfly Robot

Master's thesis in Engineering Cybernetics
Supervisor: Anton Shiriaev
May 2021

NTNU
Norwegian University of
Science and Technology

Gustav Omberg Often

Planning and Performing Dynamic Manipulations: A Case Study of the Butterfly Robot

Master's thesis in Engineering Cybernetics
Supervisor: Anton Shiriaev
May 2021

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

NTNU
Norwegian University of
Science and Technology

Abstract

Two dynamic models for the Butterfly Robot are derived, one with four degrees of freedom and one with two. The first is used to derive constraint forces for the robot, while the second is used to derive a stabilizing controller for the robot. To derive the controller the virtual holonomic constraint approach is used. First to derive a feedforward, and then a partial feedback linearization for a set of transverse coordinates. The linearization of the transverse coordinates are then stabilized with the help of a time varying linear quadratic regulator. An observer for the set of linear transverse coordinates is then found. The controller and observer is then tested on a physical realization of the Butterfly Robot at NTNU. It is found that the controller is able to achieve the goal of continuous rolling in one direction.

Sammendrag

To matematiske modeller for sommerfugl-roboten blir funnet, én med fire frihetsgrader og én med to frihetsgrader. Modellen med fire frihetsgrader blir brukt til å finne kreftene som opprettholder begrensningene i systemet mens den med to frihetsgrader blir brukt til å finne en stabiliserende regulator for systemet. For å utlede regulatoren blir virtuelle holonomiske begrensninger brukt, først for å finne en foroverkobling når systemet er på ønsket bane, deretter for å finne en delevis lineariserende tilbakekobling for et sett med transverse koordinater. De transverse koordinatene blir deretter stabilisert rundt nullpunktet av en lineær kvadratisk regulator. En observator er for de transverse koordinatene blir deretter formulert. Regulatoren og observatoren blir til slutt testet på et fysisk oppsett av sommerfugl-roboten på NTNU. Eksperimentene viser at denne regulator klarer å oppnå målet om kontinuerlig rulling i en retning.

Prephase

This master thesis has been written at the Department of Engineering Cybernetics at the Norwegian University of Science and Technology during the spring of 2021. It is a continuation of the study done by the author on the Butterfly Robot in the specialization project during the autumn of 2020.

I would like to thank my supervisor Anton Shiriaev for guidance and valuable feedback during the course of writing this thesis. I would also like to thank my fellow students at NTNU for valuable discussions during the semester.

Trondheim, May 2021
Gustav Omberg Often

List of Symbols

$\alpha(\varphi)$	The part of the reduced dynamics which is multiplied with $\ddot{\varphi}$.
$\beta(\varphi)$	Part of the reduced dynamics which is multiplied with $\dot{\varphi}^2$.
$\dot{\varphi}_\star(\varphi)$	Nominal time derivative of φ.
$\gamma(\varphi)$	Part of the reduced dynamics.
$\kappa(\varphi)$	Curvature of the path of the center of the sphere.
$\kappa_f(\phi)$	Curvature of the frame.
$\boldsymbol{C}(\vec{q}, \dot{\vec{q}})$	Coriolis and centrifugal matrix.
$\boldsymbol{G}(\vec{q})$	Gravity matrix.
$\boldsymbol{L}$	Gain matrix for the Luenberger observer.
$\boldsymbol{M}(\vec{q})$	Mass-inertia matrix.
$\boldsymbol{Q}$	Matrix of generalised constraint forces.
$\boldsymbol{R}$	Rotational matrix about the z-axis.
$\vec{\delta}(\phi)$	Vector from the origin in the body frame to the point closest to the sphere on the edge of the frame.
$\vec{\lambda}$	Vector of the Lagrangian multipliers.
$\vec{\rho}(\varphi, w)$	Vector from origin in body frame to the center of the sphere.
$\vec{\tau}(\varphi)$	Tangent vector to the path travelled by the center of the sphere.
$\vec{\tau}_f(\phi)$	Tangent vector to the frame.
$\vec{\xi}$	Alternative set of coordinates to describe the Butterfly Robot
$\vec{k}$	Unit vector along the y-axis in inertial frame.
$\vec{n}_+(\varphi)$	Normal vector to the frame pointing away from the frame.
$\vec{q}$	Vector of all the states in the system, used for both the system with four states and the system with two states.
$\vec{v}_d(\vec{q}, \dot{\vec{q}})$	Sphere/discs linear velocity in the inertial frame.
$\vec{x}^\perp$	Vector of the transverse coordinates.
$\vec{y}$	Transverse coordinates calculated from measurements of the states.
ϕ	Angle between the y-axis in the body frame and the vector $\vec{\delta}$.
ψ	Angle the sphere has rotated.
$\Theta(\varphi)$	Virtual holonomic constraint to create a link between φ and ϑ.

φ	Angle between the y-axis in the body frame and the vector $\vec{\rho}$.
ϑ	Angle the frame has rotated in the inertial frame, also angle between inertial frame and body frame.
a	Constant used to describe the frame.
a_i	Constants used to describe the frame with a cosine series.
b	Constant used to describe the frame.
c_i	Constants used to describe the VHC, $i \in \{1..5\}$.
$f_1(\vec{q})$	Contact constraint for the sphere.
$f_2(w)$	No-slip constraint
g	Gravitational acceleration.
J_f	Inertia of the frame about the z-axis.
J_s	Inertia of the sphere.
k	Matrix function to simplify feedforward term
$K(\vec{q}, \dot{\vec{q}})$	Kinetic energy of the robot.
m_d	Mass of the sphere/disc.
$P(\vec{q})$	Potential energy of the Butterfly Robot.
Q	Tuning matrix for the periodic LQR.
R	Gain matrix for the periodic LQR.
R_d	Effective rolling radius of the sphere.
$s(\varphi)$	Distance the center of the sphere has travelled.
$s_f(\phi)$	Distance the sphere has travelled along the frame.
u	Input to the system
$u_\star$	Nominal input to the system
w	shortest distance between the edge of the frame and the surface of the sphere.
$w(\varphi)$	Virtual input to the system of transverse coordinates.
X	Solution of the periodic Riccati differential equation.
x_s	Spheres x-position in the inertial frame.
y_1	First transverse coordinate related to ϑ.
y_2	Second transverse coordinate related to $\dot{\vartheta}$.
y_s	Spheres y-position in the inertial frame.
z	Third transverse coordinate related to $\dot{\varphi}$.
$^\perp$	Estimate of the transverse coordinates.
ω_s	Angular velocity of the sphere given in the inertial frame.

Notation

In this text a dot above a variable implies derivative with respect to time, i.e.

$$\dot{x} = \frac{dx}{dt} \tag{1}$$

An apostrophe after a variable implies derivative with respect to the function's variable. In this report this will always be with respect to φ or ϑ, the angle between the y-axis in body frame and the vector to the center of the sphere, i.e.

$$x' = \frac{dx}{d\varphi} \tag{2}$$

Vectors are written with an arrow over them, i.e. $\vec{v}$. After a function is defined as dependent on another variable the parenthesis showing this is often dropped. This is to make the derivations more readable. i.e.

$$\alpha(\varphi) \text{ becomes } \alpha \tag{3}$$

In this text the cross product sign, $\times$, implies the magnitude of the resulting vector from the cross product. This will be equal to the size of the third element of the resulting vector as all vectors are in the xy-plane. This is done to save space in formulas and make them more readable.

$$\vec{v} \times \vec{u} := \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \vec{w} \tag{4}$$

where $\vec{w}$ would be the resulting vector if the cross product returned a vector.

Contents

Chapter 1

Introduction

In this work the Butterfly Robot will be considered. This is a benchmark example for planning and performing nonprehensile manipulation proposed in [7]. It was also extensively investigated in [14], and much of this text is based on their work. The goal is to be able to manipulate a sphere with a hand-shaped robot arm. See an illustration in Figure 1.1. The inspiration is from a juggling trick called the butterfly where the juggler is able to make a sphere roll from the palm, over the fingers and onto the back of his or her hand.

1.1 Motivation

Non-prehensile manipulation is something humans do regularly to solve tasks, this includes everything from walking, throwing objects, pushing objects, balancing objects, carrying containers of liquids and much more. Every task where the person or robot can not grasp the object he, she or it is manipulating can be defined as non-prehensile manipulation. Here lies the problem for performing non-prehensile manipulation for robotic systems; when the object can not be grasped firmly it cannot be made part of the robot, which inherently makes any system that is performing non-prehensile manipulation underactuated. Underactuated systems are notoriously difficult to control, especially if the degree of r is larger than one, that is more than one of the states does not have a dedicated actuator. [7]

Development of better models and controllers for the Butterfly Robot is therefore of interest as it enables better understanding of such systems. This can lead to the creation of robotic systems better able to perform tasks previously done by humans. This can be of interest as such robots can work in environments which are hazardous to humans or perform mundane repetitive work. This is the motivation for the study of the Butterfly Robot.

1.2 Scope for this thesis

The goal for this thesis is to be able to achieve continuous rolling in one direction for a physical realization of the Butterfly Robot at NTNU. A model with only one degree of underactuation will be considered for controlling the Butterfly Robot.

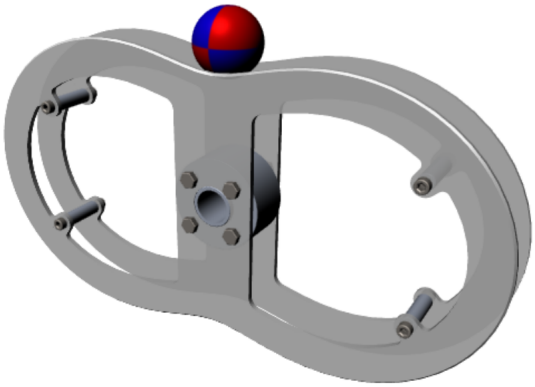

Figure 1.1: Illustration of the Butterfly, created by Maksim Surov for [14].

1.3 Outline of the text

This text is structured as follows: First some background theory is given. The kinematics for the Butterfly Robot is then derived. The kinematics is then used to derive two dynamical models for the system.
The simpler of these two models is then used to find feasible trajectories for the Butterfly Robot with the help of a virtual holonomic constraint. This constraint is then used to further reduce the model to a second degree differential equation, the $\alpha\beta\gamma - equation$.
The found trajectories are then stabilized with the help of a time varying LQR of a set of transverse coordinates. An observer is used to estimate these transverse coordinates.
Results from experiments on a physical realization of the Butterfly Robot are then given. Lastly these results are discussed.

1.4 Background theory

1.4.1 Dynamical model from Christoffel symbols

Given the Lagriangian of a system, $\mathcal{L} = K - P$, and general coordinates $q_i \; i \in \{1, .., n\}$ the equations of motions of a mechanical system can be written as given in Equation 1.1 by using Christoffel symbols [13]. K is here the kinetic energy of the system while P is the potential energy.

$$\boldsymbol{D}(q)\ddot{q} + \boldsymbol{C}(q, \dot{q})\dot{q} + \vec{g}(q) = \vec{\tau} \tag{1.1}$$

The kinetic energy of a physical system is always quadratic in velocities and can be written as:

$$K(\vec{q}, \dot{\vec{q}}) = \frac{1}{2}\dot{\vec{q}}^{\top}\boldsymbol{D}(\vec{q})\dot{\vec{q}} \tag{1.2}$$

The elements of $\boldsymbol{C}(\vec{q}, \dot{\vec{q}})$ can then be found as:

$$c_{kj} = \sum_{i=1}^{n} c_{ijk}(q)\dot{\vec{q}}, \;\; i, j, k \in \{1, .., n\} \tag{1.3}$$

where c_{ijk} are known as Christoffel symbols. Expressions for these are given in Equation 1.4.

$$c_{ijk}(q) = \frac{1}{2}(\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k}), \;\; i, j, k \in \{1, .., n\} \tag{1.4}$$

$\vec{g}(q)$ can be found as:

$$\vec{g}_k = \frac{\partial P}{\partial q_k}, \;\; k \in \{1, .., n\} \tag{1.5}$$

Chapter 2

Model of the Butterfly Robot

2.1 Kinematics of the Butterfly Robot

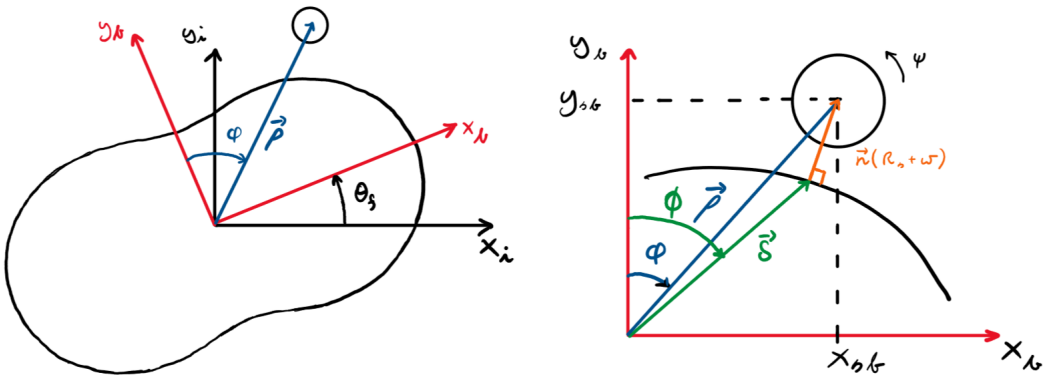

Figure 2.1: Illustration of the Butterfly robot with states

In Figure 2.1 two drawings of the Butterfly Robot are shown. The one to the left shows a closeup of the drawing to the right. These will be used to describe the various states of the system and their relations. Here the body attached frame is drawn in red while the inertial frame is drawn in black. The z-axis in the system is pointing out of the paper according to the right hand rule. The following assumptions will be used to reduce the number of degrees of freedom from twelve to four.

- The frame will not have any linear velocity, and its center is in the center of the inertial frame.
- The frame will only have rotational velocity about the z-axis.
- The sphere will not have any linear velocity along the z-axis.
- The sphere will only have rotational velocity about the z-axis.

This gives that the system is fully described by the states ϑ, the rotation of the frame about the z-axis. x_s and y_s, the spheres position in the inertial frame and ψ, the spheres rotation about the z-axis. This also reduces the problem from a sphere rolling on two plates with some distance between them to a disc rolling on a single plate. The radius of the disc will be given as the square root of the sphere's radius squared minus half the distance between the plates squared, $R_d = \sqrt{R_s^2 - R_f^2}$. See appendix for derivation.

Introducing the vector $\vec{\rho}(\varphi, w)$ shown in the right figure in Figure 2.1 gives the following equation for the discs position, where $\boldsymbol{R}$ is the rotation matrix about the z-axis.

$$\begin{bmatrix} x_s \\ y_s \end{bmatrix} = \boldsymbol{R}(\vartheta)\vec{\rho}(\varphi, w) \tag{2.1}$$

In this equation the variable w has been introduced to describe the shortest distance from the frame to the edge of the disc. When the disc is in contact with the frame this variable will be equal to zero. From Figure 2.1 it can be observed that the vector $\vec{\rho}(\varphi, w)$ is given as:

$$\vec{\rho} = \vec{\delta}(\phi) + \vec{n}_+(\phi)(w + R_d) \tag{2.2}$$

where R_d is the radius of the disc, δ is given by the shape of the frame and $\vec{n}_+$ is the normal vector to the frame for a given ϕ that is pointing away from the frame. Here it is worth noting that the angel φ is only equal to ϕ when the normal vector to the frame is pointing in the same direction as $\vec{\delta}$.

The set of states $\begin{bmatrix} \vartheta & \phi & w & \psi \end{bmatrix}$ is more convenient for describing the Butterfly Robot than the states $\begin{bmatrix} \vartheta & \psi & x_s & y_s \end{bmatrix}$ and will therefore be used for further derivations. It is worth noting that φ, w and ψ is very hard to measure directly in the actual robot. Measurements of x_s and y_s will therefore be used to calculate these variables.

2.1.1 Frenet-Serret frames for the Butterfly Robot

To describe the sphere's path it is convenient to introduce two Frenet-Serret frames. One with origin in the center of the sphere and another with the origin in the point of contact between the sphere and the frame. The variables $s(\varphi)$ and $s_f(\phi)$ shown in Figure 2.2 will be used to describe the frames position. Equations for these variables are given in Equation 2.3.

$$\begin{aligned} s(\varphi) &= \int_{\varphi_0}^{\varphi} \left\| \frac{\partial \vec{\rho}}{\partial \varphi} \right\| d\varphi \\ s_f(\phi) &= \int_{\varphi_0}^{\phi} \left\| \frac{\partial \vec{\delta}}{\partial \varphi} \right\| d\phi \end{aligned} \tag{2.3}$$

The basis vectors of a Frenet-Serret is the tangent vector to the path travelled, the normal vector to the path and the binormal vector to the path. These are defined as follows [1].

$$\vec{\tau} = \frac{\partial \vec{\rho}}{\partial s}, \quad \vec{\tau}_f = \frac{\partial \vec{\delta}}{\partial s_f} \tag{2.4}$$

$$\vec{n} = \frac{\frac{\partial \vec{\tau}}{\partial s}}{\left\| \frac{\partial \vec{\tau}}{\partial s} \right\|}, \quad \vec{n}_f = \frac{\frac{\partial \vec{\delta}}{\partial s}}{\left\| \frac{\partial \vec{\delta}}{\partial s_f} \right\|} \tag{2.5}$$

$$\vec{b} = \vec{\tau} \times \vec{n}, \quad \vec{b} = \vec{\tau}_f \times \vec{n}_f \tag{2.6}$$

From Figure 2.2 it can be observed that $\vec{\tau}$ and $\vec{\tau}_f$ will be equal and $\vec{n}$ and $\vec{n}_f$ will be equal. Note that the curvature of the two paths κ and κ_f will not be equal. The torsion of both curves will be zero as these are planar curves [1]. When the torsion is zero the Frenet-Serret formulas [1] reduces to a simpler form given as:

$$\frac{d\vec{\tau}}{ds} = \kappa \vec{n} \tag{2.7}$$

$$\frac{d\vec{n}}{ds} = -\kappa \vec{\tau} \tag{2.8}$$

2.1.2 Contact constraint

If the disc is assumed to be in contact with the frame the variable w will be equal to zero. This gives the first constraint for the Butterfly robot:

$$f_1(\vec{q}) = w = 0 \tag{2.9}$$

2.1.3 No-slip constraint

Under the assumption that the disc does not slip a relation between ψ and φ can be derived. When the disc does not slip the relation between s_f and ψ are given by Equation 2.10. The minus in the formula comes from the way that the variables are defined in Figure 2.2 which is the common way to define s in a Frenet-Serret frame.

$$s_f = -R_d\psi, \quad \dot{s}_f = -R_d\dot{\psi} \tag{2.10}$$

The variable s_f is given in terms of ϕ. An expression relating s_f and s is therefore needed. Using the fact that the tangent vector and the normal vector for the two Frenet–Serret frames are the same and Equation 2.8 the following computations can be done:

$$\frac{\partial\vec{\rho}}{\partial s_f} = \frac{\partial\vec{\delta}}{\partial s_f} + \frac{\partial\vec{n}}{\partial s_f}R_d = \vec{\tau}_f - \kappa_f\vec{\tau}_f R_d = \vec{\tau}_f(1-\kappa_f R_d) \tag{2.11}$$

Multiplying both sides by ∂s_f and lining up with the expression for the derivative of $\vec{\rho}$ with respect to s gives the following:

$$\partial\vec{\rho} = \vec{\tau}_f(1-\kappa_f R_d)\partial s_f \tag{2.12}$$

$$\partial\vec{\rho} = \vec{\tau}\partial s \tag{2.13}$$

$$\tag{2.14}$$

Since the two tangent vectors, τ and τ_f, are equal the following relation between ∂s_f and ∂s can be observed:

$$\partial s_f = \frac{1}{(1-\kappa_f R_d)}\partial s \tag{2.15}$$

Integrating both sides gives ($s(0)$ and $s_f(0)$ has been set to zero):

$$s_f = \int_0^s \frac{1}{(1-\kappa_f(s)R_d)}ds \tag{2.16}$$

As a result the no slip constraint can be written as an integral equation in terms of s:

$$f_2(\vec{q}) = \int_0^s \frac{1}{(1-\kappa_f(s)R_d)}ds + R_d\psi = 0 \tag{2.17}$$

By the fundamental theorem of calculus the time derivative is given as:

$$\frac{1}{(1-\kappa_f(s)R_d)}\dot{s} + R_d\dot{\psi} = 0 \tag{2.18}$$

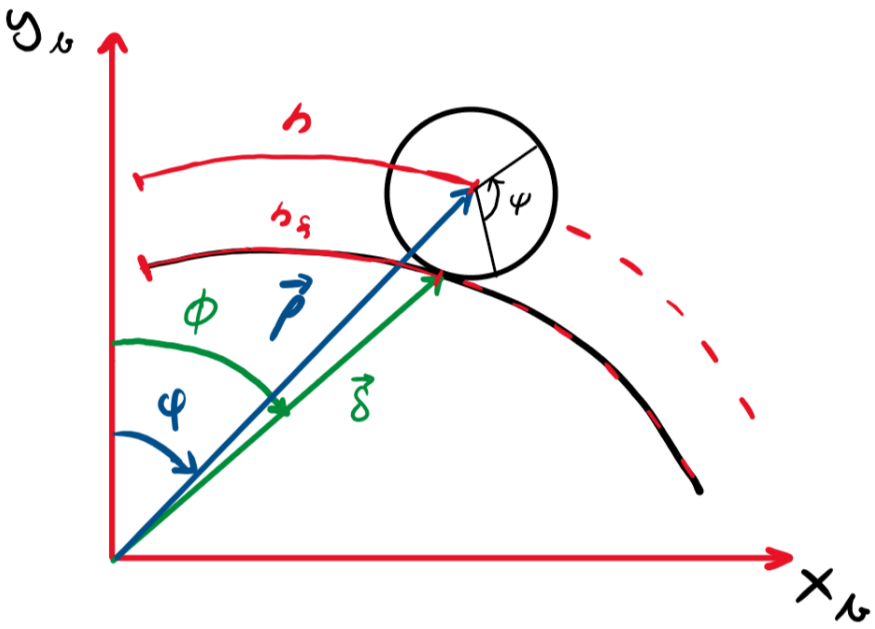

Figure 2.2: Illustration Butterfly Robot with no slip

2.2 Dynamics of the Butterfly Robot

To derive the dynamics of the Butterfly Robot Lagrange's Equations of motion will be used. For this the kinetic energy and the potential energy of the robot will be needed. Two models will be derived, one consisting of the four states ϑ, φ, ψ and w, and a reduced model consisting only of the states ϑ and φ. The first model will be used to derive the constraint forces needed to uphold the two constraints derived earlier while the second model will be used to derive a controller for the system.

2.2.1 4-DoF system

Assuming the center of mass for the frame is in the origin of the inertial frame the only potential energy in the system results from the disc's position. This gives that the potential energy is given by Equation 2.19 where the subscript 2 denotes the second element of the vector. This equation can be observed from Figure 2.1. Her m_d is the mass of the sphere/disc.

$$P(\vec{q}) = [gm_d \boldsymbol{R}(\vartheta)\vec{\rho}(\varphi, w)]_2 \tag{2.19}$$

As the frame will have no linear velocity the kinetic energy of the system will be given by the frame's angular velocity, the sphere's angular velocity and the sphere's linear velocity in the inertial frame. The angular velocity of the frame is given by the time derivative of ϑ. Since the z-axis of the inertial frame and the body frame are parallel the angular velocity of the sphere in the inertial frame can be found as the sum of $\dot{\vartheta}$ and $\dot{\psi}$. This gives:

$$\omega_s = \dot{\vartheta} + \dot{\psi} \tag{2.20}$$

The discs linear velocity will be given by the time derivative of its position in the inertial frame. The spheres position is given as:

$$\vec{r}_d(\vec{q}) = \boldsymbol{R}(\vartheta)\vec{\rho}(\varphi, w) \tag{2.21}$$

Taking the time derivative of Equation 2.21 gives by Leibniz rule:

$$\vec{v}_d(\vec{q}, \dot{\vec{q}}) = \boldsymbol{R}'(\vartheta)\dot{\vartheta}\vec{\rho}(\varphi, w) + \boldsymbol{R}(\vartheta)\frac{\partial \vec{\rho}(\varphi, w)}{\partial \varphi}\dot{\varphi} + \boldsymbol{R}(\vartheta)\vec{n}_+(\varphi)\dot{w} \tag{2.22}$$

The kinetic energy of the system can then be written as

$$K(q,\dot{q}) = \frac{1}{2}(J_f\dot{\vartheta}^2 + J_s(\dot{\vartheta}+\dot{\psi})^2 + m_d\vec{v}_d(q,\dot{q})^\top\vec{v}_d(q,\dot{q}) \tag{2.23}$$

Using Equation 2.22 the kinetic energy can be written on matrix form as given by Equation 2.24. Here it has been used that the product between a rotation matrix and its transpose is the identity matrix. The notation for dependencies of the functions has been dropped to make the expressions more readable.

$$K(\vec{q},\dot{\vec{q}}) = \frac{1}{2}\dot{\vec{q}}\begin{bmatrix} J_f + J_s + m_d\vec{\rho}^\top\vec{\rho} & m_d(\boldsymbol{R}'\vec{\rho})^\top\boldsymbol{R}\frac{\partial\vec{\rho}}{\partial\varphi} & J_s & m_d(\boldsymbol{R}'\vec{\rho})^\top\boldsymbol{R}\vec{n}_+ \\ m_d(\boldsymbol{R}'\vec{\rho})^\top\boldsymbol{R}\frac{\partial\vec{\rho}}{\partial\varphi} & m_d\frac{\partial\vec{\rho}}{\partial\varphi}^\top\frac{\partial\vec{\rho}}{\partial\varphi} & 0 & m_d\frac{\partial\vec{\rho}}{\partial\varphi}^\top\vec{n}_+ \\ J_s & 0 & J_s & 0 \\ m_d(\boldsymbol{R}'\vec{\rho})^\top\boldsymbol{R}\vec{n}_+ & m_d\frac{\partial\vec{\rho}}{\partial\varphi}^\top\vec{n}_+ & 0 & m_d \end{bmatrix}\dot{\vec{q}} \tag{2.24}$$

Using the following identities this matrix can be simplified a bit:

$$\vec{x}^\top\boldsymbol{R}'(\vartheta)^\top\boldsymbol{R}(\vartheta)\vec{z} = \vec{x}\times\vec{z} \tag{2.25}$$

$$\frac{\partial\vec{\rho}(\varphi,w)}{\partial\varphi} = \vec{\tau}(\varphi)s'(\varphi,w) \tag{2.26}$$

$$\vec{\tau}^\top\vec{\tau} = 1 \tag{2.27}$$

$$\vec{\tau}^\top\vec{n}+ = 0 \tag{2.28}$$

$$K(\vec{q},\dot{\vec{q}}) = \frac{1}{2}\dot{\vec{q}}\begin{bmatrix} J_f + J_s + m_d\vec{\rho}^\top\vec{\rho} & m_d s'\vec{\rho}\times\vec{\tau} & J_s & m_d\vec{\rho}\times\vec{n}_+ \\ m_d s'\vec{\rho}\times\vec{\tau} & m_d s'^2 & 0 & 0 \\ J_s & 0 & J_s & 0 \\ m_d\vec{\rho}\times\vec{n}_+ & 0 & 0 & m_d \end{bmatrix}\dot{\vec{q}} \tag{2.29}$$

Using Christoffel symbols the dynamics for the butterfly robot is found to be given by Equation 2.30. The derivation and expressions for $\boldsymbol{M}$, $\boldsymbol{C}$ and $\boldsymbol{G}$ can be found in Equation A.4, Equation A.15 and Equation A.16 respectively in the appendix.

$$\boldsymbol{M}(\vec{q})\ddot{\vec{q}} + \boldsymbol{C}(\vec{q},\dot{\vec{q}})\dot{\vec{q}} + \boldsymbol{G}(\vec{q}) = \boldsymbol{B}u + \boldsymbol{Q}(\vec{q}) \tag{2.30}$$

Since only ϑ is actuated $\boldsymbol{B}$ is given as $\boldsymbol{B} = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^\top$. Using the constraints in Equation 2.9 and Equation 2.17 an expression for the constraint forces $\boldsymbol{Q}(q)$ will be found. $\boldsymbol{Q}(q)$ can be expressed as given in Equation 2.31 where $\vec{f}$ is a vector consisting of the two constraints f_1 and f_2 and $\vec{\lambda}$ is a vector of the two Lagrangian multipliers λ_1 and λ_2.[3]

$$\boldsymbol{Q}(\vec{q}) = \frac{\partial\vec{f}(\vec{q})}{\partial\vec{q}}\vec{\lambda} \tag{2.31}$$

Differentiating the two constraints with respect to the states gives:

$$\frac{\partial\vec{f}(\vec{q})}{\partial\vec{q}} = \begin{bmatrix}0 & 0 \\ 0 & \frac{1}{1-\kappa_f(s)R_d}\frac{\partial s}{\partial\varphi} \\ 0 & R_d \\ 1 & 0\end{bmatrix} = \begin{bmatrix}0 & 0 \\ 0 & s'_f(\varphi) \\ 0 & R_d \\ 1 & 0\end{bmatrix} = \boldsymbol{f}_q \tag{2.32}$$

The values for λ_1 and λ_2 can now be found as:

$$\vec{\lambda} = \left\{\boldsymbol{f}_{qL}^{-1}\left(\boldsymbol{M}(\vec{q})\ddot{\vec{q}} + \boldsymbol{C}(\vec{q},\dot{\vec{q}})\dot{\vec{q}} + \boldsymbol{G}(\vec{q}) - \boldsymbol{B}\tau\right)\right\}\Big|_{\text{constraints and their derivatives upheld}} \tag{2.33}$$

Where $\boldsymbol{f}_{q_L}^{-1}$ is the left inverse of the matrix $\boldsymbol{f}_q$.

By observing that the generalized force acting on w should be a force, that is given in Newton, while the generalized force acting on ψ should be a torque, that is Newton meters, an interpretation of λ_1 and λ_2 can be found. λ_1 can be interpreted as the normal force acting on the sphere while λ_2 can be seen as the friction force between the sphere and the frame.

It can thus be concluded that λ_1 should be bigger than zero for the sphere to stay on the frame and $|\lambda_2|$ should be smaller than some threshold given by the materials of the frame and the sphere for the sphere to not slip.

2.2.2 2-DoF system

Now it will be assumed that the contact constraint and the no slip constraint is upheld. As a result the function for $\vec{\rho}$ becomes simplified a bit:

$$\vec{\rho}(\varphi, 0) = \vec{\delta}(\varphi) + \vec{n}_+(R_d + 0) = \vec{\delta}(\varphi) + \vec{n}_+ R_d = \vec{\rho}(\varphi) \tag{2.34}$$

The no slip constraint gives that the angular velocity of the disc in the inertial frame can be written as:

$$\omega_d = \dot{\vartheta} + \dot{\psi} = \dot{\vartheta} - \frac{1}{R_d - \kappa_f(s) R_d^2}\dot{s} = \dot{\vartheta} - \frac{s_f'}{R_d}\dot{\varphi} \tag{2.35}$$

This gives that the kinetic energy of the system with the constraints active is given by:

$$K(\vec{q}, \dot{\vec{q}}) = \frac{1}{2}\dot{\vec{q}}^\top \begin{bmatrix} J_f + J_s + m_d \vec{\rho}^\top \vec{\rho} & m_d s' \vec{\rho}^\top \vec{\tau} - s_f' \frac{J_s}{R_d} \\ m_d s' \vec{\rho}^\top \vec{\tau} - s_f' \frac{J_s}{R_d} & s_f'^2 \frac{J_s}{R_d^2} \end{bmatrix} \dot{\vec{q}} \tag{2.36}$$

Again using Christoffel symbols the model for the dynamics with two degrees of freedom is found to be:

$$\boldsymbol{M}(\vec{q})\ddot{\vec{q}} + \boldsymbol{C}(\vec{q}, \dot{\vec{q}})\dot{\vec{q}} + \boldsymbol{G}(\vec{q}) = \boldsymbol{B}u \tag{2.37}$$

with the matrices:

$$\boldsymbol{M}(q) = m_d \begin{bmatrix} \frac{J_f}{m_d} + \frac{J_s}{m_d} + \vec{\rho}^\top \vec{\rho} & s' \vec{\rho}^\top \vec{\tau} - s_f' \frac{J_s}{m_d R_d} \\ s' \vec{\rho}^\top \vec{\tau} - s_f' \frac{J_s}{m_d R_d} & s_f'^2 \frac{J_s}{m_d R_d^2} \end{bmatrix} \tag{2.38}$$

$$\begin{aligned} C_{11} &= m_d \vec{\rho} \cdot \vec{\tau} s' \dot{\varphi} \\ C_{12} &= m_d (\vec{\rho} \cdot \vec{\tau} s' \dot{\vartheta} + (s'' \vec{\rho} \times \vec{\tau} - s_f'' \frac{J_s}{R_d m_d} + s'^2 \vec{\rho} \times \vec{\kappa}) \dot{\varphi}) \\ C_{21} &= -m_d \vec{\rho} \cdot \vec{\tau} s' \dot{\vartheta} \\ C_{22} &= m_d (s' s'' + s_f' s_f'' \frac{J_s}{R_d^2 m_d}) \dot{\varphi} \end{aligned} \tag{2.39}$$

$$\boldsymbol{G}(q) = g m_d \begin{bmatrix} [\boldsymbol{R}'(\vartheta)\vec{\rho}]_2 \\ [\boldsymbol{R}(\vartheta)\vec{\tau} s']_2 \end{bmatrix} \tag{2.40}$$

$$\boldsymbol{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{2.41}$$

Derivation is given in the appendix.

2.2.3 Parameters of the dynamical models

To be able to use the dynamical models previously derived expressions for the parameters $\vec{\rho}$, $\vec{\tau}$, $\vec{\kappa}$, $\vec{n}_+$, s', s'', s_f' and s_f'' needs to be derived.
From Figure 2.2 it can be observed that $\vec{\rho}$ can be expressed in two ways. One dependent on φ and one dependent on ϕ. The equation dependent on ϕ can be expressed as the sum of the vector from origin to the point of contact and the vector from the point of contact to the center of the sphere. For this expression the normal vector to the frame in the point of contact is needed. The normal vector defined in the Frenet-Serret frame can not be used as it will change sign depending on the curvature of the frame. Instead the normalized tangent vector to the frame rotated anticlockwise by 90 degrees will be used. This gives the following expressions.

$$\vec{n}_+ = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \vec{\tau}_f(\phi) \tag{2.42}$$

$$\vec{\rho}(\varphi) = \vec{\rho}(\phi) = \vec{\delta}(\phi) + R_b \vec{n}(\phi) \tag{2.43}$$

$$\vec{\tau}(\varphi) = \vec{\tau}_f(\phi) = \frac{\partial \vec{\delta}}{\partial \phi} \Big/ \left\| \frac{\partial \vec{\delta}}{\partial \phi} \right\| \tag{2.44}$$

$$\vec{\kappa}(\varphi) = \frac{\partial^2 \vec{\rho}}{\partial s^2} = \frac{\partial^2 \vec{\rho}}{\partial \varphi^2}\frac{\partial \varphi^2}{\partial s^2} = \frac{\partial^2 \vec{\rho}}{\partial \phi^2}\frac{\partial \phi^2}{\partial \varphi^2}\frac{\partial \varphi^2}{\partial s^2} \tag{2.45}$$

The expression for s' is derived as follows:

$$s'(\varphi) = \frac{\partial}{\partial \varphi}\int_{\varphi_0}^{\varphi} \left\| \frac{\partial \vec{\rho}}{\partial \varphi} \right\| d\varphi = \left\| \frac{\partial \vec{\rho}}{\partial \varphi} \right\| = \left\| \frac{\partial \vec{\rho}}{\partial \phi}\frac{\partial \phi}{\partial \varphi} \right\| = \left\| \frac{\partial \vec{\delta}(\phi) + R_b \vec{n}(\phi)}{\partial \phi} \right\| \left| \frac{\partial \phi}{\partial \varphi} \right| \tag{2.46}$$

Assuming that the derivative of ϕ with respect to φ is never equal to zero this can be written as:

$$s'(\varphi) = \left\| \frac{\partial \vec{\delta}(\phi) + R_b \vec{n}(\phi)}{\partial \phi} \right\| \left| \frac{\partial \varphi}{\partial \phi} \right|^{-1} \tag{2.47}$$

To find $s''(\varphi)$ a similar approach will be used:

$$s''(\varphi) = \frac{\partial s'(\varphi)}{\partial \varphi} = \frac{\partial s'(\varphi)}{\partial \phi}\frac{\partial \phi}{\partial \varphi} = \frac{\partial \left(\left\| \frac{\partial \vec{\rho}}{\partial \phi} \right\| \left| \frac{\partial \varphi}{\partial \phi} \right|^{-1} \right)}{\partial \phi}\frac{\partial \phi}{\partial \varphi} \tag{2.48}$$

$$= \frac{\frac{\partial \vec{\rho}}{\partial \phi}^{\top}\left(\frac{\partial^2 \vec{\rho}}{\partial \phi^2}\right)}{\left\| \frac{\partial \vec{\rho}}{\partial \phi} \right\|} \left| \frac{\partial \varphi}{\partial \phi} \right|^{-1} \frac{\partial \phi}{\partial \varphi} - \left\| \frac{\partial \vec{\rho}}{\partial \phi} \right\| \frac{\operatorname{sign}\left(\frac{\partial \varphi}{\partial \phi}\right)\frac{\partial^2 \varphi}{\partial \phi^2}}{\left| \frac{\partial \varphi}{\partial \phi} \right|^2}\frac{\partial \phi}{\partial \varphi} \tag{2.49}$$

All of these expressions depend on φ and ϕ. Therefore expressions $f(\varphi) = \phi$ and $g(\phi) = \varphi$ will be needed. From figure 2.2 it can be observed that $\varphi = \operatorname{atan2}(\rho_x, \rho_y)$, here subscript x denotes the x-component while subscript y denotes the y-component. This gives that the function $g(\phi) = \varphi$ can be written as:

$$g(\phi) = \operatorname{atan2}(\delta_x(\phi) - R_b \tau_{fy}(\phi), \delta_y(\phi) + R_b \tau_{fx}(\phi)) \tag{2.50}$$

$f(\varphi)$ can be found numerically as the inverse of $g(\phi)$. In this report this is done in Matlab through a `spline` function. Differentiating $g(\phi)$ we will find that $g'(\phi) > 0 \ \forall \ \phi$. This gives that $\vec{\kappa}$, s' and s'' can be simplified to:

$$\vec{\kappa} = \frac{\partial^2 \vec{\rho}}{\partial \phi^2}\frac{1}{g'(\phi)^2 s'(\phi)^2} \tag{2.51}$$

$$s' = \left\| \frac{\partial \vec{\delta}(\phi) + R_b \vec{n}(\phi)}{\partial \phi} \right\| \frac{1}{g'(\phi)} \tag{2.52}$$

$$s'' = \frac{\frac{\partial \vec{\rho}}{\partial \phi}^{\top}\frac{\partial^2 \vec{\rho}}{\partial \phi^2}}{\left\| \frac{\partial \vec{\rho}}{\partial \phi} \right\|}\frac{1}{g'(\phi)^2} - \left\| \frac{\partial \vec{\rho}}{\partial \phi} \right\| \frac{g''(\phi)}{g'(\phi)^3} \tag{2.53}$$

Using the the function $g(\phi)$ s'_f can be written as:

$$s'_f = \left\| \frac{\partial \vec{\delta}}{\partial \phi} \right\| \frac{\partial \phi}{\partial \varphi} = \left\| \frac{\partial \vec{\delta}}{\partial \phi} \right\| \frac{1}{g'(\phi)} \tag{2.54}$$

Taking the partial derivative of this with respect to φ gives:

$$s''_f = \frac{\frac{\partial \vec{\delta}}{\partial \phi}^{\top}\frac{\partial^2 \vec{\delta}}{\partial \phi^2}}{\left\| \frac{\partial \vec{\delta}}{\partial \phi} \right\|}\frac{1}{g'(\phi)^2} - \left\| \frac{\partial \vec{\delta}}{\partial \phi} \right\| \frac{g''(\phi)}{g'(\phi)^3} \tag{2.55}$$

Chapter 3

Motion planning, controller and observer

3.1 Motion planing

The motion that will be considered in this text is continuous rolling of the sphere along the frame. This will be done using an approach with a virtual holonomic constraint and a reduced model of the system known as the $\alpha\beta\gamma$-equation.

3.1.1 Virtual holonmic constraint

If the contact constraint and the no slip constraint is upheld the Butterfly will be an underactuated system of degree one. To be able to control this system a virtual holonomic constraint (VHC) will be introduced. This will create a relationship between the two states ϑ and φ. This constraint will be imposed on the system through feedback. As a result the system can be described as a one degree of freedom system if the VHC is upheld. Firstly a constraint that is active when the system is on the desired trajectory will be considered. This constraint will be given as:

$$\vartheta = \Theta(\varphi) \tag{3.1}$$

where Θ is a function that is twice differentiable and has some desired properties associated with the equilibria of the reduced system. Differentiating the VHC with respect to time gives the following:

$$\dot{\vartheta} = \frac{\partial \Theta(\varphi)}{\partial \varphi}\dot{\varphi} = \Theta'(\varphi)\dot{\varphi} \tag{3.2}$$

$$\ddot{\vartheta} = \frac{\partial^2 \Theta(\varphi)}{\partial \varphi^2}\dot{\varphi}^2 + \frac{\Theta(\varphi)}{\partial \varphi}\ddot{\varphi} = \Theta''(\varphi)\dot{\varphi}^2 + \Theta'(\varphi)\ddot{\varphi} \tag{3.3}$$

3.1.2 Reduced Dynamics, the $\alpha\beta\gamma$-equation

Inserting the VHC and its time derivatives in the 2-DoF system dynamics given in Equation 2.37 results in the following dynamics:

$$\boldsymbol{M}\left(\begin{bmatrix}\Theta\\ \varphi\end{bmatrix}\right)\begin{bmatrix}\Theta''\dot{\varphi}^2 + \Theta'\ddot{\varphi}\\ \ddot{\varphi}\end{bmatrix} + \boldsymbol{C}\left(\begin{bmatrix}\Theta\\ \varphi\end{bmatrix}, \begin{bmatrix}\Theta'\dot{\varphi}\\ \dot{\varphi}\end{bmatrix}\right)\begin{bmatrix}\Theta'\dot{\varphi}\\ \dot{\varphi}\end{bmatrix} + \boldsymbol{G}\left(\begin{bmatrix}\Theta\\ \varphi\end{bmatrix}\right) = \boldsymbol{B}u \tag{3.4}$$

From this it can be observed that the dynamics is only dependent on φ, $\dot{\varphi}$, $\ddot{\varphi}$ and u if the VHC is upheld. Additionally u is only acting directly on the first differential equation. Multiplying both

sides with $\boldsymbol{B}_D = \begin{bmatrix} 0 & 1 \end{bmatrix}$ gives a second order differential equation only dependent on φ and its derivatives.

$$\boldsymbol{B}_D \left(\boldsymbol{M} \begin{bmatrix} \Theta'' \dot{\varphi}^2 + \Theta' \ddot{\varphi} \\ \ddot{\varphi} \end{bmatrix} + \boldsymbol{C} \begin{bmatrix} \Theta' \dot{\varphi} \\ \dot{\varphi} \end{bmatrix} + \boldsymbol{G} \right) = 0 \tag{3.5}$$

Since $\boldsymbol{M}$ and $\boldsymbol{G}$ is only dependent on φ while $\boldsymbol{C}$ is linear in $\dot{\varphi}$, this can be observed from the Christoffel symbols, this equation can be written on a special form known as the $\alpha\beta\gamma$-equation.

$$\alpha(\varphi)\ddot{\varphi} + \beta(\varphi)\dot{\varphi}^2 + \gamma(\varphi) = 0 \tag{3.6}$$

From Equation 3.5 and Equation 1.4 it can be observed that α, β and γ will be given as:

$$\begin{aligned} \alpha(\varphi) &= \boldsymbol{M}_{21}\Theta' + \boldsymbol{M}_{22} \\ \beta(\varphi) &= \boldsymbol{M}_{21}\Theta'' + c_{112}\Theta'^2 + (c_{212} + c_{122})\Theta' + c_{222} \\ \gamma(\varphi) &= \boldsymbol{G}_2(\Theta, \varphi) \end{aligned} \tag{3.7}$$

Inserting the expressions given for the mass matrix, the Christoffel symbols and the gravity matrix gives:

$$\alpha(\varphi) = m_d \left(\left(s' \vec{\rho} \times \vec{\tau} - s'_f \frac{J_s}{R_d m_d} \right) \Theta' + s'^2 + s'^2_f \frac{J_s}{R_d^2 m_d} \right) \tag{3.8}$$

$$\beta(\varphi) = m_d \left(\left(s' \vec{\rho} \times \vec{\tau} - s'_f \frac{J_s}{R_d m_d} \right) \Theta'' - \vec{\rho}^{\top} \vec{\tau} s' \Theta'^2 + s' s'' + s'_f s''_f \frac{J_s}{R_d^2 m_d} \right) \tag{3.9}$$

$$\gamma(\varphi) = m_d g [\boldsymbol{R}(\Theta)\vec{\tau} s']_2 \tag{3.10}$$

A general formula for the kth $\alpha\beta\gamma$-equation is given in the appendix. Assuming α is never zero an expression for the second time derivative of φ on the desired trajectory can be found as:

$$\ddot{\varphi} = -\frac{1}{\alpha(\varphi)} (\beta(\varphi)\dot{\varphi}^2 + \gamma(\varphi)) \tag{3.11}$$

From equation Equation 3.8 it can be seen that α will be larger than zero if Equation 3.12 is upheld. Here it has been used that $s' > 0$, $s'_f > 0$ and $\vec{\rho} \times \vec{\tau} < 0$

$$\Theta' < -\frac{s'^2 + s'^2_f \frac{J_s}{R_d^2 m_d}}{s' \vec{\rho} \times \vec{\tau} - s'_f \frac{J_s}{R_d m_d}} \tag{3.12}$$

3.1.3 Equilibria of the $\alpha\beta\gamma$-equation

Rewriting the equation for the reduced dynamics on state space form will make the analysis of the equilibria of the $\alpha\beta\gamma$-equation easier. Using $x_1 = \varphi$ and $x_2 = \dot{\varphi}$ Equation 3.6 can be written as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{\beta(x_1)x_2^2 + \gamma(x_1)}{\alpha(x_1)} \end{bmatrix} \tag{3.13}$$

From the state space form it can be observed that the reduced dynamics has an equilibrium whenever $x_2 = \dot{\varphi} = 0$ and $\gamma(x_1) = \gamma(\varphi) = 0$. From Equation 3.10 it can be observed that for $\gamma(\varphi)$ to be zero the second element of $\boldsymbol{R}(\Theta)\vec{\tau}$ needs to be zero.

Since the VHC can be chosen freely to some degree these equilibria can be placed according to the desired behaviours. A natural choice for two of the equilibria is when the disc is in the valley of the frame and when the disc is on the top of the frame, $\varphi = n\pi$, $\vartheta = n\pi$ and $\varphi = \frac{\pi}{2} + n\pi$, $\vartheta = \frac{\pi}{2} + n\pi$, $n \in \mathbb{N}$. At the first point $\vec{\tau}$ will be pointing along the x-axis in the body frame, while at the second point $\vec{\tau}$ will point along the y-axis. For the second element of $\boldsymbol{R}(\Theta)\vec{\tau}$ to be zero at the first point Θ will need to be $n\pi$ while at the second point Θ will need to be equal $\frac{\pi}{2} + n\pi$.
The choice of virtual constraint will also potentially introduce additional equilibria between $\varphi = 0$ and $\varphi = \frac{\pi}{2}$.

For determining the properties of these equilibria a linearization and the extended Lyapunov's lemma will be used. It was shown in [10] that for the $\alpha\beta\gamma$-equation a center for the nonlinear system can be shown by linearization. This is not the case in general, and makes the $\alpha\beta\gamma$-equation particularly useful to derive possible periodic motions.
Linearizing Equation 3.6 about one of the equilibria gives Equation 3.14. Here z_1 represents the linearized φ while z_2 is the linearized $\dot{\varphi}$. In this equation it has been used that $\gamma(z_1^*) = 0$ and that $z_2^* = 0$. z_i^* represents the value of z_i at the equilibrium.

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\left.\frac{\frac{\partial\gamma(z_1)}{\partial z_1}}{\alpha(z_1)}\right|_{z_1=z_1^*} & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \tag{3.14}$$

Taking the partial derivative of γ with respect to φ gives the following, here the unit vector in y-direction, $\vec{k}$ has been used rather than the second element to make further derivations simpler.

$$\frac{\partial\gamma(\varphi)}{\partial\varphi} = m_d g\vec{k}^\top(\frac{\partial \boldsymbol{R}(\Theta)}{\partial\Theta}\Theta'\vec{\tau}s' + \boldsymbol{R}(\Theta)\frac{\partial\vec{\tau}}{\partial\varphi}s' + \boldsymbol{R}(\Theta)\vec{\tau}s'') \tag{3.15}$$

$$= m_d g\vec{k}^\top(\boldsymbol{R}'(\Theta)\Theta'\vec{\tau}s' + \boldsymbol{R}(\Theta)\vec{\kappa}s'^2 + \boldsymbol{R}(\Theta)\vec{\tau}s'') \tag{3.16}$$

Inserting for the two equilibria gives:

$$\left.\frac{\partial\gamma(\varphi)}{\partial\varphi}\right|_{\varphi=n\pi} = m_d g(\vec{\tau}\times\vec{k}s'\Theta' + \vec{k}^\top\vec{\kappa}) \tag{3.17}$$

$$\left.\frac{\partial\gamma(\varphi)}{\partial\varphi}\right|_{\varphi=\frac{\pi}{2}+n\pi} = m_d g(-\vec{k}^\top\vec{\tau}s'\Theta' + \vec{\tau}\times\vec{\kappa}) \tag{3.18}$$

$$\tag{3.19}$$

Here it has been used that $\vec{k}$ is orthogonal to $\boldsymbol{R}(\Theta)\vec{\tau}$ in the equilibra. From these equations it can be observed that Θ' can be used to chose the sign of γ at these equilibria.
The sign of the partial derivative of γ divided by α at the equilibrium will determine if the equilibrium is a center or a saddle. If γ' divided by α is smaller than zero it can be observed from Equation 3.14 that the eigenvalues of the linearization at the equilibrium have only a complex part, this gives a center. If it is larger than zero one eigenvalue will be positive and one negative, this gives a saddle [5].
Taking the partial derivative of γ divided by α gives:

$$\frac{\frac{\partial\gamma(\varphi)}{\partial\varphi}}{\alpha(\varphi)} = \frac{m_d g\vec{k}^\top(\boldsymbol{R}'(\Theta)\Theta'\vec{\tau}s' + \boldsymbol{R}(\Theta)\vec{\kappa}s'^2 + \boldsymbol{R}(\Theta)\vec{\tau}s'')}{m_d\left(\left(s'\vec{\rho}\times\vec{\tau} - s'_f\frac{J_s}{R_b m_d}\right)\Theta' + s'^2 + s'^2_f\frac{J_s}{R_b^2 m_d}\right)} \tag{3.20}$$

Since α is made to be larger than zero by the constraint in Equation 3.12 determining the properties of the equilibira is only dependent on the sign of the derivative of γ.

In total this gives a constraint for Θ' for all values of φ and on constraint to make each of the equilibira have the desired properties, either a center or a saddle.

3.1.4 Parameters of the VHC

The virtual holonomic constraint that will be used for the Butterfly Robot is given in Equation 3.21. It can be observed that this function will be equal to zero when φ is equal to zero and it will be equal to $\frac{\pi}{2}$ when φ is equal to $\frac{\pi}{2}$. It is also infinitely differentiable. c_i is constants that will be used to give the equilibria of the $\alpha\beta\gamma$-equation the desired properties.

$$\Theta(\varphi) = c_1\arctan\{c_2\sin(2\varphi) + c_3\sin(4\varphi) + c_4\sin(6\varphi) + c_5\sin(8\varphi)\} + \varphi \tag{3.21}$$

To determine the parameters of the VHC constrained optimization will be used. The constraints that will be used are $\alpha(\varphi)$ larger than zero and that the equilibria have the desired properties

discussed previously. In addition some additional equilibria will be placed between $\varphi = 0$ and $\varphi = \frac{\pi}{2}$.

To make the nominal trajectory as slow as possible the region of attraction for the centers should not be too big. This will be achieved by minimizing with respect to the derivative of γ with respect to φ. This gives the following optimization given in Equation 3.22.

$$\min_{c_1,c_2,c_3,c_4,c_5} \quad \sum_{\varphi_{center}} \gamma'(\varphi_{center}) + \sum_{\varphi_{saddle}} \gamma'(\varphi_{saddle}) \tag{3.22a}$$

$$\text{subject to} \quad \alpha(\varphi) > 0, \tag{3.22b}$$

$$\gamma'(\varphi_{center}) > 0, \tag{3.22c}$$

$$\gamma'(\varphi_{saddle}) < 0, \tag{3.22d}$$

$$\gamma(\varphi_{center}) = 0, \tag{3.22e}$$

$$\gamma(\varphi_{saddle}) = 0, \tag{3.22f}$$

To implement these optimization problems YALMIP was used [6]. The Matlab solver FMINCON was used as a solver.

3.1.5 Shape of the Butterfly Robot

To be able to use the model and controller derived earlier a description of the frame is needed, that is the function $\delta(\phi)$. In [14] a π periodic trigonometric function was used for this. This function is given in Equation 3.23.

$$\delta(\phi) = a - b\cos(2\phi) \tag{3.23}$$

This function has the advantage of being smooth and infinitely differentiable. This is important as the model depends on the third derivative of this function. In this text three different descriptions will be tested, the function given in Equation 3.23, the cosine series given in Equation 3.24 and a spline interpolation of the actual frame.

$$\delta = a_0 + \sum_{n=1}^{4} a_n \cos(2n\phi) \tag{3.24}$$

In Figure 3.1 the function δ and it's derivatives are plotted. Data of the actual frame is shown in blue, Equation 3.23 is shown in red while a cosine series with five elements is shown in yellow. To find the derivatives of the numerical data of the frame a finite differences scheme was used. The two cosine functions were curve fitted to data from the frame using Matlab's curve fitting tool[8]. As can be seen from the plots both the functions match the values for the frame well, but becomes progressively worse at describing the frames derivative as these increase in order. Especially at a little before $\frac{\pi}{4}$ and a little after $\frac{3\pi}{4}$ the values of the approximations are not good.

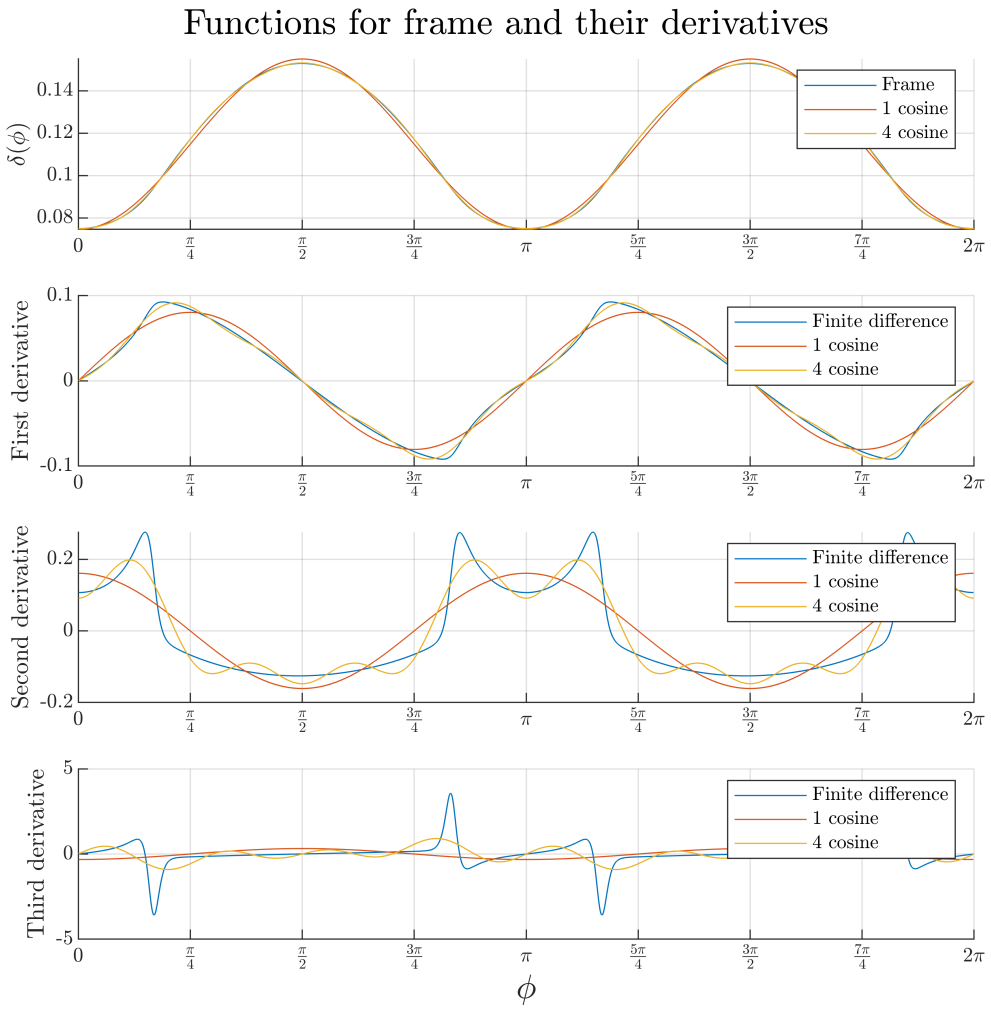

Figure 3.1: Plots of the frame and its derivatives

The statement that $\frac{\partial g(\phi)}{\partial \phi}$ should be larger than zero can now be checked for the three different representations of the frame. A plot showing the three functions from $\phi = 0$ to $\phi = \pi$ is shown in Figure 3.2. Here the radius of the disc has been set equal to 9.5 millimeters. It can be observed from this plot that $g'(\phi)$ will be larger than zero for all $\phi \in \mathbb{R}$, this because $g'(\phi)$ is π periodic.

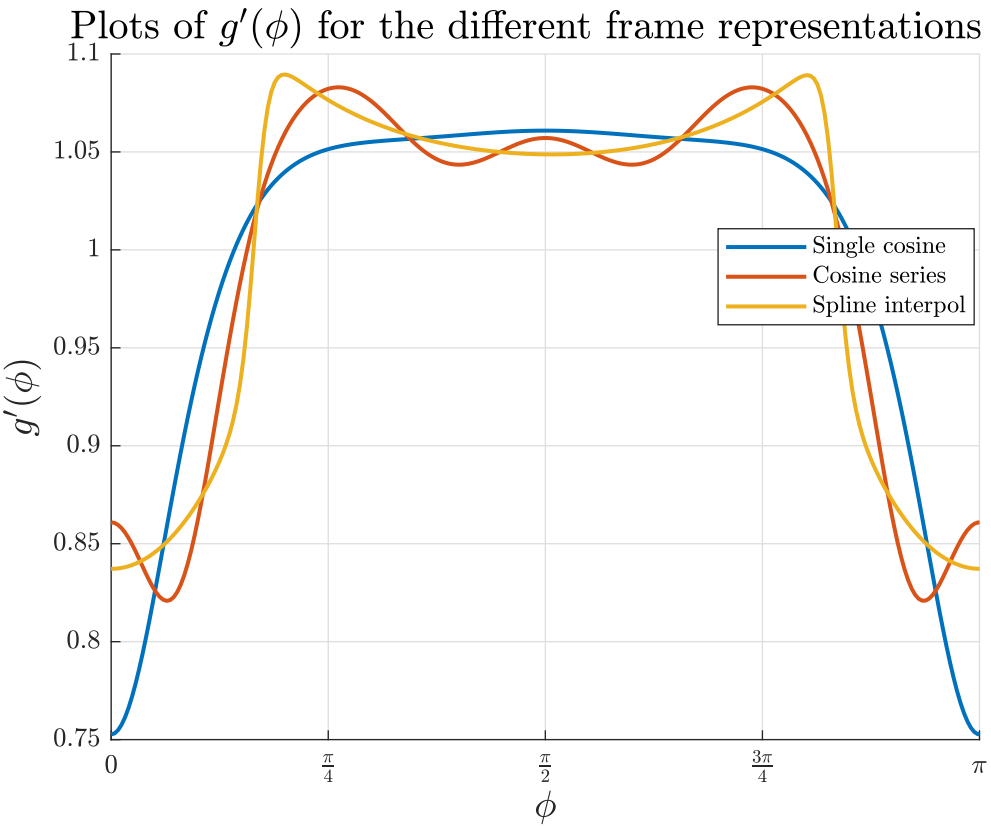

Figure 3.2: Plot of $\frac{\partial g(\phi)}{\partial \phi}$ for the different representations of the frame.

3.1.6 Continuous rolling for the sphere on the Butterfly Robot

The optimization problem for the virtual holonomic constraint will now be used to find a nominal trajectory for the Butterfly Robot. Three phase planes will be considered, one for each of the previously discussed descriptions of the frame. The same VHC will be used for all of them.

Using the optimization problem one set of possible parameters for the virtual holonomic constraint was found to be:

$$c_1 = -4.95 \times 10^{1},\ c_2 = 1.17 \times 10^{-2},\ c_3 = 2.96 \times 10^{-3},\ c_4 = 4.73 \times 10^{-4},\ c_5 = -5.47 \times 10^{-5} \tag{3.25}$$

In Figure 3.3 a plot of the phase plane with Equation 3.23 used to describe the frame is shown. Figure 3.4 shows the phase plane with the cosine series used while Figure 3.5 shows the phase plane with the interpolation used to describe the frame.

From these phase planes it is worth noting that there can be found trajectories that result in continuous motion for all three cases. Oscillations about $\frac{\pi}{2}$ and two other equilibria can also be found. The cases where the initial condition results in continuous motion will be studied in this text. It is also worth noting that the phase plane with the cosine series matches the interpolation better than the single cosine. In addition the equilibria of the system does not seem to change character between the three different descriptions, that is a saddle remains a saddle while a center remains a center.

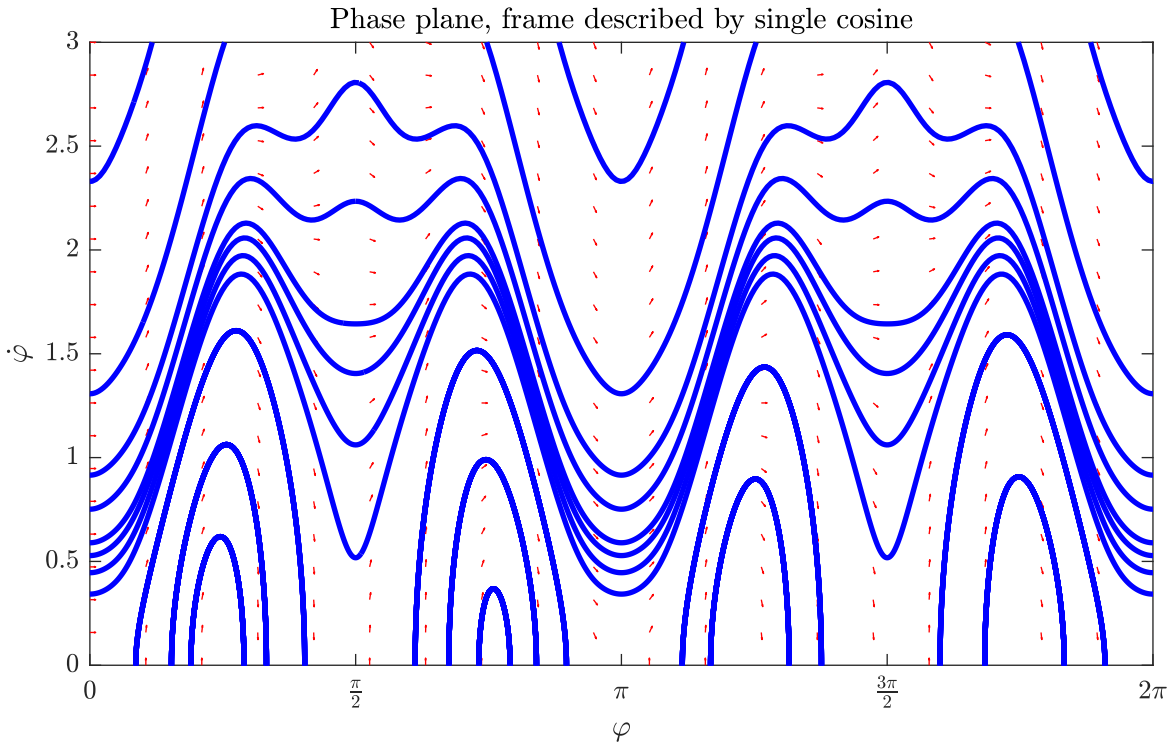

Figure 3.3: Phase plane with δ described by Equation 3.23

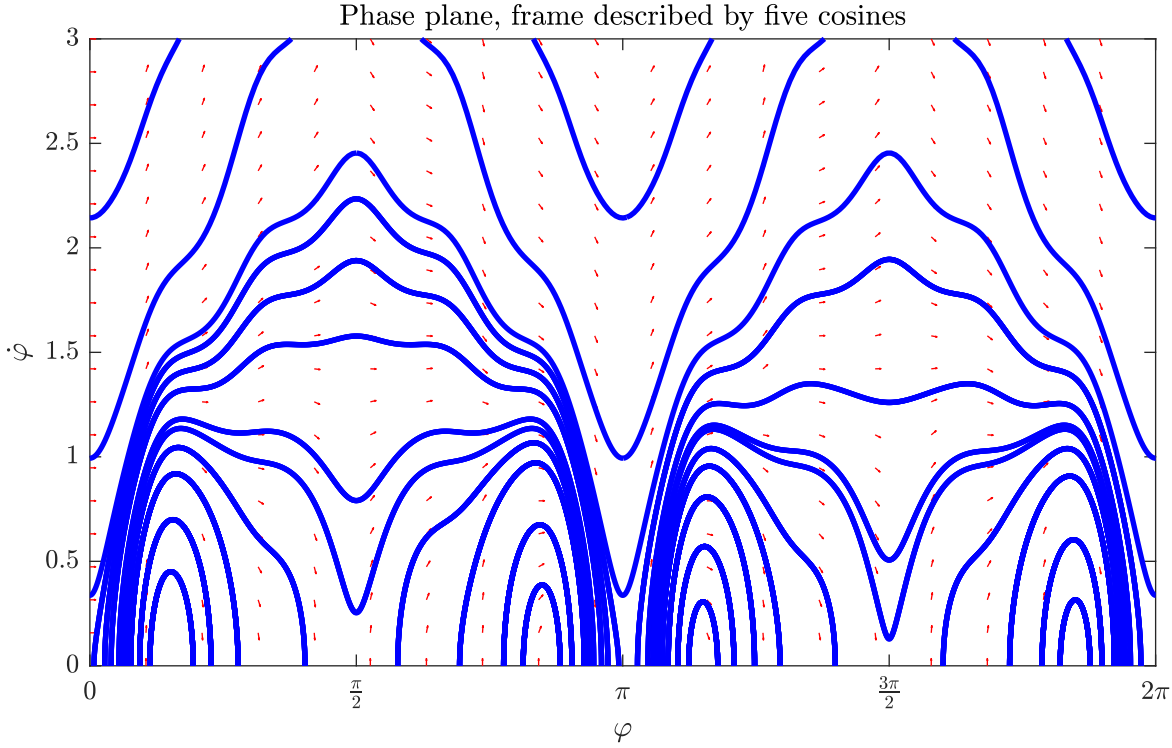

Figure 3.4: Phase plane with δ described by a cosine series consisting of five terms.

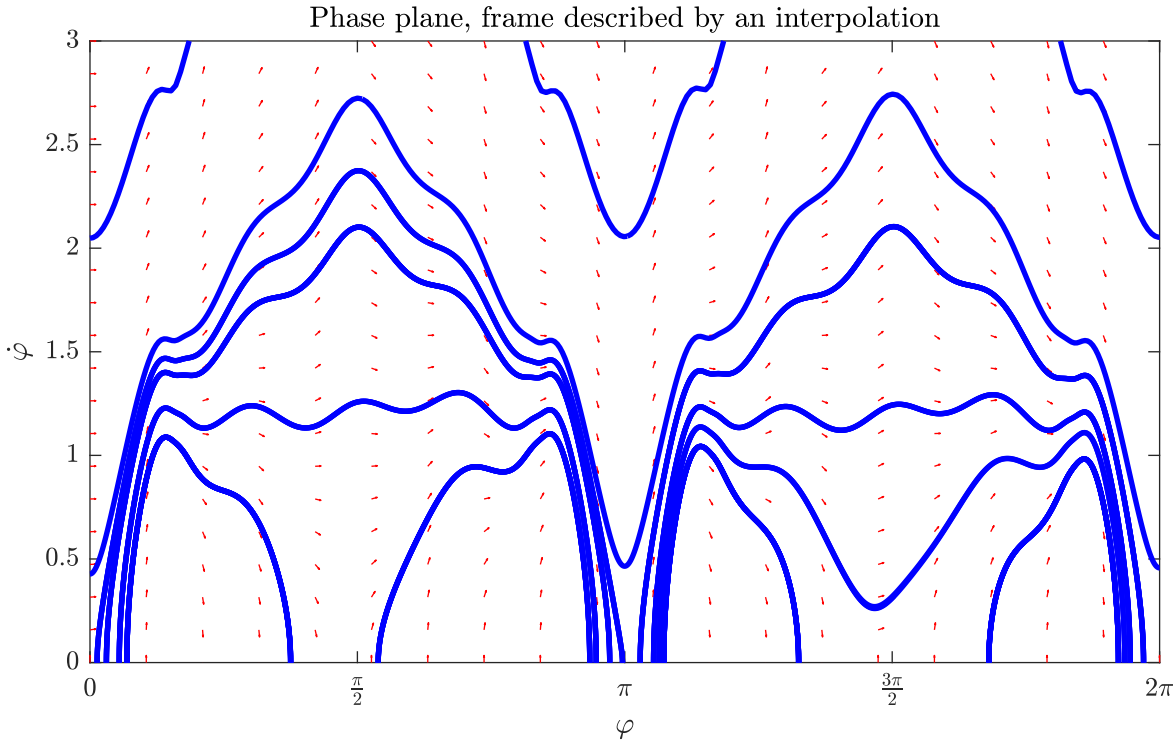

Figure 3.5: Phase plane with δ given as a spline interpolation of the data of the frame.

3.2 Controller for the Butterfly Robot

A controller that is able to stabilize the nominal trajectory found earlier will now be derived. This will be done in two stages. First a feedforward term will be derived that will keep the system on the nominal trajectory if it already is on the nominal trajectory. As this is a feedforward it will not be able to handle any noise or deviation from the nominal trajectory. To deal with this a feedback linearization of a set of transverse coordinates will be introduced. The resulting time varying linear system will then be controlled using a time varying LQR. An illustration of the resulting control structure is given in Figure 3.6.

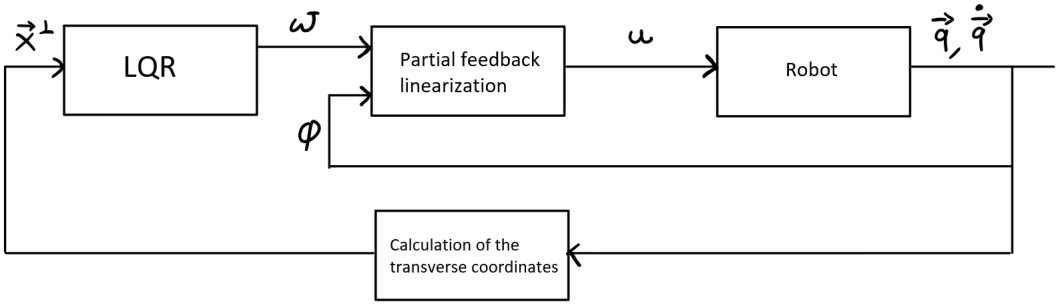

Figure 3.6: Control structure for the Butterfly Robot

3.2.1 Feedforward on the nominal trajectory

Multiplying both sides of Equation 3.4 with the transpose of $\boldsymbol{B}$ gives an expression for u. Assuming the system is on the nominal trajectory and inserting the $\alpha\beta\gamma$-equation gives the following expression for the nominal input. Note that all expressions are evaluated as a function of φ. Here $\star$ denotes a function on the nominal trajectory.

$$u_\star(\varphi) = \boldsymbol{B}^\top \left(-\boldsymbol{M} \begin{bmatrix} \Theta' \\ 1 \end{bmatrix} \frac{\beta \dot{\varphi}_\star^2 + \gamma}{\alpha} + \boldsymbol{M} \begin{bmatrix} \Theta'' \dot{\varphi}_\star \\ 0 \end{bmatrix} + \boldsymbol{C} \begin{bmatrix} \Theta' \dot{\varphi}_\star \\ \dot{\varphi}_\star \end{bmatrix} + \boldsymbol{G} \right) \tag{3.26}$$

This expression can be further simplified to Equation 3.27 by using the expressions for α, β and γ. The derivation is given in the appendix.

$$u_\star(\varphi) = k \left(\begin{bmatrix} k^{-1} & -1 \end{bmatrix} \boldsymbol{M} \begin{bmatrix} \Theta'' \dot{\varphi}_\star^2 \\ 0 \end{bmatrix} + \begin{bmatrix} k^{-1} & -1 \end{bmatrix} \boldsymbol{C} \begin{bmatrix} \Theta' \dot{\varphi}_\star \\ \dot{\varphi}_\star \end{bmatrix} + \begin{bmatrix} k^{-1} & -1 \end{bmatrix} \boldsymbol{G} \right) \tag{3.27}$$

Here k is given by the following expression:

$$k = \frac{\begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{M} \begin{bmatrix} \Theta' & 1 \end{bmatrix}}{\begin{bmatrix} \Theta' & 1 \end{bmatrix} \boldsymbol{M} \begin{bmatrix} 0 \\ 1 \end{bmatrix}} \tag{3.28}$$

3.2.2 Transverse Coordinates

The feedforward derived earlier will only work if the system already is on the desired trajectory. It will also be unstable on this trajectory, a small deviation or disturbance will make it leave the nominal trajectory and thus make the feedforward useless. A set of coordinates that stabilizes this trajectory is thus needed.

Since the Butterfly Robot is a system with two degrees of freedom three transverse coordinates are needed to describe how far away the system is from the desired trajectory. In addition a motion generator will be needed.

An alternative set of coordinates for the system given in Equation 2.37 can be given as:

$$\vec{\xi}(\vec{q},\dot{\vec{q}}) = \begin{bmatrix} \vartheta - \Theta(\varphi) \\ \varphi \\ \dot{\vartheta} - \Theta'(\varphi)\dot{\varphi} \\ \dot{\varphi} - \dot{\varphi}_\star \end{bmatrix} \tag{3.29}$$

This is the same set of coordinates used to describe the Butterfly Robot in [14]. They showed that that this new set of coordinates is valid for the whole state space, that is the transformation from $\vec{\xi}$ to $\{\vec{q},\dot{\vec{q}}\}$ exists and the Jacobian has full rank.

In Equation 3.29 it can be observed that three of the new states will be zero when the system is on the nominal trajectory. In addition if a nominal trajectory with continuous motion in one of the directions is chosen $\dot{\varphi}$ will always be larger than zero (alternatively always smaller than zero if a negative $\dot{\varphi}$ is chosen as the initial condition from one of the phase planes.) Thus φ will be a monotonically increasing function on the nominal trajectory.

φ will therefore be chosen as the motion generator for the system, while the three remaining states will be used as transverse coordinates, $\vec{x}^{\perp} = \begin{bmatrix} y_1 & y_2 & z \end{bmatrix}^\top$.

$$y_1 = \vartheta - \Theta(\varphi) \tag{3.30}$$
$$y_2 = \dot{\vartheta} - \Theta'(\varphi)\dot{\varphi} \tag{3.31}$$
$$z = \dot{\varphi} - \dot{\varphi}_\star(\varphi) \tag{3.32}$$

3.2.3 Dynamics for the transverse coordinates

Differentiating the expressions given for the transverse coordinates with respect to time gives the following:

$$\dot{y}_1 = y_2 \tag{3.33}$$
$$\dot{y}_2 = \ddot{\vartheta} - \Theta''(\varphi)\dot{\varphi}^2 - \Theta'(\varphi)\ddot{\varphi} = \ddot{y} \tag{3.34}$$
$$\dot{z} = \ddot{\varphi} - \frac{\partial \dot{\varphi}_\star(\varphi)}{\partial \varphi}\dot{\varphi} \tag{3.35}$$

Inserting that $\dot{\varphi} = z + \dot{\varphi}_\star$ and writing the expression for $\dot{y}_2$ in a more convenient form gives:

$$\dot{y}_1 = y_2 \tag{3.36}$$
$$\dot{y}_2 = \begin{bmatrix} 1 & -\Theta'(\varphi) \end{bmatrix} \begin{bmatrix} \ddot{\vartheta} \\ \ddot{\varphi} \end{bmatrix} - \Theta''(\varphi)(z + \dot{\varphi}_\star)^2 \tag{3.37}$$
$$\dot{z} = \ddot{\varphi} - \frac{\partial \dot{\varphi}_\star(\varphi)}{\partial \varphi}(z + \dot{\varphi}_\star) \tag{3.38}$$

$\dot{\varphi}_\star$ is a solution of the $\alpha\beta\gamma$-equation . This equation has an analytic solution given by Equation 3.39. This was shown in [11]. Note that the system is assumed to be on the desired trajectory for this derivation.

$$\dot{\varphi}_\star^2 = \exp\left\{-2\int_{\varphi_0}^{\varphi} \frac{\beta(w)}{\alpha(w)}dw\right\}\left(\dot{\varphi}_{0\star}^2 - \int_{\varphi_0}^{\varphi} \frac{2\gamma(w)}{\alpha(w)} \exp\left\{2\int_{\varphi_0}^{w} \frac{\beta(v)}{\alpha(v)}dv\right\}dw\right) \tag{3.39}$$

Differentiating $\dot{\varphi}_\star^2$ with respect to φ gives:

$$\frac{\partial \dot{\varphi}_\star^2}{\partial \varphi} = 2\dot{\varphi}_\star \frac{\partial \dot{\varphi}_\star}{\partial \varphi} \tag{3.40}$$

The partial derivative of $\dot{\varphi}_\star$ can thus be found as:

$$\frac{\partial \dot{\varphi}_\star}{\partial \varphi} = \frac{\frac{\partial \dot{\varphi}_\star^2}{\partial \varphi}}{2\dot{\varphi}_\star} \tag{3.41}$$

Differentiating Equation 3.39 with respect to φ gives the following, derivation is given in the appendix.

$$\frac{\partial \dot{\varphi}_\star^2}{\partial \varphi} = -2\dot{\varphi}_\star^2 \frac{\beta(\varphi)}{\alpha(\varphi)} - 2\frac{\gamma(\varphi)}{\alpha(\varphi)} \tag{3.42}$$

The expression for the partial derivative of $\dot{\varphi}_\star$ is thus given as:

$$\frac{\partial \dot{\varphi}_\star}{\partial \varphi} = -\dot{\varphi}_\star \frac{\beta(\varphi)}{\alpha(\varphi)} - \frac{\gamma(\varphi)}{\dot{\varphi}_\star \alpha(\varphi)} \tag{3.43}$$

Inserting this in the expression for $\dot{z}$ gives:

$$\dot{z} = \ddot{\varphi} + \dot{\varphi}_\star \left(\frac{\beta}{\alpha} + \frac{\gamma}{\dot{\varphi}_\star^2 \alpha} \right) (z + \dot{\varphi}_\star) \tag{3.44}$$

When the system is not on the desired trajectory the $\alpha\beta\gamma$-equation can be written as given in Equation 3.45 [12]. Note that W is zero when the system is on the desired trajectory.

$$\alpha(\varphi)\ddot{\varphi} + \beta(\varphi)\dot{\varphi}^2 + \gamma(\varphi) = W \tag{3.45}$$

Inserting the $\alpha\beta\gamma$-equation when the system is not on the trajectory into the equation for $\dot{z}$ gives:

$$\dot{z} = \frac{-\beta\dot{\varphi}^2 - \gamma}{\alpha} + \frac{W}{\alpha} + \dot{\varphi}_\star \left(\frac{\beta}{\alpha} + \frac{\gamma}{\dot{\varphi}_\star^2 \alpha} \right) (z + \dot{\varphi}_\star) \tag{3.46}$$

$$= \frac{-\beta(z + \dot{\varphi}_\star)^2 - \gamma}{\alpha} + \frac{W}{\alpha} + \dot{\varphi}_\star \left(\frac{\beta}{\alpha} + \frac{\gamma}{\dot{\varphi}_\star^2 \alpha} \right) (z + \dot{\varphi}_\star) \tag{3.47}$$

$$= \frac{-\beta z^2 - 2\beta z \dot{\varphi}_\star - \beta\dot{\varphi}_\star^2 - \gamma}{\alpha} + \frac{W}{\alpha} + \dot{\varphi}_\star \left(\frac{\beta}{\alpha} + \frac{\gamma}{\dot{\varphi}_\star^2 \alpha} \right) (z + \dot{\varphi}_\star) \tag{3.48}$$

$$= \frac{W}{\alpha} + \frac{1}{\alpha} \left(-\beta z^2 - \beta z \dot{\varphi}_\star + \frac{z\gamma}{\dot{\varphi}_\star} \right) \tag{3.49}$$

To be able to use these dynamics an expression for W is needed. It was shown in [12] that W can be written as:

$$W = g_{y_1} y_1 + g_{y_2} y_2 + g_{\dot{y}_2} \dot{y}_2 \tag{3.50}$$

The dynamics for the transverse coordinates can thus be written as given in Equation 3.51. Note that $\dot{y}_2$ has been renamed to w, this is because it will be used as a virtual control input to the system of transverse coordinates.

$$\begin{aligned} \dot{y}_1 &= y_2 \\ \dot{y}_2 &= \begin{bmatrix} 1 & -\Theta'(\varphi) \end{bmatrix} \begin{bmatrix} \ddot{\vartheta} \\ \ddot{\varphi} \end{bmatrix} - \Theta''(\varphi)(z + \dot{\varphi}_\star)^2 = w \\ \dot{z} &= \frac{1}{\alpha} \left(g_{y_1} y_1 + g_{y_2} y_2 + g_w w - \beta z^2 - \beta z \dot{\varphi}_\star + \frac{z\gamma}{\dot{\varphi}_\star} \right) \end{aligned} \tag{3.51}$$

To find the expressions for g_y, $g_{\dot{y}}$ and g_w the expressions for ϑ and it's derivatives given in Equation 3.52 is inserted in the $\alpha\beta\gamma$-equation in addition to φ and it's derivatives.

$$\begin{aligned} \vartheta &= y_1 + \Theta(\varphi) \\ \dot{\vartheta} &= y_2 + \Theta'(\varphi)\dot{\varphi} \\ \ddot{\vartheta} &= w + \Theta''(\varphi)\dot{\varphi}^2 + \Theta'(\varphi)\ddot{\varphi} \end{aligned} \tag{3.52}$$

This results in the expressions given Equation 3.53, derivation is given in the appendix. These are similar to the expressions found in [14].

$$\begin{aligned} g_y &= m_b g \begin{bmatrix} -\cos(\varphi + \frac{y}{2}) & \sin(\varphi + \frac{y}{2}) \end{bmatrix} \operatorname{sinc}\left(\frac{y}{2}\right) \vec{\tau} s' \\ g_{\dot{y}} &= -m_b \vec{\rho}^\top \vec{\tau} s' (\dot{y} + 2\Theta'(\varphi)\dot{\varphi}) \\ g_w &= -\left(s' \vec{\rho} \times \vec{\tau} - s'_f \frac{J_s}{R_b m_b} \right) \end{aligned} \tag{3.53}$$

A new input u will be needed to control the system. This will be given by Equation 3.54 where k is given in Equation 3.28. This is the same as the feedforward derived earlier when the system is on the nominal trajectory. A similar controller was used in [14] to control the Butterfly Robot, it was shown in [12] that such a controller is able to achieve stabilization of the original nonlinear system in a neighborhood of the desired trajectory.

$$u(\varphi, w) = k\left(\begin{bmatrix} k^{-1} & -1 \end{bmatrix} \boldsymbol{M} \begin{bmatrix} w + \Theta''\dot{\varphi}_\star^2 \\ 0 \end{bmatrix} + \begin{bmatrix} k^{-1} & -1 \end{bmatrix} \boldsymbol{C} \begin{bmatrix} \Theta'\dot{\varphi}_\star + y_2 \\ \dot{\varphi}_\star \end{bmatrix} + \begin{bmatrix} k^{-1} & -1 \end{bmatrix} \boldsymbol{G}\right) \tag{3.54}$$

With this the problem of controlling the butterfly robot to the nominal trajectory has been reduced to controlling the system of transverse coordinates to the trivial solution, that is all transverse coordinates equal to zero. w will be used as a virtual input to achieve this.
In addition since the system will be controlled in terms of φ rather than time a change from the time derivative to the derivative with respect to φ is needed. Her $x^\perp$ is introduced as the vector consisting of the transverse coordinates. Note that for this controller to be available the transformation from time to φ needs to exist, that is φ needs to be monotonically increasing.

$$\dot{\vec{x}}^\perp = \frac{d}{dt}\vec{x}^\perp = \frac{d}{dt}\vec{x}^\perp \frac{\partial \varphi}{\partial \varphi} = \frac{\partial}{\partial \varphi}\vec{x}^\perp \frac{d\varphi}{dt} = \frac{\partial}{\partial \varphi}\vec{x}^\perp \dot{\varphi} \tag{3.55}$$

Assuming $\dot{\varphi}$ is bigger than zero for all time gives that the derivative of $\vec{x}^\perp$ can be written as:

$$\frac{\partial}{\partial \varphi}\vec{x}^\perp = \frac{\vec{f}(\vec{x}^\perp, w)}{\dot{\varphi}} \tag{3.56}$$

Here the vector $\vec{f}$ represents the dynamics of the transverse coordinates given in Equation 3.51.

3.2.4 Linearization of the transverse coordinates

Linearizing the dynamics for the transverse coordinates given in Equation 3.51 along the nominal trajectory, that is all the transverse coordinates equal to zero, gives the following:

$$\frac{d}{d\varphi}\vec{z}^\perp = \boldsymbol{A}(\varphi)\vec{z}^\perp + \boldsymbol{b}(\varphi)w \tag{3.57}$$

$$\boldsymbol{A}(\varphi) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ \frac{\bar{g}_{y1}}{\bar{\alpha}} & \frac{\bar{g}_{y2}}{\bar{\alpha}} & \frac{\bar{\gamma} - \bar{\beta}\dot{\varphi}_\star}{\bar{\alpha}\dot{\varphi}_\star} \end{bmatrix} \frac{1}{\dot{\varphi}_\star} \tag{3.58}$$

$$\boldsymbol{b}(\varphi) = \begin{bmatrix} 0 & 1 & \frac{\bar{g}_w}{\bar{\alpha}} \end{bmatrix}^\top \frac{1}{\dot{\varphi}_\star} \tag{3.59}$$

Here a bar over the functions indicates that they are evaluated on the nominal trajectory. The expressions for g_y, $g_{\dot{y}}$ and g_w becomes a bit simpler when evaluated on the nominal trajectory:

$$\begin{aligned} \bar{g}_y &= m_b g \begin{bmatrix} -\cos(\varphi) & \sin(\varphi) \end{bmatrix} \vec{\tau} s' \\ \bar{g}_{\dot{y}} &= -2m_b \vec{\rho}^\top \vec{\tau} s' \Theta'(\varphi)\dot{\varphi}_\star \\ \bar{g}_w &= -\left(s'\vec{\rho} \times \vec{\tau} - s'_f \frac{J_s}{R_b m_b}\right) \end{aligned} \tag{3.60}$$

To control this linear φ-varying system a φ-dependent linear quadratic regulator (LQR) will be used. That is a controller on the form:

$$w(\varphi) = \boldsymbol{b}(\varphi) X(\varphi) \vec{x}^\perp \tag{3.61}$$

Here $X(\varphi)$ is a stabilizing solution of the periodic Riccati differential equation.[12]

3.2.5 Periodic LQR

It was shown in [12] that the controller given in Equation 3.62 is able to stabilize a periodic linear time varying system that is controllable.

$$w(t) = -R(t)^{-1}B(t)X(t)\vec{x}^{\perp} \tag{3.62}$$

Here $X(t)$ is the solution to the periodic Riccati differential equation given as:

$$-\dot{X}(t) = A(t)^{\top}X(t) + X(t)A(t) - X(t)B(t)R(t)^{-1}B(t)X(t) + Q(t) \tag{3.63}$$

It was shown in [4] that an approximation to the solution of this matrix differential equation can be found through semi definite programming (sdp). Here the problem of finding the solution $X(t)$ is reformulated as a discrete optimization problem. X(t) is written as N matrices X_n over the time interval $t \in [0, T]$ and $n \in [1, N]$. Note that for the Butterfly Robot this will be a period in φ rather than time. The period will be π.
Using a Fourier series the solution can be written as $X = \sum_{k=-M}^{M} \exp(ik\omega t)F_k(X)$ where F_0 is a symmetric real matrix, $F_k,\ k = 1..M$ is a symmetric complex matrix, F_{-k} is the complex conjugate of F_k and $\omega = \frac{2\pi}{T}$.[4]
It was further shown in [4] that the following maximization problem converges to the solution of the periodic Riccati differential equation as M goes to infinity and $\frac{N}{M^3}$ goes to infinity.

$$\begin{aligned}
&\max && trace(F_0(X_n)) && \text{(3.64a)}\\
&\text{subject to} && \mathcal{L}(X, t_j) \geq 0,\ \ t_j = (j-1)\frac{T}{N}\ j = 1, .., N && \text{(3.64b)}\\
& && -d\boldsymbol{I} \leq X(t_j) \leq d\boldsymbol{I} && \text{(3.64c)}
\end{aligned}$$

Here $\mathcal{L}(X, t_j)$ is known as the Riccati operator and is given as:

$$\mathcal{L}(X, t) = \begin{bmatrix} \frac{d}{dt}X(t) + X(t)A(t) + A^{\top}(t)H(t) + Q(t) & X(t)B(t) \\ B^{\top}X(t) & R(t) \end{bmatrix} \tag{3.65}$$

To make the solution dependent on the states, φ will be used rather than time. This can be done as φ is a monotonically increasing function on all the intervals that will be used. This will give a solution consisting of N matrices $X_n(\varphi),\ n \in 1, .., N$.
The implementation of the optimization problem was done in Matlab with YALMIP [6], it was solved with SDPT3 [15]. It can be found at https://github.com/GustavOften/ttk4900_master_thesis.

3.3 Observers for the transverse coordinates

To be able to use the controller derived earlier measurements of the states of the system are needed. These measurements are prone to noise, especially the derivatives, so an observer will be introduced to remove some of this noise. Since the control system is not using the states directly for control but rather the set of transverse coordinates an observer for these are enough to be able to control the system, note that φ is still needed for the partial feedback linearization.

As the linearized transverse coordinates can be used to describe the system when the system is close to the nominal trajectory a linear observer will be introduced to estimate the linearized transverse coordinates. The control structure with the observer included is shown in Figure 3.7.

Since all of the states in the Butterfly Robot are measured, all of the states in the linearized transverse coordinates can be calculated from the measurements. This gives that the $\boldsymbol{C}$ can be taken as the identity. This gives that the system of transverse coordinates are observable.

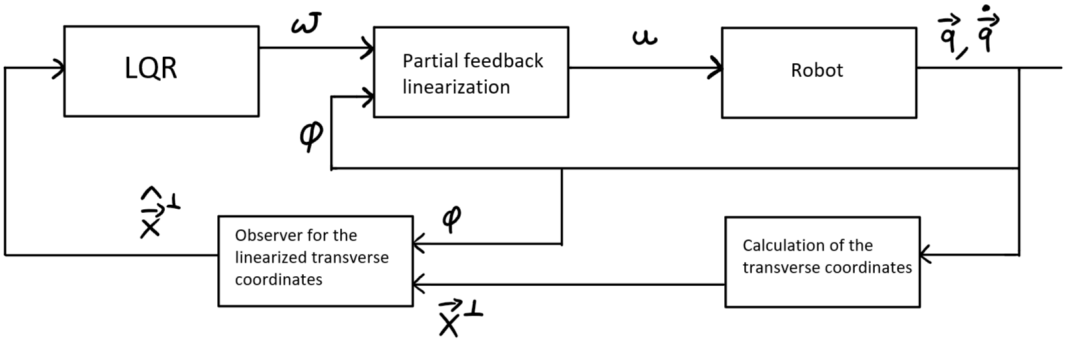

Figure 3.7: Control structure with observer included

3.3.1 Luenberger observer

A Luenberger observer can be used to estimate the linearized transverse coordinates when the system is close to the nominal trajectory. The Luenberger observer is given in Equation 3.66 [2]. Her the matrix $\boldsymbol{L}$ is a positive definite gain matrix, while $\boldsymbol{A}$ and $\boldsymbol{B}$ is the matrices for the linearization of the transverse coordinates derived earlier. In this text a constant matrix will be used for $\boldsymbol{L}$, though it could be made ϕ varying. Since all of the states of the system can be measured and the transverse coordinates can be calculated from the measurements the matrix $\boldsymbol{C}$ will be the identity matrix. The measurements $\vec{y}$ are the transverse coordinates calculated from Equations 3.30, 3.31 and 3.32.

$$\dot{\hat{x}}^{\perp} = \boldsymbol{A}(\varphi)\hat{x}^{\perp} + \boldsymbol{B}(\varphi)u + \boldsymbol{L}(\vec{y} - \boldsymbol{C}\hat{x}^{\perp}) \tag{3.66}$$

3.3.2 Kalman filter

A Kalman filter is an optimal estimator in the same way that a LQR is an optimal controller if the covariances of the measurements noise and process noise are known. This estimator assumes a system on the form given in Equation 3.67. Here $\vec{v}(t)$ and $\vec{w}(t)$ are white noise processes with covariance matrices given by $\boldsymbol{Q}(t)$ and $\boldsymbol{R}(t)$ respectively.

$$\begin{aligned} \dot{\vec{x}} &= \boldsymbol{A}(t)\vec{x} + \boldsymbol{B}(t)u + \boldsymbol{G}(t)\vec{v}(t) \\ y(t) &= \boldsymbol{C}(t)\vec{x} + \boldsymbol{E}(t)\vec{w}(t) \end{aligned} \tag{3.67}$$

Note that this is not the standard way to write these equations. The matrix $\boldsymbol{E}$ is usually not included. But, since the transverse coordinates are not directly measurable the noise in these will

be multiplied with a nonlinear function. Assuming the following:

$$\vartheta_{measured} = \vartheta_{true} + w_{\vartheta} \tag{3.68}$$
$$\dot{\vartheta}_{measured} = \dot{\vartheta}_{true} + w_{\dot{\vartheta}} \tag{3.69}$$
$$\varphi_{measured} = \varphi_{true} \tag{3.70}$$
$$\dot{\varphi}_{measured} = \dot{\varphi}_{true} + w_{\dot{\varphi}} \tag{3.71}$$

The measurement of φ is assumed to be true, as this is the motion generator for this system. This gives that the measurements of the transverse coordinates will be given as:

$$y_{1,measured} = y_{1,true} + w_{\vartheta} \tag{3.72}$$
$$y_{2,measured} = y_{2,true} + w_{\dot{\vartheta}} - \Theta'(\varphi) w_{\dot{\varphi}} \tag{3.73}$$
$$z_{measured} = z_{true} + w_{\dot{\varphi}} \tag{3.74}$$

The continuous time Kalman filter will then be given as follows [9].

$$\dot{\hat{x}} = \boldsymbol{A}(t)\hat{x} + \boldsymbol{B}(t)u + \boldsymbol{L}(t)(\vec{y} - \boldsymbol{C}(t)\hat{x}) \tag{3.75}$$

The main difference between this and the Luenberger observer is how the matrix $\boldsymbol{L}$ is obtained. For the Luenberger this is often tuned to to fit the system while in the Kalman filter a Riccati equation is solved to find the optimal gain matrix, in this case a periodic Riccati differential equation (PRDE). The same procedure as was used to find the PRDE for the LQR will be used for the Kalman filter.

With this the Kalman gain $\boldsymbol{L}(t)$ can be found as:

$$\boldsymbol{L}(t) = \boldsymbol{P}^{\top}\boldsymbol{C}\left(\boldsymbol{E}\boldsymbol{R}\boldsymbol{E}^{\top}\right)^{-1} \tag{3.76}$$

Where $\boldsymbol{P}(t)$ is found from the periodic Riccati differential equation:

$$\dot{\boldsymbol{P}} = \boldsymbol{A}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}^{\top} + \boldsymbol{G}\boldsymbol{Q}\boldsymbol{G}^{\top} - \boldsymbol{P}\boldsymbol{C}\left(\boldsymbol{E}\boldsymbol{R}\boldsymbol{E}^{\top}\right)^{-1}\boldsymbol{C}^{\top}\boldsymbol{P}^{\top} \tag{3.77}$$

Under the assumption that $\vec{w}$ is given as $\vec{w} = \begin{bmatrix} w_{\vartheta} & w_{\dot{\vartheta}} & w_{\dot{\varphi}} \end{bmatrix}^{\top}$ the matrix $\boldsymbol{E}$ will be given as follows for the Butterfly Robot.

$$\boldsymbol{E} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\Theta'(\varphi) \\ 0 & 0 & 1 \end{bmatrix} \tag{3.78}$$

Chapter 4

Experiment and results

4.1 Physical system of the Butterfly Robot

Now some of the parameters for the physical robot located at the Norwegian University of Science and Technology will be found.

4.1.1 Parameters

With the help of a pair of calipers the diameter of the sphere was measured to be 33 millimeters. The distance between the plates of frame was measured to be 27 millimeters. According to Equation A.1 this gives an effective rolling radius, R_d, of 9.5 millimeters.

The mass of the sphere was taken as 3 grams, this is the same as was used in [14].

The inertia of the sphere was found as the inertia of a spherical shell. That is:

$$J_s = \frac{2}{3} m_s R_s^2 = 1.805 \times 10^{-7} [kgm^2] \tag{4.1}$$

4.1.2 Friction in rotation of the frame

To determine the friction for the rotation of the frame a PI-controller was used. This controller was used to keep the angular velocity of the frame constant at one radian per second. Assuming the frame can be described by the model given in Equation 4.4 the friction will then be given as the negative of the required torque. The friction was found as the average over a period of forty seconds. That is:

$$F_f = -\frac{1}{n} \sum_{i=1}^{n} \tau_i \tag{4.2}$$

With this the friction was found to be given as -0.002[Nm] for positive rotation of the frame and 0.0032[Nm] for negative rotation of the frame.

$$F_f = \begin{cases} -0.002, & \text{if } \vartheta > 0 \\ 0.0032, & \text{if } \vartheta < 0 \end{cases} \tag{4.3}$$

To counteract this friction a term with the opposite sign to the friction will be added to the controller.

4.1.3 Inertia of the frame

To find the inertia of the frame a step response was used. Assuming the center of mass for the frame is in the center of the frame the dynamics of the frame can be found to be given by Equation 4.4 by Newton's second law. Here τ is the output torque of the DC-motor, while F_f is the torque generated by friction in the motor and the bearings found earlier.

$$J_f \ddot{\vartheta} = \tau + F_f \tag{4.4}$$

In Figure 4.1 the angular velocity of the frame is plotted in response to an input of 0.05[Nm]. The system was run four times until it reached an angular velocity of 40[rad/s]. As can be seen from the plot the angular velocity increases almost linearly.

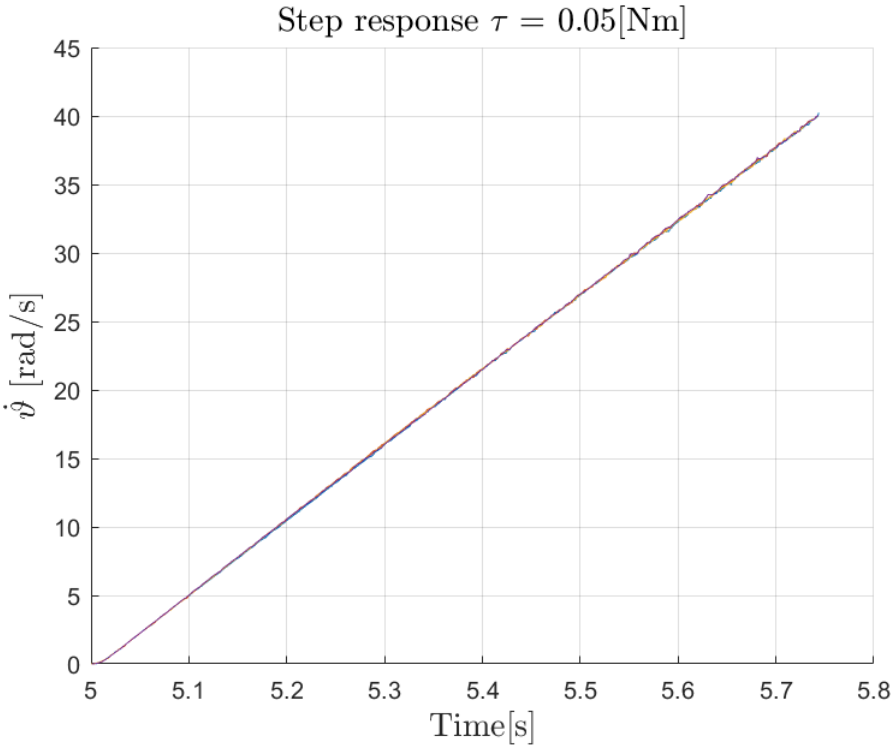

Figure 4.1: Step response of the frame with an input 0f 0.05[Nm]

The average acceleration was found to be 54[$\frac{\text{rad}}{\text{s}^2}$]. Using Equation 4.4 the inertia was then found to be given as:

$$J_f = 8.85 \times 10^{-4}[\text{Nm}] \tag{4.5}$$

4.1.4 Measurements of the states

To find the values for the four states for a given time value a camera and an encoder will be used. The encoder will give the value for ϑ while the camera will find the sphere's xy-position in the inertial frame. As φ is given in body frame it can be found as :

$$\varphi = \vartheta - \text{atan2}(x_s, y_s) \tag{4.6}$$

Where x_s and y_s is the measurement of the sphere's position. This formula can easily be observed from Figure 2.1.
The time derivatives of the states are found by using a numerical differentiation scheme.

4.2 Experimental results

The previously derived controller with the found parameters of the Butterfly Robot was tested on a physical setup of the Butterfly Robot at NTNU. The code for the controller was written in Matlab, which was then transcribed to C++ with the help of Matlab Coder. For tracking the sphere, controlling the current in the DC-motor and reading the encoder C++ code written by Maksim Surov for [14] was used.
A video of the Butterfly Robot in action can be found at https://photos.google.com/share/AF1QipMviik3RWN3ou2166vBMF2LVt0r1V9DNqoTejouI-MYvOk70X7oNO8gTbeZYWldgg?key=X01fTF9iWVlmekFFZ3VYbHN2WV9kOHBOcGhFc3Nn

4.2.1 With Luenberger observer

In the following the system was run with a Luenberger observer. $\boldsymbol{L}$ was chosen as:

$$\boldsymbol{L} = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 60 & 0 \\ 0 & 0 & 60 \end{bmatrix} \tag{4.7}$$

For the LQR $\boldsymbol{Q}$ and R was choosen as:

$$\boldsymbol{Q} = \begin{bmatrix} 40 & 0 & 0 \\ 0 & 40 & 0 \\ 0 & 0 & 20 \end{bmatrix}, \quad R = 1 \tag{4.8}$$

These values were found to work well for the Butterfly Robot by tuning.
For calculating the approximation of the stabilizing solution for the periodic Riccati differential equation n was set to 200 while m was set to 40. These are close to the values found in [4] for another periodic time varying system and the solution did not seem to change much by increasing the values.

The controller was also only run if a new measurement of φ was obtained since the previous run. The nominal trajectory was found with the help of the VHC and the $\alpha\beta\gamma$-equation . An interpolation was used to describe the frame.

In Figure 4.3 the measured φ is plotted over a part of the run. From this plot it can be observed that continuous rolling of the sphere is achieved. It can also be observed that φ is monotonically increasing with time. Figure 4.2 shows the frames rotation ϑ and the virtual holonomic constraint $\Theta(\varphi)$. It can be observed that the controller is able make ϑ almost equal to Θ. Figure 4.4 shows the time derivative of ϑ and the time derivative of the VHC plotted against φ. The controller is also able to make these quite equal, note also that since the derivative of the VHC is dependent on $\dot{\varphi}$ the camera measurements introduce some noise into the desired value for $\dot{\vartheta}$.

Figure 4.5 shows the measurement of $\dot{\varphi}$ and the nominal value for $\dot{\varphi}$. Here it can be observed that the measured value and the desired value are quite far from each other in certain parts of the period, at most about one radian per second. A small jump in $\dot{\varphi}_\star$ at $\varphi = \pi$ can also be observed. This is due to the spline interpolation. It can also be noted that of ϑ, φ, $\dot{\vartheta}$ and $\dot{\varphi}$ $\dot{\varphi}$ is the one with the most noise.

The measured transverse coordinates are plotted in Figure 4.6 while the estimates from the Luenberger observer are plotted in Figure 4.7. From these plots it can be observed that the LQR is able to stabilize the system, though convergence of all the transverse coordinates to zero is not achieved. The observer is also able to remove most of the high frequency noise from the transverse coordinates. Both the measured and the estimated transverse coordinates have less noise then the ones obtained in [14].

In Figure 4.8 a scatter plot of the transverse coordinates over the entire run is shown. These are plotted against the 2π modulo of φ over the entire run. Here it can be observed that the

deviation from zero of the transverse coordinates does not seem to be random. In these plots there is a total of 52 complete rotations of the frame, that is rotations by 2π radians.

The constraint forces given in Equation 2.33 are plotted in Figure 4.9. Here it can be observed that the force corresponding to the normal force is always larger than zero while the force corresponding to the friction is somewhere between 0.01 and minus 0.01. These forces were calculated in Simulink with data from the run.

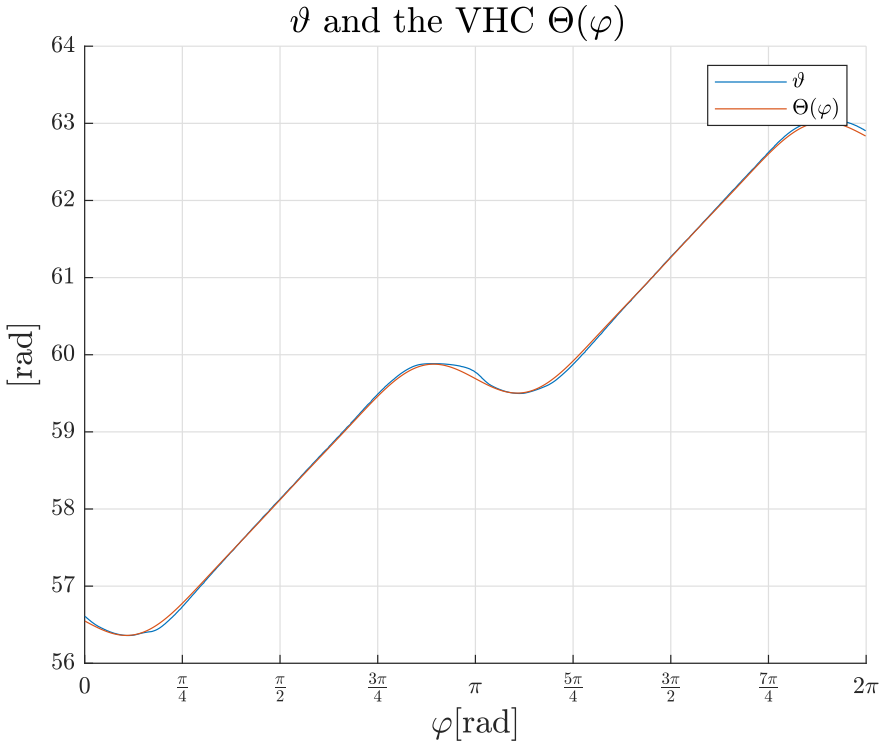

Figure 4.2: ϑ and Θ as a function of φ over 2 periods

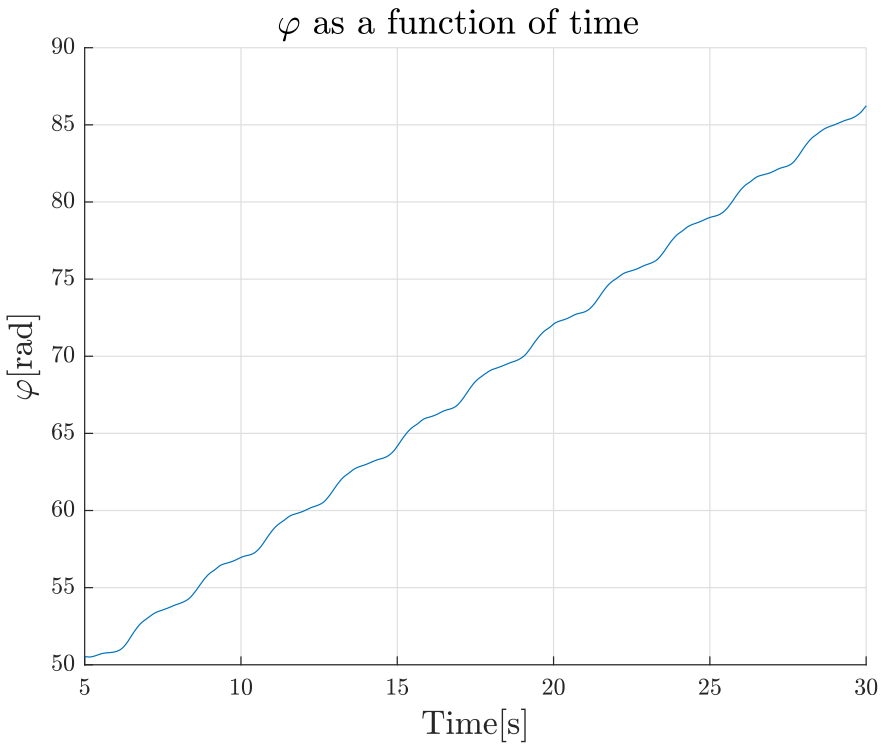

Figure 4.3: φ as a function of time

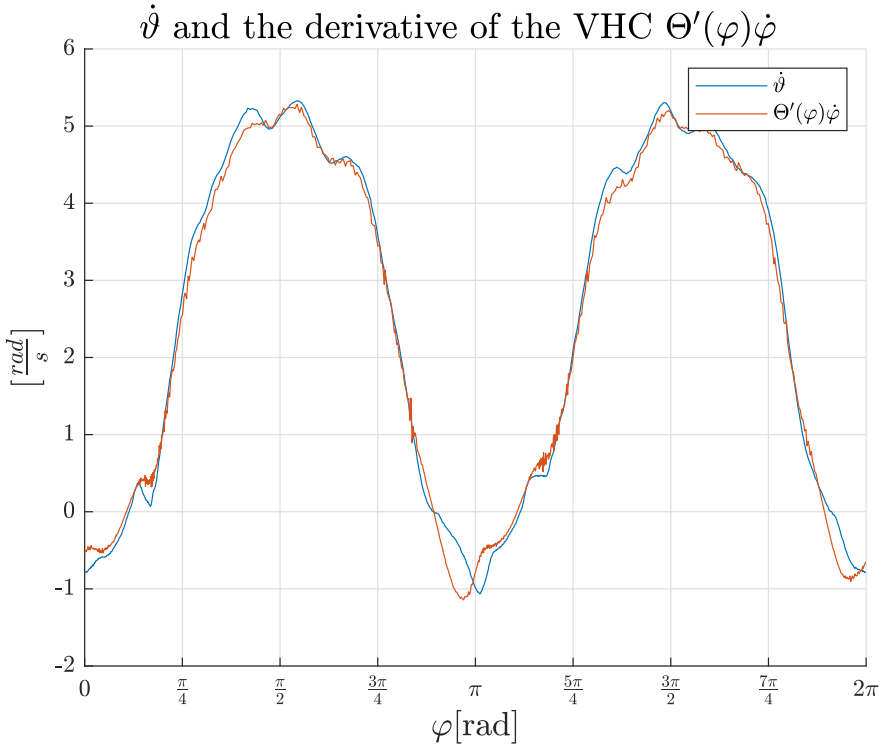

Figure 4.4: $\dot{\vartheta}$ and $\Theta'(\varphi)\dot{\varphi}$ over 2 periods of the system

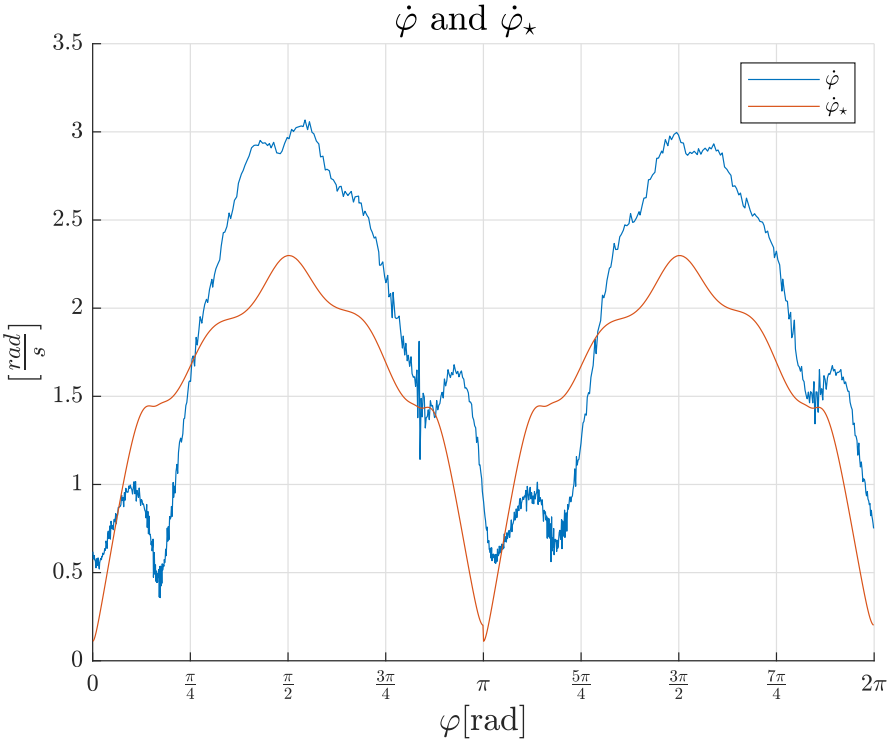

Figure 4.5: $\dot{\varphi}$ and $\dot{\varphi}_\star$ over 2 periods

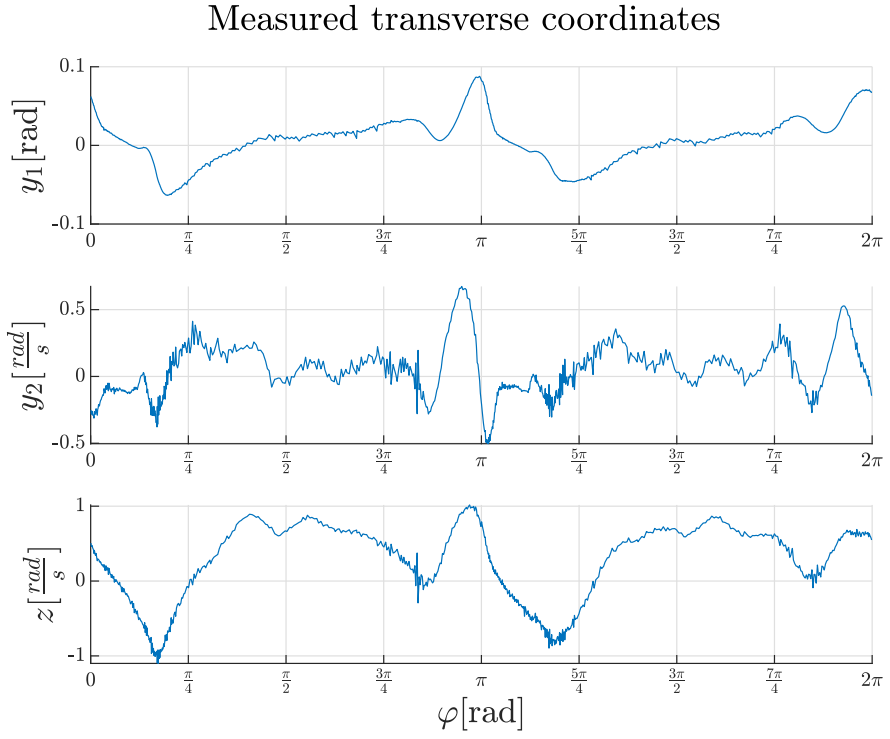

Figure 4.6: Measured transverse coordinates over 2 periods

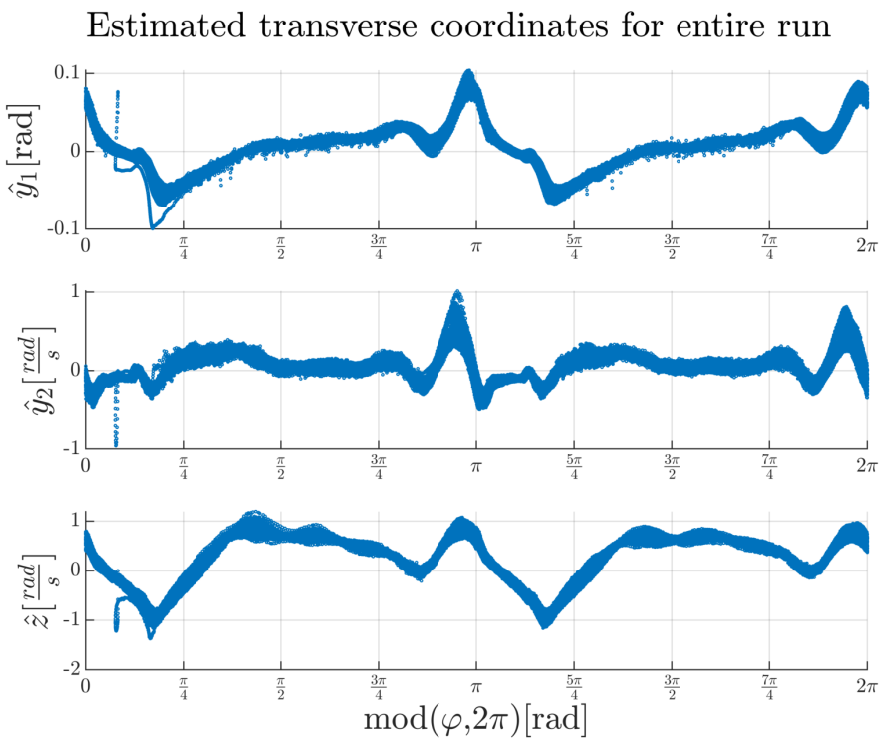

Figure 4.8: Transverse coordinates for the entire run as functions of 2π modulo of φ

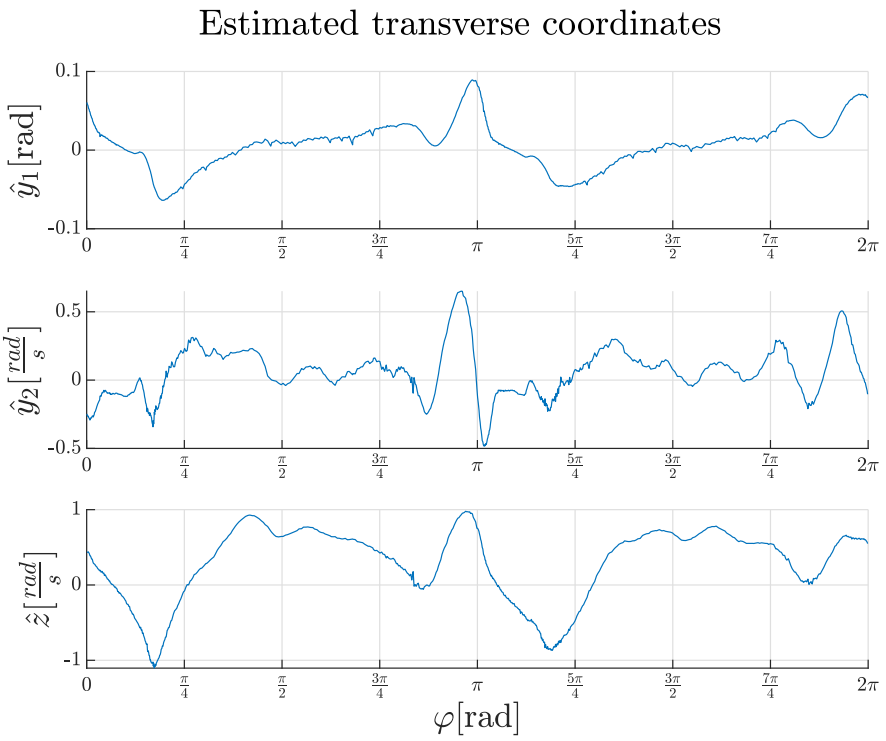

Figure 4.7: Estimated transverse coordinates over 2 periods

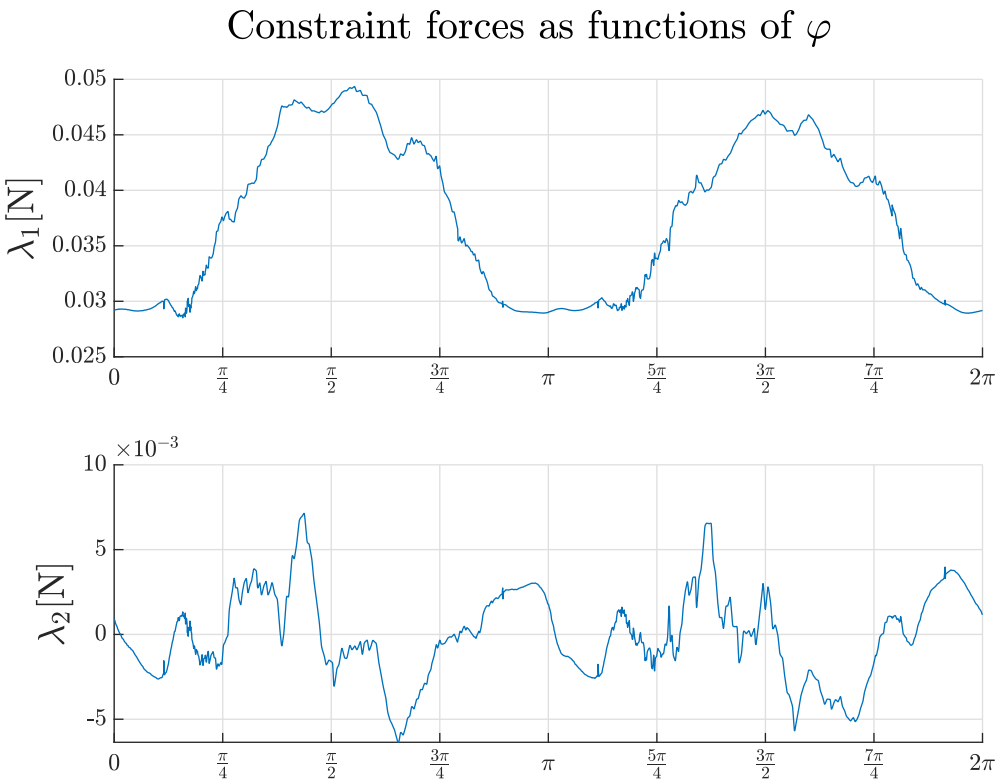

Figure 4.9: Calculated constraint forces from measurements

4.2.2 Effects of measurement rate

Now the system was run with no limitation on how often the control loop could be run, measurements of the states were also done as fast as possible. The resulting transverse coordinates for two periods is shown in Figure 4.10.
Here it can be observed that the first transverse coordinate has a lot more noise than earlier, the second transverse coordinate has a bit more noise while the third has about the same as the previous run. The shape of the noise in y_1 is similar to the noise that can be seen in the transverse coordinates in [14].

This made the system less robust, and as a result the system was only able to complete five rotations.

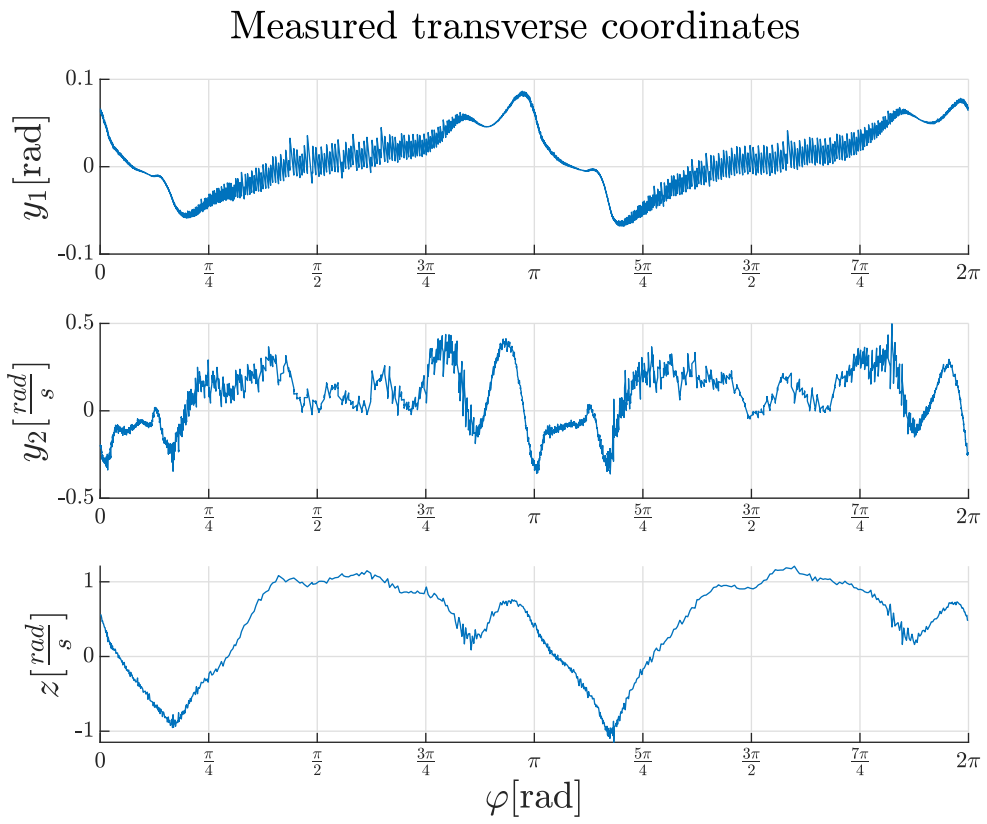

Figure 4.10: Measured transverse coordinates free running controller

4.3 Model compared with the physical robot

The measurements obtained from the run with the Luenberger observer were now used as an input to the dynamical model given in Equation 2.37. The resulting second derivatives were then integrated. This was done to see if the model was able to produce the expected results. In Figure 4.11 the results of inserting the measurements are plotted. In this plot the measurements of the two derivatives are shown in blue while the integrated dynamics are shown in red. The plot shows four runs of the system with different start times, the initial conditions of the integrators were set to the values of $\dot{\vartheta}$ and $\dot{\varphi}$ at the start time. This is done to show how fast the model diverges from the measurements depending on the initial conditions.

Only the first derivatives of ϑ and φ are considered, not ϑ and φ. This is because ϑ and φ will drift away from the measurements very fast as a result of the double integration.

From the plots it can be observed that the model gives a similar response to the measured response when $\dot{\vartheta}$ and $\dot{\varphi}$ is increasing. When they are not increasing the model and the measured value diverge quite quickly, especially around $\varphi = n\pi$ where n is an integer.

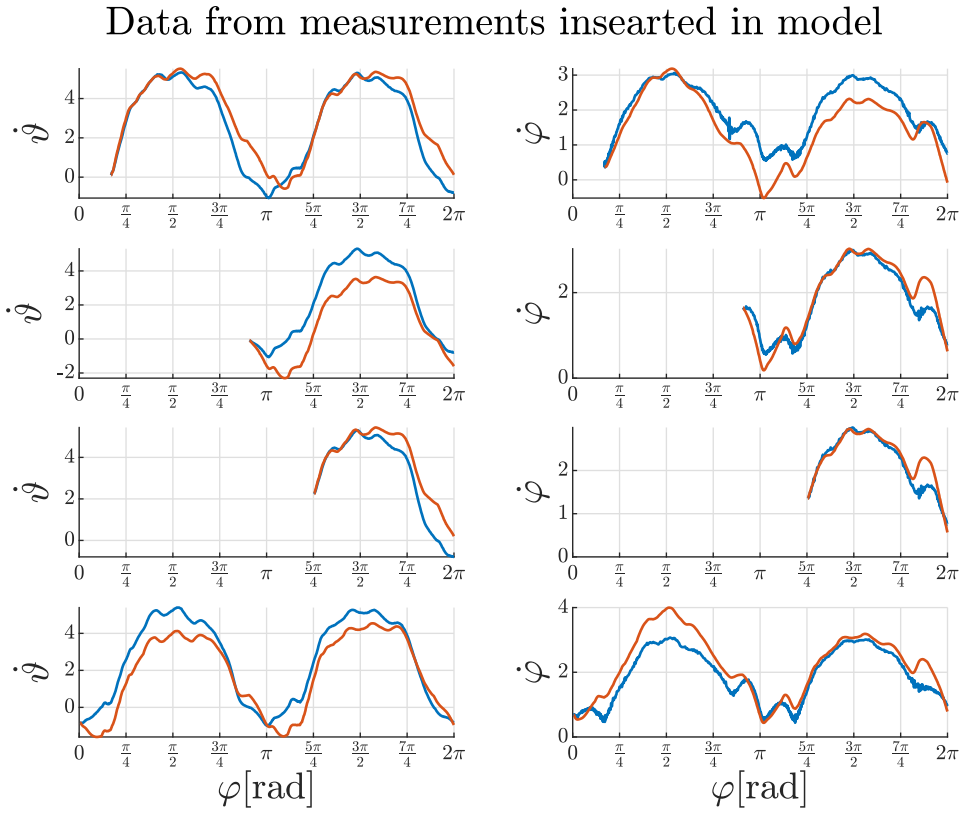

Figure 4.11: Measurements from run with observer inserted in the dynamical model. Blue line represents the measurements while the red line is the calculated values.

Chapter 5

Discussion, further work and conclusion

5.1 Discussion

The results obtained in the modeling and experiments will now be discussed.

5.1.1 Experiment on the Butterfly Robot

As was shown in the results the controller is able to achieve the goal of continuous rotation. The observer is also able to reduce the amount of noise in the estimated transverse coordinates compared to the measured transverse coordinates, particularly in y_2 and z as these had the most noise. This could probably have been improved by better tuning of the Luenberger observer or switching to a more sophisticated observer.

The transverse coordinates do not converge to zero as time goes on, though they seem to have a similar amount of error for a given φ independent of the time. This is believed to be because of the following reasons:

- Differences between the description of the frame and the actual frame:

  At a little before $\varphi = \frac{\pi}{4} + n$, where n is a integer, a deviation can also be observed in all of the transverse coordinates. Comparing this with the plots for the frame given in Figure 3.1 it can be seen that this the same area as the approximations of the derivatives of the frame deviate the most from the interpolation. So even though this run was done with the spline interpolation it is reasonable to assume that in this area the interpolation deviate the most from the actual frame.

- Large change in the nominal value for $\dot{\varphi}$:

  From Figure 4.8 it can be seen that the transverse coordinates have the most deviation from zero at φ approximately equal to a multiple of π. The velocity profile that the sphere was trying to follow has quite a large change around these points, this can be observed from the phase portraits in subsection 3.1.6. A trajectory with a smaller change would probably have been easier for the sphere to follow.

- Non-periodicity of the spline interpolation

A small jump can be observed in the nominal trajectory for $\dot{\varphi}$ at $\varphi = \pi$ in Figure 4.5. This is probably an artifact of the interpolation of the frame not being perfectly π periodic. A similar problem can be observed in the phase portrait associated with the spline interpolation, Figure 3.5. Compared with the other two it is not perfectly symmetrical about $\varphi = \frac{\pi}{2}$.

- Center of mass for the frame not the center of the frame:

  From the plot showing the transverse coordinates over the entire run against φ between 0 and 2π, Figure 4.8, a small deviation between the part of the plot from $\varphi \in \{0, \pi\}$ and $\varphi \in \{\pi, 2\pi\}$. It seems that the the transverse coordinates have a larger error in the peaks for the first part than in the second part. This may imply that the assumption that the center of mass for the frame is in the center of the frame is wrong.

- Slip of the sphere

  The plot of the dynamic model run with data from the measurements shows that the model has problems describing the system when the sphere and frame is decelerating. In these parts measured values and the calculated values are further apart than when the sphere and frame are accelerating. From Figure 4.9 it can be seen that these areas also need the highest friction force between the sphere and the frame. A possible explanation for this discrepancy between the measured values and the calculated values is therefore that the sphere did not have pure rolling when it was decelerating.

5.1.2 Different representations of the frame

In subsection 3.1.5 three different descriptions for the frame were given. These are based on data of the frame, so none of the three descriptions describe the frame perfectly. In addition the second and third derivative of the frame is probably non-continuous for the real frame because of production errors. As a result this description will always cause a problem for the controller.

Comparing the phase portraits shows that the three descriptions give similar dynamics for the reduced dynamics. They seem to have the same number of equilibria and these equilibria seem to have the same characteristics. That is, a center remains a center and a saddle remains a saddle. Though the two centers seem to move a bit between the phase portraits. This shows that the description chosen for the frame has consequences for which trajectories can be found for the Butterfly Robot. It also shows that these trajectories are quite sensitive to small differences in the description of the frame.

5.1.3 Virtual holonomic constraint

The virtual holonomic constraint used in this thesis was taken as arctan of a sum of four sin terms. This number of terms was found to give a satisfactory result in the optimization problem. Less terms (that is one or two) resulted in a VHC that was not able to create the desired behaviours in the phase planes for the $\alpha\beta\gamma$-equation , while adding additional terms did not contribute much and also made finding a feasible solution harder. Four terms was therefore chosen as a good middle ground. It can also be observed from the values found in Equation 3.25 that the magnitude of the terms decreases by about one order of magnitude for each term after the second.

The optimization problem is quite nonlinear and is not convex. As a result of this for some placements of the equilibria no solution was possible to find.

5.1.4 Noise as a result of low measurement frequency of the motion generator

As the LQR and the system dynamics was calculated in terms of φ rather than time a problem occurred when ϑ and $\dot{\vartheta}$ was measured with a higher frequency than φ. This because ϑ changed at a higher frequency than $\Theta(\varphi)$, and $\dot{\vartheta}$ changed at a higher frequency than $\Theta'(\varphi)\dot{\varphi}$. ϑ was measured about four times as often as φ. z was not affected by this as it is only dependent on the measurement of φ and $\dot{\varphi}$.

It seems that this had more of an impact on y_1, that is the transverse coordinate dependent on ϑ, than y_2. The noise in y_1 also seems to appear mainly when ϑ is increasing almost linearly, see Figure 4.2, that is φ between $\frac{\pi}{4}$ and $\frac{3\pi}{4}$. This is also the part of the trajectory where ϑ has the least deviation from Θ and increasing the fastest.

Since ϑ has a higher measurement frequency than φ and the controller was ran faster than the measurement frequency of φ this caused the value of Θ to stay the same for a couple of iterations of the controller while ϑ increased. This caused the transverse coordinate to increase (or decrease depending on if ϑ was smaller or larger than Θ for the first measurement) steadily until a new measurement of φ was obtained, at this point the transverse coordinate jumped to the actual value. This gives this saw-tooth-like behaviour in the transverse coordinates which can be observed in Figure 4.10. This shows the importance of not measuring other states faster than the motion generator when using this type of controller. It could possibly have been resolved by using an estimator for φ rather than the measurement of φ.

5.1.5 Usability of Matlab Coder

To generate the C++ code that was used to run the Butterfly Robot Matlab Coder was used. This is a Matlab add-on which turns a Matlab function into valid C++ code. This application was used rather than writing the C++ code myself as it made prototyping the controller faster and gave the possibility to run the same controller for the simulation as the actual system. This turned out to work well, integration of the C++ code from Matlab Coder into the already existing code for the Butterfly Robot went smoothly. The code also ran fast enough to not become a bottleneck in the system. The downside with doing it like this is that the resulting C++ code is quite hard to understand for a human, making it a bit difficult to make changes to the code after the generation.

5.1.6 Kalman filter

In subsection 3.3.2 a setup for a Kalman filter for the Butterfly Robot was described. This turned out to work very poorly in simulation of the Butterfly Robot, it was therefore not tested on the real robot. The reason for this is believed to be because of how the noise enters the measurement of the transverse coordinates. The measurement of the second transverse coordinate gets a term dependent on the derivative of the VHC times the noise in the measurement of $\dot{\varphi}$. As a result the noise becomes linked to the states of the system through φ.

This may cause the separation principle to fall apart. The assumption that the measurement of φ has no noise is also not true in the real system.

5.2 Further work

In this work a very simple model for the friction in the DC-motor and bearings was used, a constant for each direction of rotation. A more sophisticated model for this would probably improve the controller, especially if it is able to describe the change in friction when the frame changes direction of rotation. Taking the model for the DC-motor into account rather than looking at it as part of the inertia of the frame could also have improved the model.

Rather than the Luenberger observer a Kalman filter could have been used. This would have needed further study of the noise that appears in the transverse coordinates. Some way to incorporate the noise from φ would also be needed. Maybe an extended Kalman filter could work.

Using φ as the motion generator is causing some problems, the measurement rate is lower than ϑ and it has more noise. If another variable was found that is monotonically increasing on the trajectory this could have improved the controller. Splitting the trajectory into different parts and using a different motion generator for each part could also have been a possibility. Another possibility is to search for a linear combination of φ and ϑ that is monotonically increasing on the trajectory. That is a motion generator on the following form, where δ and ϵ are some gains, possibly time varying.

$$\psi = \delta\varphi + \epsilon\vartheta \tag{5.1}$$

If the assumption that the sphere only has angular velocity about the z-axis is dropped the resulting system will have a higher order of underactuation. A model that describes the resulting system would give further insight into the Butterfly Robot and the control problem.

Further investigation of how the phase portrait changes as a result of the frame is also of interest. Especially finding out if at some point there will be a bifurcation, that is a sudden change in the characteristics of the equilibria, for the equilibria of the $\alpha\beta\gamma$-equation.

5.3 Conclusion

In this thesis a benchmark example for performing non-prehensile manipulation has been considered, the Butterfly Robot. The interest in this robot lies in that valid control strategies for this robot can further our understanding of underactuated systems, and potentially create human-like behaviour in robotic systems.

With the help of Lagrange's equations of motion two models were derived for the Butterfly Robot, one with four degrees of freedom and one with two degrees of freedom. The 4-DoF model was used to derive two constraint forces representing the normal force and the friction force. The 2-DoF model was used to derive a control strategy consisting of a virtual holonomic constraint and a partial feedback linearization of a set of transverse coordinates. Three different descriptions of the frame were considered for the Butterfly Robot, and the impact these had on the dynamics of the $\alpha\beta\gamma$-equation was studied. The linearized system of transverse coordinates was then used to design a φ varying LQR and a Luenberger observer. This controller was then tested on a physical realization of the Butterfly Robot at NTNU. It was found that the controller was able to stabilize the found trajectories.

The main contributions of this thesis is the following:

- A new virtual holonmic constraint for the Butterfly Robot.
- Analysis of the equilibria of the $\alpha\beta\gamma$-equation as a result of different descriptions of the frame and showing that the $\alpha\beta\gamma$-equation is quite sensitive to small variations in the frame.
- Confirming that the work done by Maksim Surov et al in [14] is repeatable with a different VHC.

Bibliography

[1] Robert A Adams. *Calculus: A Complete Course*. 2017. ISBN: 9788578110796.

[2] S. Barnett. *Linear system theory and design. By C.-T. Chen*. 1986. DOI: 10.1016/0005-1098(86)90039-7.

[3] Olav Egeland and Jan Gravdahl. *Modeling and Simulation for Automatic Control*. May 2002.

[4] Sergei V Gusev, Anton S Shiriaev and Leonid B Freidovich. 'SDP-based approximation of stabilising solutions for periodic matrix Riccati differential equations'. In: *International Journal of Control* 89.7 (2016), pp. 1396–1405. DOI: 10.1080/00207179.2015.1131850. URL: https://doi.org/10.1080/00207179.2015.1131850.

[5] H K Khalil. *Nonlinear Systems, Third Edition*. 2002. ISBN: 0130673897. DOI: 10.1016/j.physa.2006.08.011.

[6] Johan Löfberg. 'YALMIP: A toolbox for modeling and optimization in MATLAB'. In: *Proceedings of the IEEE International Symposium on Computer-Aided Control System Design*. 2004. DOI: 10.1109/cacsd.2004.1393890.

[7] Kevin M. Lynch et al. 'The roles of shape and motion in dynamic manipulation: The butterfly example'. In: *Proceedings - IEEE International Conference on Robotics and Automation*. 1998. ISBN: 078034300X. DOI: 10.1109/ROBOT.1998.680600.

[8] Matlab. *Matlab Curve Fitting Toolbox*. 2014.

[9] Patrick Y. C. Hwang Robert Grover Brown. *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises*. 2012.

[10] A. Shiriaev et al. 'Periodic motion planning for virtually constrained (Hybrid) mechanical systems'. In: *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, CDC-ECC '05*. 2005. ISBN: 0780395689. DOI: 10.1109/CDC.2005.1582793.

[11] Anton Shiriaev, John W. Perram and Carlos Canudas-de-Wit. 'Constructive tool for orbital stabilization of underactuated nonlinear systems: Virtual constraints approach'. In: *IEEE Transactions on Automatic Control* 50.8 (2005). ISSN: 00189286. DOI: 10.1109/TAC.2005.852568.

[12] Anton S. Shiriaev, Leonid B. Freidovich and Sergei V. Gusev. 'Transverse linearization for controlled mechanical systems with several passive degrees of freedom'. In: *IEEE Transactions on Automatic Control* (2010). ISSN: 00189286. DOI: 10.1109/TAC.2010.2042000.

[13] Mark W. Spong, Seth Hutchinson and M. Vidyasagar. *Robot modeling and control*. 2006. DOI: 10.1109/MCS.2006.252815.

[14] Maksim Surov et al. 'Case study in non-prehensile manipulation: Planning and orbital stabilization of one-directional rollings for the 'Butterfly' robot'. In: *Proceedings - IEEE International Conference on Robotics and Automation*. 2015. DOI: 10.1109/ICRA.2015.7139385.

[15] K. C. Toh, M. J. Todd and R. H. Tütüncü. 'SDPT3 - a MATLAB software package for semidefinite programming, version 1.3'. In: *Optimization Methods and Software* (1999). ISSN: 10556788. DOI: 10.1080/10556789908805762.

Appendix A

Appendix

A.1 Code

All of the code used in this thesis can be found at: https://github.com/GustavOften/ttk4900_master_thesis.

A.2 Effective rolling radius

In Figure A.1 an illustration of a sphere rolling on two discs is shown. The effective rolling radius for this sphere will be given by the value R_d in the drawing. From the Pythagorean theorem it can be seen that this radius will be given as:

$$R_d = \sqrt{{R_s}^2 - {R_f}^2} \tag{A.1}$$

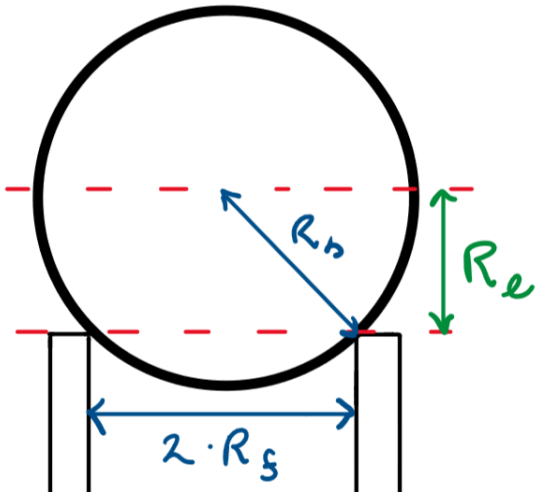

Figure A.1: Illustration for effective rolling radius

A.3 Derivation 4-DoF dynamics

The kinetic energy of the system is given by:

$$K(\vec{q},\dot{\vec{q}}) = \frac{1}{2}\dot{\vec{q}}\begin{bmatrix} J_f + J_s + m_d\vec{\rho}^\top\vec{\rho} & m_d s'\vec{\rho}\times\vec{\tau} & J_s & m_d\vec{\rho}\times\vec{n}_+ \\ m_d s'\vec{\rho}\times\vec{\tau} & m_d s'^2 & 0 & 0 \\ J_s & 0 & J_s & 0 \\ m_d\vec{\rho}\times\vec{n}_+ & 0 & 0 & m_d \end{bmatrix}\dot{\vec{q}} \tag{A.2}$$

The mass-inertia matrix $\boldsymbol{M}(\vec{q})$ can be found from the identity:

$$K(\vec{q},\dot{\vec{q}}) = \frac{1}{2}\dot{\vec{q}}^\top\boldsymbol{M}(\vec{q})\dot{\vec{q}} \tag{A.3}$$

This gives that it is given as:

$$\boldsymbol{M}(\vec{q}) = \begin{bmatrix} J_f + J_s + m_d\vec{\rho}^\top\vec{\rho} & m_d s'\vec{\rho}\times\vec{\tau} & J_s & m_d\vec{\rho}\times\vec{n}_+ \\ m_d s'\vec{\rho}\times\vec{\tau} & m_d s'^2 & 0 & 0 \\ J_s & 0 & J_s & 0 \\ m_d\vec{\rho}\times\vec{n}_+ & 0 & 0 & m_d \end{bmatrix} \tag{A.4}$$

Using Christoffel symbols the centrifugal and Coriolis matrix can be found as:

$$c_{kj} = \sum_{i=1}^{4} c_{ijk}(q)\dot{q}_i \tag{A.5}$$

Where c_{ijk} is given as:

$$c_{ijk} = \frac{1}{2}\left(\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k}\right) \tag{A.6}$$

Some matrices for the derivatives of $\boldsymbol{M}(q)$ will help with this:

$$\frac{\partial \boldsymbol{M}}{\partial \vartheta} = \frac{\partial \boldsymbol{M}}{\partial \psi} = \boldsymbol{0} \tag{A.7}$$

$$\frac{\partial \boldsymbol{M}}{\partial \varphi} = m_d\begin{bmatrix} 2\vec{\rho}^\top\vec{\tau}s' & s''\vec{\rho}\times\vec{\tau} + \vec{\rho}\times\vec{\kappa}s' & 0 & s' - \|\vec{\kappa}\|\,\vec{\rho}\times\vec{\tau}s' \\ s''\vec{\rho}\times\vec{\tau} + \vec{\rho}\times\vec{\kappa}s' & 2s''s' & 0 & 0 \\ 0 & 0 & 0 & 0 \\ s' - \|\vec{\kappa}\|\,\vec{\rho}\times\vec{\tau}s' & 0 & 0 & 0 \end{bmatrix} \tag{A.8}$$

$$\frac{\partial \boldsymbol{M}}{\partial w} = m_d\begin{bmatrix} 2\vec{\rho}^\top\vec{n}_+ & s'\vec{\rho}\times\vec{n}_+ & 0 & 0 \\ s'\vec{\rho}\times\vec{n}_+ & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.9}$$

This gives that the Christoffel symbols can be written as four matrices:

$$\boldsymbol{C}_{ij1} = m_d\begin{bmatrix} 0 & \vec{\rho}^\top\vec{\tau}s' & 0 & \vec{\rho}^\top\vec{n}_+ \\ \vec{\rho}^\top\vec{\tau}s' & s''\vec{\rho}\times\vec{\tau} + \vec{\rho}\times\vec{\kappa}s' & 0 & \frac{1}{2}(s' - \|\vec{\kappa}\|\,\vec{\rho}\times\vec{\tau}s') \\ 0 & 0 & 0 & 0 \\ \vec{\rho}^\top\vec{n}_+ & \frac{1}{2}(s' - \|\vec{\kappa}\|\,\vec{\rho}\times\vec{\tau}s') & 0 & 0 \end{bmatrix} \tag{A.10}$$

$$\boldsymbol{C}_{ij2} = m_d\begin{bmatrix} \vec{\rho}^\top\vec{\tau}s' & 0 & 0 & 0 \\ 0 & s''s' & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.11}$$

$$\boldsymbol{C}_{ij3} = \boldsymbol{0} \tag{A.12}$$

$$\boldsymbol{C}_{ij4} = m_d\begin{bmatrix} -\vec{\rho}^\top\vec{\tau}s' & \frac{1}{2}(s' - \|\vec{\kappa}\|\,\vec{\rho}\times\vec{\tau}s') & 0 & 0 \\ \frac{1}{2}(s' - \|\vec{\kappa}\|\,\vec{\rho}\times\vec{\tau}s') & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.13}$$

This gives that the $\boldsymbol{C}$ is given as:

$$\begin{aligned}
c_{11} &= \vec{\rho}^{\top}\vec{\tau}s'\dot{\varphi} + \vec{\rho}^{\top}\vec{n}_{+}\dot{w} \\
c_{12} &= \vec{\rho}^{\top}\vec{\tau}s'\dot{\vartheta} + (s''\vec{\rho}\times\vec{\tau} + \vec{\rho}\times\vec{\kappa}s')\dot{\varphi} + \frac{1}{2}(s' - \|\vec{\kappa}\|\,\vec{\rho}\times\vec{\tau}s')\dot{w} \\
c_{13} &= 0 \\
c_{14} &= \vec{\rho}^{\top}\vec{n}_{+}\dot{\vartheta} + \frac{1}{2}(s' - \|\vec{\kappa}\|\,\vec{\rho}\times\vec{\tau}s')\dot{\varphi} \\
c_{21} &= \vec{\rho}^{\top}\vec{\tau}s'\dot{\vartheta} \\
c_{22} &= s''s'\dot{\varphi} \\
c_{23} &= 0 \\
c_{24} &= 0 \\
c_{31} &= c_{32} = c_{33} = c_{34} = 0 \\
c_{41} &= -\vec{\rho}^{\top}\vec{\tau}s'\dot{\vartheta} + \frac{1}{2}(s' - \|\vec{\kappa}\|\,\vec{\rho}\times\vec{\tau}s')\dot{\varphi} \\
c_{42} &= \frac{1}{2}(s' - \|\vec{\kappa}\|\,\vec{\rho}\times\vec{\tau}s')\dot{\vartheta} \\
c_{43} &= c_{44} = 0
\end{aligned} \tag{A.14}$$

$$\boldsymbol{C}(\vec{q},\dot{\vec{q}}) = m_d \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \\ c_{41} & c_{42} & c_{43} & c_{44} \end{bmatrix} \tag{A.15}$$

To find the gravity matrix the potential energy, $P(\vec{q})$, is differentiated with respect to the states. This gives:

$$G(q) = gm_d \begin{bmatrix} [\boldsymbol{R}'(\vartheta)\vec{\rho}(\varphi,w)]_2 \\ [\boldsymbol{R}(\vartheta)\vec{\tau}(\varphi)s']_2 \\ 0 \\ [\boldsymbol{R}(\vartheta)\vec{n}_{+}(\varphi)]_2 \end{bmatrix} \tag{A.16}$$

A.4 Derivation of the 2-DoF dynamics

From the kinetic energy of the system it can be observed that $\boldsymbol{M}$ will be given as:

$$\boldsymbol{M}(\vec{q}) = m_d \begin{bmatrix} \frac{J_f}{m_d} + \frac{J_s}{m_d} + \vec{\rho}^{\top}\vec{\rho} & s'\vec{\rho}^{\top}\vec{\tau} - s'_f\frac{J_s}{m_d R_d} \\ s'\vec{\rho}^{\top}\vec{\tau} - s'_f\frac{J_s}{m_d R_d} & s'^2_f\frac{J_s}{m_d R_d^2} \end{bmatrix} \tag{A.17}$$

The derivatives of $\boldsymbol{M}$ with respect to the states is then given as:

$$\frac{\partial \boldsymbol{M}}{\partial \vartheta} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{A.18}$$

$$\frac{\partial \boldsymbol{M}}{\partial \varphi} = m_d \begin{bmatrix} 2\vec{\rho}^{\top}\vec{\tau}s' & (s''\vec{\rho}\times\vec{\tau} - s''_f\frac{J_s}{R_b m_d} + s'(\frac{\partial\vec{\rho}}{\partial\varphi}\times\vec{\tau} + \vec{\rho}\times\frac{\partial\vec{\tau}}{\partial\varphi})) \\ s''\vec{\rho}\times\vec{\tau} - s''_f\frac{J_s}{R_b m_d} + s'^2\vec{\rho}\times\vec{\kappa} & 2(s's'' + s'_f s''_f\frac{J_s}{R_b^2 m_d}) \end{bmatrix} \tag{A.19}$$

The Christoffel symbols can then be found to be:

$$\begin{aligned}
\boldsymbol{C}_{ij1} &= \begin{bmatrix} 0 & m_b\vec{\rho}^{\top} \\ m_b\vec{\rho}^{\top} & m_b(s''\vec{\rho}\times\vec{\tau} - s''_f\frac{J_s}{R_b m_b} + s'^2\vec{\rho}\times\vec{\kappa}) \end{bmatrix} \\
\boldsymbol{C}_{ij2} &= \begin{bmatrix} -m_b\vec{\rho}^{\top}\vec{\tau}s' & 0 \\ 0 & m_b(s's'' + s'_f s''_f\frac{J_s}{R_b^2 m_b}) \end{bmatrix}
\end{aligned} \tag{A.20}$$

Which gives the following expressions for the elements of the centrifugal and Coriolis matrix:

$$\boldsymbol{C}(\vec{q},\dot{\vec{q}}) = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \tag{A.21}$$

$$\begin{aligned} c_{11} &= m_b \vec{\rho}\cdot\vec{\tau}s'\dot{\varphi} \\ c_{12} &= m_b(\vec{\rho}\cdot\vec{\tau}s'\dot{\vartheta} + (s''\vec{\rho}\times\vec{\tau} - s_f''\frac{J_s}{R_b m_b} + s'^2\vec{\rho}\times\vec{\kappa})) \\ c_{21} &= -m_b\vec{\rho}\cdot\vec{\tau}s'\dot{\vartheta} \\ c_{22} &= m_b(s's'' + s_f's_f''\frac{J_s}{R_b^2 m_b})\dot{\varphi} \end{aligned} \tag{A.22}$$

Again, differentiating the potential energy with respect to the states gives the gravity matrix $\boldsymbol{G}$:

$$\boldsymbol{G}(q) = g m_d \begin{bmatrix} [\boldsymbol{R}'(\vartheta)\vec{\rho}]_2 \\ [\boldsymbol{R}(\vartheta)\vec{\tau}s']_2 \end{bmatrix} \tag{A.23}$$

A.5 $\alpha\beta\gamma$-equation

Assuming a system on the form given in Equation 2.37 with at least one actuator. And a motion generator such that $n-1$ of the states can be written as a function of the last state:

$$\vec{q} = \begin{bmatrix} q_1 \\ q_2 \\ ... \\ q_n \end{bmatrix} = \begin{bmatrix} \Theta_1(q_n) \\ \Theta_2(q_n) \\ ... \\ \Theta_n(q_n) \end{bmatrix} = \vec{\Theta} \tag{A.24}$$

This gives that:

$$\dot{\vec{q}} = \dot{\vec{\Theta}} = \vec{\Theta}'\dot{q}_n \tag{A.25}$$

$$\ddot{\vec{q}} = \ddot{\vec{\Theta}} = \vec{\Theta}''\dot{q}_n^2 + \vec{\Theta}'\ddot{q}_n \tag{A.26}$$

Multiplying both sides of the robot model with a row matrix with a one on the kth row, assuming state number k is not actuated gives the kth $\alpha\beta\gamma$-equation . Here the Christoffel symbols has been used to describe the coriolis and centrifugal matrix.

$$\alpha_k = \sum_{j=1}^{n} m_{kj}\Theta_j' \tag{A.27}$$

$$\beta_k = \left(\sum_{j=1}^{n} m_{kj}\Theta_j'' + \sum_{j=1}^{n}\left(\sum_{i=1}^{n} c_{ijk}\Theta_i'\right)\Theta_j'\right) \tag{A.28}$$

$$\gamma_k = \boldsymbol{G}(\vec{\Theta})_k \tag{A.29}$$

A.6 Derivation of feedforward

In this derivation all of the matrices are evaluated on the nominal trajectory. Using the fact that the $\alpha\beta\gamma$-equation is zero on the trajectory $\ddot{\varphi}_\star$ can be written as:

$$\ddot{\varphi}_\star = -\frac{\beta(\varphi)\dot{\varphi}_\star^2 + \gamma(\varphi)}{\alpha(\varphi)} \tag{A.30}$$

Inserting for α, β and γ given in Equation 3.7 gives:

$$\ddot{\varphi}_\star = -\frac{(M_{21}\Theta'' + c_{112}\Theta'^2 + c_{122}\Theta' + c_{212}\Theta' + c_{222})\dot{\varphi}_\star^2 + G_2}{M_{21}\Theta' + M_{22}} \tag{A.31}$$

This can be written as:

$$\ddot{\varphi}_\star = -\frac{M_{21}\Theta''\dot{\varphi}_\star^2 + \begin{bmatrix}0 & 1\end{bmatrix}\boldsymbol{C}\begin{bmatrix}\Theta'\dot{\varphi}_\star \\ \dot{\varphi}_\star\end{bmatrix} + \begin{bmatrix}0 & 1\end{bmatrix}\boldsymbol{G}}{M_{21}\Theta' + M_{22}} \tag{A.32}$$

Which can be further simplified to:

$$\ddot{\varphi}_\star = -\frac{\begin{bmatrix}0 & 1\end{bmatrix}\boldsymbol{M}\begin{bmatrix}\Theta''\dot{\varphi}_\star^2 \\ 0\end{bmatrix} + \begin{bmatrix}0 & 1\end{bmatrix}\boldsymbol{C}\begin{bmatrix}\Theta'\dot{\varphi}_\star \\ \dot{\varphi}_\star\end{bmatrix} + \begin{bmatrix}0 & 1\end{bmatrix}\boldsymbol{G}}{\begin{bmatrix}\Theta' & 1\end{bmatrix}\boldsymbol{M}\begin{bmatrix}0 \\ 1\end{bmatrix}} \tag{A.33}$$

Inserting this expression for $\ddot{\varphi}_\star$ into Equation 3.4 gives

$$u_\star(\varphi) = k\left(\begin{bmatrix}k^{-1} & -1\end{bmatrix}\boldsymbol{M}\begin{bmatrix}\Theta''\dot{\varphi}_\star^2 \\ 0\end{bmatrix} + \begin{bmatrix}k^{-1} & -1\end{bmatrix}\boldsymbol{C}\begin{bmatrix}\Theta'\dot{\varphi}_\star \\ \dot{\varphi}_\star\end{bmatrix} + \begin{bmatrix}k^{-1} & -1\end{bmatrix}\boldsymbol{G}\right) \tag{A.34}$$

Where k is defined to be:

$$k = \frac{\begin{bmatrix}1 & 0\end{bmatrix}\boldsymbol{M}\begin{bmatrix}\Theta' & 1\end{bmatrix}}{\begin{bmatrix}\Theta' & 1\end{bmatrix}\boldsymbol{M}\begin{bmatrix}0 \\ 1\end{bmatrix}} \tag{A.35}$$

A.7 Derivative of $\dot{\varphi}_\star^2(\varphi)$ with respect to φ

$$\frac{\partial\dot{\varphi}_\star^2}{\partial\varphi} = \frac{\partial}{\partial\varphi}\left[\exp\left\{-2\int_{\varphi_0}^{\varphi}\frac{\beta(w)}{\alpha(w)}dw\right\}\left(\dot{\varphi}_{0\star}^2 - \int_{\varphi_0}^{\varphi}\frac{2\gamma(w)}{\alpha(w)}\exp\left\{2\int_{\varphi_0}^{w}\frac{\beta(v)}{\alpha(v)}dv\right\}dw\right)\right] \tag{A.36}$$

Using the following function will simplify the derivation a bit, here the fundamental theorem of calculus [1] has been used to compute the derivative.

$$\psi(a,b) = 2\int_a^b\frac{\beta(w)}{\alpha(w)}dw \tag{A.37}$$

$$\frac{\partial\psi(a,b)}{\partial b} = 2\frac{\beta(b)}{\alpha(b)} \tag{A.38}$$

This gives that $\dot{\varphi}_\star^2$ can be written as:

$$\frac{\partial\dot{\varphi}_\star^2}{\partial\varphi} = \frac{\partial}{\partial\varphi}\left[\exp\left\{-\psi(\varphi_0,\varphi)\right\}\left(\dot{\varphi}_{0\star}^2 - \int_{\varphi_0}^{\varphi}\frac{2\gamma(w)}{\alpha(w)}\exp\left\{\psi(\varphi_0,w)\right\}dw\right)\right] \tag{A.39}$$

Using Leibniz formula this can be written as:

$$\begin{aligned}\frac{\partial\dot{\varphi}_\star^2}{\partial\varphi} = \exp\left\{-\psi(\varphi_0,\varphi)\right\}\left(-\frac{\partial\psi(a,b)}{\partial b}\right)\left(\dot{\varphi}_{0\star}^2 - \int_{\varphi_0}^{\varphi}\frac{2\gamma(w)}{\alpha(w)}\exp\left\{\psi(\varphi_0,w)\right\}dw\right) + \\ \exp\left\{-\psi(\varphi_\star,\varphi)\right\}\frac{\partial}{\partial\varphi}\left[\left(\dot{\varphi}_{0\star}^2 - \int_{\varphi_0}^{\varphi}\frac{2\gamma(w)}{\alpha(w)}\exp\left\{\psi(\varphi_0,w)\right\}dw\right)\right]\end{aligned} \tag{A.40}$$

Here it can be observed that the first part of the sum is the expression for $\dot{\varphi}_\star^2$ times the derivative of ψ, thus:

$$\frac{\partial\dot{\varphi}_\star^2}{\partial\varphi} = -2\dot{\varphi}_\star^2\frac{\beta(\varphi)}{\alpha(\varphi)} + \exp\left\{-\psi(\varphi_0,\varphi)\right\}\frac{\partial}{\partial\varphi}\left[\left(\dot{\varphi}_{0\star}^2 - \int_{\varphi_0}^{\varphi}\frac{2\gamma(w)}{\alpha(w)}\exp\left\{\psi(\varphi_0,w)\right\}dw\right)\right] \tag{A.41}$$

As $\dot{\varphi}_{0\star}^2$ is constant and the derivative of the integral can be calculated by the fundamental theorem of calculus the derivative of the last part becomes:

$$\frac{\partial}{\partial\varphi}\left[\left(\dot{\varphi}_{0\star}^2 - \int_{\varphi_0}^{\varphi}\frac{2\gamma(w)}{\alpha(w)}\exp\left\{\psi(\varphi_0,w)\right\}dw\right)\right] = -\frac{2\gamma(w)}{\alpha(w)}\exp\left\{\psi(\varphi_0,\varphi)\right\} \tag{A.42}$$

This gives that the last expression in the sum can be written as:

$$-\exp\left\{-\psi(\varphi_0,\varphi)\right\}\frac{2\gamma(\varphi)}{\alpha(\varphi)}\exp\left\{\psi(\varphi_0,\varphi)\right\} = -\exp\left\{-\psi(\varphi_0,\varphi)+\psi(\varphi_0,\varphi)\right\}\frac{2\gamma(\varphi)}{\alpha(\varphi)} = -\frac{2\gamma(\varphi)}{\alpha(\varphi)} \tag{A.43}$$

Thus the expression for the derivative of $\dot{\varphi}_\star^2$ is given by:

$$\frac{\partial\dot{\varphi}_\star^2}{\partial\varphi} = -2\dot{\varphi}_\star^2\frac{\beta(\varphi)}{\alpha(\varphi)} - \frac{2\gamma(\varphi)}{\alpha(\varphi)} \tag{A.44}$$

A.8 Derivation of g_y, $g_{\dot{y}}$ and g_w

Inserting the expressions for ϑ, $\dot{\vartheta}$ and $\ddot{\vartheta}$ given in Equation 3.52 into the $\alpha\beta\gamma$-equation gives the following:

$$M_{21}(\ddot{y}+\Theta''(\varphi)\dot{\varphi}^2+\Theta'(\varphi)\ddot{\varphi})+M_{22}\ddot{\varphi}+c_{112}(\dot{y}+\Theta'(\varphi)\dot{\varphi})^2+c_{222}\dot{\varphi}^2+m_b g\vec{k}\boldsymbol{R}(y+\Theta(\varphi))\vec{\tau}s' = 0 \tag{A.45}$$

Here it can be observed that some of these terms are similar to $\alpha(\varphi)$ and $\beta(\varphi)$, this gives:

$$\alpha(\varphi)\ddot{\varphi} + M_{21}\ddot{y} + \beta(\varphi)\dot{\varphi}^2 + c_{112}(\dot{y}^2 + 2\dot{y}\Theta'(\varphi)\dot{\varphi}) + m_b g\vec{k}\boldsymbol{R}(y+\Theta(\varphi))\vec{\tau}s' = 0 \tag{A.46}$$

Moving some of the elements to the other side and adding $\gamma(\varphi)$ to both sides gives:

$$\alpha(\varphi)\ddot{\varphi} + \beta(\varphi)\dot{\varphi}^2 + \gamma(\varphi) = -M_{21}\ddot{y} - c_{112}(\dot{y}^2 + 2\dot{y}\Theta'(\varphi)\dot{\varphi}) - m_b g\vec{k}\boldsymbol{R}(y+\Theta(\varphi))\vec{\tau}s' + \gamma(\varphi) \tag{A.47}$$

$$g_{\dot{y}} = -m_b\vec{\rho}^\top\vec{\tau}s'(\dot{y} + 2\Theta'(\varphi)\dot{\varphi}) \tag{A.48}$$

$$g_w = -\left(s'\vec{\rho}\times\vec{\tau} - s'_f\frac{J_s}{R_b m_b}\right) \tag{A.49}$$

The derivation for g_y needs a few more steps:

$$\begin{aligned}
g_y &= \frac{1}{y}(\gamma(\varphi) - m_b g\vec{k}\boldsymbol{R}(y+\Theta(\varphi))\vec{\tau}s') && \text{(A.50)}\\
&= \frac{1}{y}(m_b g\vec{k}\boldsymbol{R}(\Theta(\varphi))\vec{\tau}s' - m_b g\vec{k}\boldsymbol{R}(y+\Theta(\varphi))\vec{\tau}s') && \text{(A.51)}\\
&= \frac{m_b g\vec{k}}{y}\left(\begin{bmatrix}\cos(\varphi) & -\sin(\varphi)\\ \sin(\varphi) & \cos(\varphi)\end{bmatrix} - \begin{bmatrix}\cos(y+\varphi) & -\sin(y+\varphi)\\ \sin(y+\varphi) & \cos(y+\varphi)\end{bmatrix}\right)\vec{\tau}s' && \text{(A.52)}\\
&= \frac{m_b g\vec{k}}{y}\begin{bmatrix}-2\sin(\frac{2\varphi+y}{2})\sin(-\frac{y}{2}) & -2\cos(\frac{2\varphi+y}{2})\sin(\frac{y}{2})\\ 2\cos(\frac{2\varphi+y}{2})\sin(-\frac{y}{2}) & -2\sin(\frac{2\varphi+y}{2})\sin(-\frac{y}{2})\end{bmatrix}\vec{\tau}s' && \text{(A.53)}\\
&= \frac{m_b g\vec{k}}{\frac{y}{2}}\begin{bmatrix}-\sin(\frac{\varphi+y}{2})\sin(-\frac{y}{2}) & -\cos(\frac{\varphi+y}{2})\sin(\frac{y}{2})\\ \cos(\frac{\varphi+y}{2})\sin(-\frac{y}{2}) & -\sin(\frac{\varphi+y}{2})\sin(-\frac{y}{2})\end{bmatrix}\vec{\tau}s' && \text{(A.54)}\\
&= m_b g\vec{k}\begin{bmatrix}\sin(\varphi+\frac{y}{2}) & 2\cos(\varphi+\frac{y}{2})\\ -\cos(\varphi+\frac{y}{2}) & 2\sin(\varphi+\frac{y}{2})\end{bmatrix}\operatorname{sinc}\left(\frac{y}{2}\right)\vec{\tau}s' && \text{(A.55)}\\
&= m_b g\begin{bmatrix}-\cos(\varphi+\frac{y}{2}) & \sin(\varphi+\frac{y}{2})\end{bmatrix}\operatorname{sinc}\left(\frac{y}{2}\right)\vec{\tau}s' && \text{(A.56)}
\end{aligned}$$

Her it has been used that $\vec{k} = \begin{bmatrix}0 & 1\end{bmatrix}$, $\operatorname{sinc}(x) = \frac{sin(x)}{x}$ and the two trigonometric identities

$$\sin(u) - \sin(v) = 2\cos\left(\frac{u+v}{2}\right)\sin\left(\frac{u-v}{2}\right) \tag{A.57}$$

and

$$\cos(u) - \cos(v) = -2\sin\left(\frac{u+v}{2}\right)\sin\left(\frac{u-v}{2}\right) \tag{A.58}$$

NTNU

Norwegian University of
Science and Technology