Master's thesis

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

Ruben B. Brecke

Performing Dynamic Manipulation of a Rolling Object on the Butterfly Robot with Perturbation in System Parameters

Master's thesis in Cybernetics and Robotics
Supervisor: Anton Shiriaev
June 2021

NTNU
Norwegian University of Science and Technology

Ruben B. Brecke

Performing Dynamic Manipulation of a Rolling Object on the Butterfly Robot with Perturbation in System Parameters

Master's thesis in Cybernetics and Robotics
Supervisor: Anton Shiriaev
June 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

One of the most challenging aspects of reproducing human behavior in robotics is performing non-prehensile (non-gripping) object manipulation. Robots must be able to move and perform complex activities if they are to be integrated into a human-built world.
This paper aims to introduce a new stabilizing controller to the Butterfly Robot, given that the transverse linearized dynamics have been found. The system consisting of a figure eight-shaped frame with a ball rolling on top was proposed as a benchmark example for developing new techniques within the field of non-prehensile manipulation. While previous stabilizing methods rely on a linear nominal feedback controller based on the stabilizing solution of the periodic Ricatti differential equation (PRDE), this paper will introduce a sliding mode controller to deal with unavoidable perturbations of the system.
The work presented in this paper uses the virtual holonomic constraint approach. Here, a scalar variable known as a motion generator is utilized in a geometric relation known as a virtual holonomic constraint to reduce the system's complexity by keeping the geometric relation invariant through a feedback controller. The nominal trajectory for the nonlinear system can then be derived based on the dynamics of the motion generator. The stabilization method makes use of the well-known equivalence between a nonlinear system's local, stable behavior and its linearized system's local, stable behavior. A set of transverse coordinates whose origin corresponds to the nominal trajectory can then be linearized and stabilized using a feedback controller such that its resulting nonlinear system will be stabilized as well.
Based on the transverse linearization, two alternative sliding mode controller architectures will be designed. The first method is simpler and is based on Lyapunov theory. A control law will be established that enables finite-time convergence towards a so-called sliding surface (that matches the nominal behavior). For the second technique, a real invariant subspace of the linearization will be created before its annihilator, whose co-dimensions equals the number of control input, is used in a switching function for the controller. Because both methods use the PRDE's stabilizing solution, semi-definite programming will be used to find an approximation.
The first method was proven to swiftly converge towards the nominal trajectory for nominal system parameters by numerical simulations. In contrast, the perturbed parameters showed a considerable stationary error but still converged towards the trajectory. The second method did not result in any successful stabilization, which was expected given that the found subspace failed to satisfy the required conditions, motivating for further work.

Sammendrag

En av de vanskeligste aspektene ved å reprodusere menneskelig atferd i robotikk er å utføre ikke-prehensile (ikke-gripende) objektmanipulering. Dersom roboter skal kunne integreres i en menneskeskapt verden, må de være i stand til å utføre slike komplekse aktiviteter.
Denne oppgaven tar sikte på å introdusere en ny stabiliserende kontroller til Sommerfugl roboten gitt at den transverse lineariserte dynamikken er funnet. Systemet bestående av en åtteformet ramme med en ball som rullet på toppen ble foreslått som et standardeksempel for utvikling av nye teknikker innen ikke-prehensil manipulasjon. Mens tidligere stabiliseringsmetoder er avhengig av en lineær nominell kontroller basert på stabiliserende løsningen av den periodiske Ricatti differensialligningen (PRDE), vil denne oppgaven introdusere en "sliding mode control" for å håndtere uunngåelige forstyrrelser i systemet.
Arbeidet presentert i denne oppgaven bruker en "virtual holonomic constraint" tilnærming. Her brukes en skalar-variabel kjent som en bevegelsesgenerator i et geometrisk forhold kjent som en virtual holonomic constraint for å redusere systemets kompleksitet ved å holde den geometriske relasjonen uforanderlig gjennom en feedback kontroller. Den nominelle banen for det ulineære systemet kan deretter avledes ut fra dynamikken til bevegelsesgeneratoren. Stabiliseringsmetoden bruker den velkjente ekvivalensen mellom et uineært systems lokale, stabile oppførsel og dets lineariserte systems lokale, stabile oppførsel. Et sett med transverse koordinater hvis opprinnelse tilsvarer den nominelle banen, kan deretter lineariseres og stabiliseres ved hjelp av en feedback regulator slik at det resulterende ulineære systemet også blir stabilisert.
På grunnlag av den transverse lineariseringen vil det utformes to alternative sliding mode kontrollere. Den første metoden er enklere og er basert på Lyapunov-teori, der det vil bli etablert kontrollov som muliggjør en endelig tidskonvergens mot en såkalt glidende overflate (som samsvarer med den nominelle oppførselen). For den andre teknikken vil det opprettes et reelt "subspace" av det lineariserte systemet før dens "annihilator", hvis samdimensjoner tilsvarer antall kontrollinnganger, brukes i en svitchfunksjon for kontrolleren. Fordi begge metodene bruker PRDEs stabiliserende løsning, vil "semi-definite" programmering brukes til å finne en tilnærming. Det ble påvist at den første metoden raskt konvergerte mot den nominelle banen for nominelle systemparametere ved numeriske simuleringer, mens for de forstyrrede parametrene viste en betydelig stasjonær feil, men likevel konvergerte mot banen. Den andre metoden resulterte ikke i noen vellykket stabilisering, som var forventet gitt at det ikke ble funnet et subspace som oppfylte de nødvendige betingelsene

Preface

This thesis is written as the concluding work of my Master of Science program in Cybernetics and Robotics at the Norwegian University of Science and Technology (NTNU) in Trondheim, Norway, during the spring of 2021. It documents my work done on implementing a stabilizing controller on a robotic system known as the Butterfly Robot. This has been the most extensive work during my time as a student at NTNU, and a great amount of effort has been put into both research and implementation of interesting subjects I have not encountered previously.

Acknowledgements
I would like to thank my supervisor Professor Anton Shiriaev for guidance on my work on both the pre-project from the fall of 2020, and in this thesis. I would also like to thank PhD Candidate Christian Fredrik Sætre for his help during the end of the work. Anton and Christian have provided crucial knowledge which without a doubt have contributed to the work done in this paper. I would also like to thank my family and friends for their support, especially my roommate and friend, Torkel, who have provided me with companionship during this semester. Last but definitely not least, I would also like give a special thanks to Marthe for all her emotionally support during what can be considered my toughest semester.

Contents

Abstract **i**

Sammendrag **iii**

Preface **v**

List of Figures **ix**

List of Tables **xi**

Nomenclature **xiii**

1 Introduction **1**
1.1 Background and Motivation 1
1.2 Outline of the Thesis 3

2 Theory **5**
2.1 Generalized Coordinates 5
2.2 Euler-Lagrange Equation of Motion 6
2.3 Periodic Trajectories 8
2.4 The Virtual Holonomic Constraint Approach 9
2.5 Stabilizing Controller Design 14
2.5.1 LQR 15
2.5.2 The Periodic Riccati Differential Equation 15
2.5.3 Sliding Mode Control 17

3 Equation of Motion **25**
3.1 Model Assumptions 25
3.2 Generalized Coordinates 25
3.3 Kinematics 28

3.4 Energy Functions 29
3.5 Equation of Motion 30

4 Motion Planning **33**
4.1 Reduced Dynamics 33
4.2 Desired Trajectory 34
4.3 Choosing a VHC 35
4.4 Simulating the Reduced Dynamics 35
4.5 Discussion 37

5 Controller Design **39**
5.1 Feedforward Controller 39
5.2 Stabilizing Controller 41
5.3 Sliding Mode Control 43
5.4 Discussion 58

6 Conclusion and Further Recommendations **61**
6.1 Conclusion 61
6.2 Recommendations for Further Work 62
References 64

Appendices

Appendix A Various Calculations and Equations **71**
A.1 The Christoffel Symbols 71
A.2 Constructing a Floquet-Lyapunov Factorization 72

List of Figures

1.1.1 A schematic view of the "Butterfly" Robot 2

2.5.1 Trajectory being restricted to head towards the surface S 18
2.5.2 Typical desired behavior in sliding mode control 19
2.5.3 Illustration of the undesired "chattering" effect 20

3.2.1 Illustration of the Butterfly Robot modeled with four degrees of freedom. . . 26
3.2.2 Illustration of the Butterfly Robot modeled with two degrees of freedom. . . . 27
3.2.3 Illustration of the relation between the radius of the ball R_b and the effective rolling radius R. 28

4.4.1 Phase portrait of the reduced dynamics with the geometric relation from Equation 4.3.3, with multiple different initial conditions. 36
4.4.2 Phase portrait of the reduced dynamics with the geometric relation from Equation 4.3.3, with initial conditions $(\varphi, \dot{\varphi}) = (0, 0.17)$. 37

5.1.1 Phase portrait of the nominal trajectory vs. the feedforward output trajectory. 40
5.3.1 Phase portrait of the nominal trajectory vs. the sliding mode control output trajectory, simulated with nominal system parameters. 45
5.3.2 Nominal input vs. actual input torque of the sliding mode controller, simulated with nominal system parameters. 46
5.3.3 The transverse coordinates under the sliding mode controller, simulated with nominal system parameters. 47
5.3.4 Zoomed in version of Figure 5.3.3. Note the axis values. 47
5.3.5 Phase portrait of the nominal trajectory vs. the sliding mode control output trajectory, simulated with system parameters scaled up by 10%. 49
5.3.6 Nominal input vs. actual input torque of the sliding mode controller, simulated with system parameters scaled up by 10%. 50

5.3.7 The transverse coordinates under the sliding mode controller, simulated with system parameters scaled up by 10%. 51
5.3.8 The nominal feedback matrix $K_\perp(s)$ given by Equation 5.3.11. 55
5.3.9 The elements of the switching function $S_\perp(s)$ 56
5.3.10 The matrix function $S_\perp(s)B_\perp(s)$ 56

List of Tables

4.1 Constants used in the geometric relation 35
4.2 System parameters used in simulations 36

Nomenclature

BVP	Boundary Value Problem
DOF	Degrees of Freedom
FL	Floquet-Lyapunov
LMI	Linear Matrix Inequality
LQR	Linear Quadratic Regulator
LTI	Linear Time Invariant
LTP	Linear Time Periodic
LTV	Linear Time Varying
MG	Motion Generator
ODE	Ordinary Differential Equation
PFL	Partial Feedback Linearization
PRDE	Periodic Ricatti Differential Equation
RDE	Ricatti Differential Equation
SDP	Semi-Definite Programming
SMC	Sliding Mode Control
STM	State Transition Matrix
VHC	Virtual Holonomic Constraint
w.r.t.	With respect to

Chapter 1

Introduction

1.1 Background and Motivation

When it comes to robotics and their applications, the research of non-prehensile (non-gripping) manipulation is becoming increasingly important [1]. The issue with such a topic is how to influence degrees of freedom that cannot be directly reached by a manipulator's control action. Thus, properties like pushing objects, performing robotic surgeries, and even walking are categorized as non-prehensile manipulation. The task of performing such manipulation is indeed challenging, both theoretically and practically [2]. A great effort has been invested into attempting to solve such an issue, motivated by the numerous applications that a solution to the problem may provide.

A couple of decades ago, Prof. Lynch et al. proposed a comprehensive analysis and implementation method based on a simplified benchmark example, which is still of interest within the robotic community today ([3], [2]). The benchmark example they proposed is currently known as the "Butterfly Robot," which consists of a figure-eight shape frame attached to an actuator through its shaft and a ball that can move freely. The frame is made up of two identical plates placed parallel to each other, making it possible for the ball to roll along its outer curve, see Figure 1.1.1.
The challenge they proposed was to find a family of feasible motions for the combined system, e.g., a perpetual rolling of the ball in one direction. Clearly, the ball is not attached to the frame in any way, making it possible to depart if the wrong torque is applied to the shaft. Note that once the ball has departed the frame, it is no longer possible to influence its motion.

One method that has proven to be successful in some applications in non-prehensile manipulation is the *virtual holonomic constraint* (VHC) approach [4]. Here, a scalar variable known as a motion generator is used in the VHC, which will be held invariant by a feedback controller to synchronize two configuration variables, effectively lowering the complexity. The dynamics of the motion generator can then be used as a candidate for the nominal trajectory of the nonlinear system.

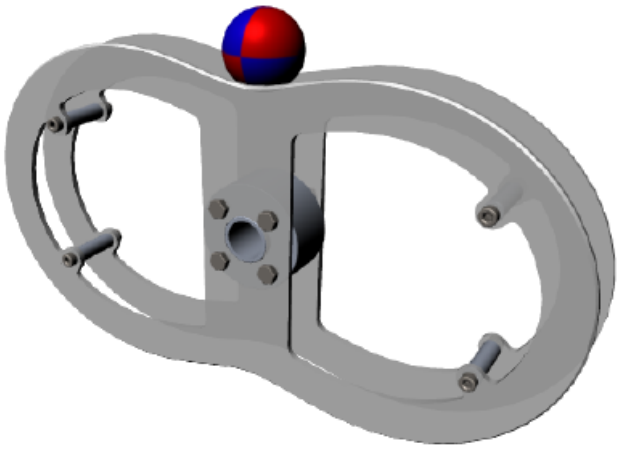


Figure 1.1.1: A schematic view of the "Butterfly" Robot

In [5] the VHC approach was used for orbital stabilization of a robotic system underactuated with degree one. In [6] it was shown that a set of *transverse coordinates*, whose origin corresponds to the nominal trajectory, could be linearized and complemented by a stabilizing controller. Using the well-known fact that the local behavior of a nonlinear system should comply with the local behavior of the linearized system, the stabilization of the transverse linearization should imply stabilization of the nonlinear system. In [2] both the VHC approach and the stabilization of the transverse dynamics was combined and applied to the Butterfly Robot, where an approximating solution of the *periodic ricatti differential equation* (PRDE) was used in a stabilizing feedback controller. A method for finding such an approximating solution of the PRDE was proposed in [7].

When it comes to stabilizing a mechanical system, several control strategies are available in the literature. One that has successfully been applied to robotic manipulators since the 1980s is the *sliding mode controller* (SMC) [8]. The benefits of using such a control scheme are its simple approach and design, yet obtaining robustness towards disturbances and variations in system parameters. In [9] they proposed a robust orbital stabilization method where the SMC synthesis was used to converge the linearized transverse dynamics towards its stable invariant subspace, which results in a stable invariant manifold of the original nonlinear mechanical system.

The main contribution of this thesis is to present how one can implement two different designs of a sliding mode controller to the Butterfly Robot. The work can be considered an extension of previous work as the transverse linearized dynamics have already been found, and a stabilizing controller is now needed. The fact that the Butterfly Robot is a practical example with real parameters being physically measured implies there are variations in such

a linearized system. This variation will naturally complicate any solution to the stabilization problem, as the solution should, in fact, hold for a family of systems simultaneously. The first part of this paper will derive the equations of motion and use the virtual holonomic constraint approach from [2] to generate a nominal trajectory, before the transverse linearized dynamics will be derived. In the second part, two different designs based on the SMC synthesis will be applied to the system to deal with uncertainties and disturbances in the modeling, as well as in the system parameters. One of these methods will introduce the transverse linearization to a general first-order stabilization method for SMC. In contrast, the second method is based on the technique presented in [9] where the annihilator of a real subspace of the linearized system will be used in a switching function for the controller. Both methods will use the approximating stabilizing solution of the PRDE, which will be found through the semi-definite programming method presented in [7]. Both SMC designs will then be numerically simulated and evaluated.

1.2 Outline of the Thesis

The outline of the thesis is as follows. In Chapter 2 the theoretical background of this paper is presented. This includes a quick notation on the generalized coordinates of a mechanical system; how to derive the Euler-Lagrange equation of motion; an introduction to periodic trajectories and orbital stability; the virtual holonomic constraint approach and how this can be used to find a feasible nominal trajectory, and lastly, the development of a stabilizing controller for a mechanical system. In Chapter 3 the model assumptions made throughout this thesis are stated; the generalized coordinates are found and used to obtain the kinematics; the energy functions of the system are computed and used to find the equation of motion using the Euler-Lagrange approach. In Chapter 4 the reduced dynamics of the system is found before the desired trajectory is chosen; a brief description on how to find a VHC are discussed before choosing one from literature; the reduced dynamics are then simulated, and the results are discussed. In Chapter 5 the feedforward controller for the system is derived and simulated before being complemented by a stabilizing term based on the sliding mode control scheme. The results from the simulations are then presented and discussed. Lastly, Chapter 6 will give a conclusion of this thesis, as well as some recommendations for further work.

Chapter 2

Theory

In this chapter, the theoretical background for this thesis is presented in order to supplement the reader with an understanding of the topics that will be used and discussed. The chapter is outlined as follows. Section 2.1 gives a quick introduction to generalized coordinates as well as the notation for position and velocity vectors, in which they are used. Section 2.2 present how to derive the Euler-Lagrange equations, which is a system of second-order *ordinary differential equations* (ODE). It will also give a definition of a fully actuated as well as underactuated systems. Section 2.3 discuss periodic trajectory of a nonlinear system, orbits and orbital stability. Section 2.4 present a method for motion planning for underactuated mechanical systems, as well as how to derive a linearized system based on its transverse dynamics. Lastly, Section 2.5 presents two different approaches to designing a sliding mode controller.

2.1 Generalized Coordinates

Consider a set of N particles, where each of them are free to move in any direction in a three-dimensional space. In order to specify the configuration of the system a set of $3N$ coordinates are needed. The system is then said to have $3N$ *degrees of freedom* (DOF). Using the Cartesian coordinate system, such a set would consist of $[x_1, y_1, z_1, ..., x_N, y_N, z_N]$, or, rewritten in a more general way; $[x_1, x_2, x_3, .., x_n]$ where $n = 3N$. These coordinates, denoted the *generalized coordinates*, are independent and the least amount of coordinates needed to accurately describe the configuration of the system. Conventionally speaking, when a set of generalized coordinates are used to describe displacements and angles, they are represented as $\mathbf{q}(t) = [q_1, ..., q_n]^\top$, where n is the minimum number of DOF. If there are any constraints between two of these coordinates, the complexity of $\mathbf{q}$ can be reduced to $n = 3N - r$, where r is the number of constraints [10].

Having found a set of general coordinates, the position, velocity and angular velocity vectors

for each object/particle can be defined.

$$\begin{aligned}\vec{r}_i &= \vec{r}_i(q_1, ..., q_n, t) \\ \vec{v}_i &= \vec{v}_i(q_1, ..., q_n, \dot{q}_1, ..., \dot{q}_n, t) = \frac{d\vec{r}_i}{dt} \\ \vec{\omega}_i &= \vec{\omega}(q_1, ..., q_n)\end{aligned} \tag{2.1.1}$$

2.2 Euler-Lagrange Equation of Motion

To find the equation of motion, i.e. the dynamics of a system, one approach is to derive the *Euler-Lagrange equations*. This is equivalent to using the *Newton's law of motion* in classical mechanics, however with the benefits of using the generalized coordinates instead [11].

The first step in deriving the Euler-Lagrange equations is to define the *Lagrangian*, $\mathcal{L}(q, \dot{q})$, given by

$$\mathcal{L}(q, \dot{q}) := \mathcal{K} - \mathcal{P} \tag{2.2.1}$$

where $\mathcal{K}$ is the kinetic energy of the system, and $\mathcal{P}$ is the potential energy. These can be found respectively by computing

$$\begin{aligned}\mathcal{P}_i &= m_i \vec{g} \cdot \vec{r}_i \\ \mathcal{K}_i &= \frac{1}{2} m_i \vec{v}_i \cdot \vec{v}_i + \frac{1}{2} J_i \vec{\omega}_i \cdot \vec{\omega}_i\end{aligned} \tag{2.2.2}$$

where i denote the object/particle, m is the mass, $\vec{g} = [0 \quad 9.81 \quad 0]^\top$ is the gravitational vector, J is the moment of inertia and ω is the angular velocity. The Euler-Lagrange equation can now be found by solving

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}(q, \dot{q})}{\partial q} = B(q)u \tag{2.2.3}$$

where the matrix $B(q)$ is the *coupling matrix* of the control input $u \in \mathbb{R}^m$. Equation 2.2.3 can further be written on the compact, standard form

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = B(q)u \tag{2.2.4}$$

where $M(q)$ is the *inertia matrix*, $C(q, \dot{q})$ is the *Christoffel symbols* matrix and contains the centrifugal and Coriolis terms and $G(q)$ is the *gravitational matrix*. The elements of the three

matrices in Equation 2.2.4 can be computed as follows

$$\begin{aligned} m_{ij} &:= \frac{\partial}{\partial \dot{q}_i}\left(\frac{\partial \mathcal{K}}{\partial \dot{q}_j}\right) \\ c_{jk} &:= \sum_{i=1}^{2} c_{ijk}(q)\dot{q}_i, \quad c_{ijk}(q) = \frac{1}{2}\left[\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k}\right] \\ g_i &:= \frac{\partial \mathcal{P}}{\partial q_i} \end{aligned} \tag{2.2.5}$$

Observing that the inertia matrix M is positive definite, and thus invertible, the equation of motion can be written w.r.t the acceleration vector $\ddot{q}$

$$\ddot{q} = M(q)^{-1}[-C(q,\dot{q})\dot{q} - G(q) + B(q)u] \tag{2.2.6}$$

Fully Actuated vs Underactuated

A mechanical system, like the one in Equation 2.2.4, with $q \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$, is said to be fully actuated if $n = m$ [12], meaning that the coupling matrix $B(q)$ will have full rank. Physically this can be interpreted that the system has as many DOF as it has control inputs to the system. For cases when $u = u(t)$ is a fixed control input, e.g. $u(t) = a\cos(t)$ for some constant a, the analysis of Equation 2.2.4 quickly becomes a challenging task [22]. However, when u is unrestricted and can be arbitrarily designed, it can be chosen as the nonlinear feedback law that cancels the dynamics in Equation 2.2.4 [22]

$$u = B(q)^{-1}\Big\{M(q)v + C(q,\dot{q})\dot{q} + G(q)\Big\} \tag{2.2.7}$$

In theory, with a perfect knowledge of the dynamic model, the feedback linearization law in Equation 2.2.7 will transform Equation 2.2.4 into the linear system of n decoupled chains of double integrators [4]

$$\ddot{y} = v \tag{2.2.8}$$

If, however, $m < n$, the coupling matrix $B(q)$ will not have full rank and the system is said to be underactuated of degree $n - m$. This implies that the control input u is restricted to a given subspace of $\mathbb{R}^n$ and the transformation in Equation 2.2.7 will not be valid [22]. However, as we will see later, the concept of *partial feedback linearization* (PFL) can be implemented.

2.3 Periodic Trajectories

In systems like the Butterfly Robot, it is often wanted to create a periodic behavior. The nonlinear system described by

$$\dot{x} = f(x) \tag{2.3.1}$$

is said to have a periodic solution $x^*(t) = x^*(t, x_0^*)$ if

$$x^*(t+T) = x^*(t) \quad \forall t$$

with a period $T > 0$. To numerically find a T-periodic solution $x^*(t)$ however, is not so trivial. In the case of finding an equilibrium point x_0 of Equation 2.3.1, one would only need to solve the algebraic equation $f(x_0) = 0$. To rather find a T-periodic solution, in general, one would need to integrate analytically Equation 2.3.1. Except for particular cases, this is very challenging [23]. Instead, one can use e.g. the Poincare-Bendixson Criterion to determine whether some bounded subset of the plane spanned out by Equation 2.3.1 contains any periodic orbits [4].

Theorem 2.3.1 ([4]) *Consider the system in Equation 2.3.1 and a bounded subset $\mathcal{M}$ of its plane such that*

- *$\mathcal{M}$ either contains no equilibrium points, or contains only one unstable node or unstable focus*
- *Every trajectory starting in $\mathcal{M}$, will stay in $\mathcal{M}$ for all time*

Then $\mathcal{M}$ contains a periodic orbit of Equation 2.3.1

Orbital Stability

As with all other mechanical systems, the importance of stability arises. However, it does not make sense to talk about asymptotic stability for systems with a periodic behavior [24]. Due to the presence of consistent perturbations in the phase, standard tools for stability analysis like linearization comes short. Consider the linearization of the system in Equation 2.3.1

$$\dot{z} = \left[\frac{\partial}{\partial x} f(x)\right]\Bigg|_{x=x^*(t)} \cdot z \tag{2.3.2}$$

Recall that the function $f(\cdot)$ is T-periodic. Clearly such a system cannot be asymptotically stable since $x(t)$ do not converge to zero as time approach infinity. The more appropriate term for stability would therefor be *orbital stability*. The definition of orbital stability is based on Definition 2.1 and 2.2 in [19].

Definition 1. *Consider the nonlinear system in Equation 2.3.1 and one of its solutions* $x^*(t) = x^*(t, x_0^*)$*. Let* $\mathcal{O}_{x^*} \subset \mathbb{R}^n$ *denote the orbit of the solution* $x^*(t)$*. The periodic solution* $x^*(t)$ *is called*

- *Orbitally stable, if for any* $\epsilon > 0$ *there exists a* $\delta > 0$ *such that*

  $$\textit{if} \quad \left\|x(0) - x_0^*\right\| < \delta, \textit{ then } \operatorname{dist}(x(t), \mathcal{O}_{x^*}) < \epsilon \quad \forall t \geq 0$$

- *Asymptotically orbitally stable, if orbitally* stable and

  $$\lim_{t \to +\infty} \operatorname{dist}(x(t), \mathcal{O}_{x^*}) = 0$$

The term $\operatorname{dist}(x(t), \mathcal{O}_{x^*})$ is referred to as the distance between the current state x and the orbit of interest $\mathcal{O}_{x^*}$.

2.4 The Virtual Holonomic Constraint Approach

In control theory, if a mechanical system is underactuated, it follows that it cannot be commanded to follow any arbitrarily trajectory [20]. Naturally this will complicate the task of motion planning for the system. However, by introducing a virtual holonomic constraint the complexity of the motion planning can be decreased. Consider a solution $q^*(t)$ to the system in Equation 2.2.4, obtained in response to the control signal $u^*(t)$. Suppose the motion $q^*(t)$ admits to the kinematic relations described by

$$q_1^* = \phi_1(\theta^*), \quad q_2^* = \phi_2(\theta^*), \quad ..., \quad q_n^* = \phi_n(\theta^*), \tag{2.4.1}$$

where θ^* is a scalar variable and $\phi_i : \theta^* \to \mathbb{R}, \quad i = 1, ..., n$ is $\mathcal{C}^2$-smooth functions. Equation 2.4.1 is referred to as an alternative, nested representation of the motion [23]. The kinematic relations is said to be a virtual holonomic constraints if the relations are kept invariant by a feedback controller [21]. The scalar variable θ^* is referred to as a *motion generator* (MG), and can to some extent be chosen freely [4]. However, it is often convenient to choose the MG as one of the generalized coordinates of the mechanical system, as it will be used as a synchronization variable [21]. Through the relation $\phi_i(\theta^*)$, herein referred to as a *'geometric relation'* or *'synchronization function'*, the motion generator will synchronize the motion of the system.

The Reduced Dynamics

Consider a mechanical system on the alternative nested representation, as in Equation 2.4.1. Also, let the system be underactuated of degree 1, meaning that one of the generalized coordinates can be described in terms of the motion generator. This implies there exist a dynamic constraint to the system due to the presence of one passive degree of freedom [21]. As it is well known, the dynamics of the motion generator should comply with this dynamic

constraint. This in turn implies that the dynamics of the motion generator can be uniquely described by the geometric relations. The dynamics of the motion generator enforced through the kinematic relations will herein be referred to as the *reduced dynamics*.

Given that the geometric relations are $\mathcal{C}^2$ smooth functions, it implies the corresponding relations of the first and second order derivatives, with θ as the motion generator

$$\begin{aligned} q(t) &= [\phi_1(\theta(t)); ...; \phi_n(\theta(t))] = \Phi(\theta(t)) \\ \dot{q} &= \Phi'(\theta)\dot{\theta} \\ \ddot{q} &= \Phi''(\theta)\dot{\theta}^2 + \Phi'(\theta)\ddot{\theta} \end{aligned} \tag{2.4.2}$$

To obtain the reduced dynamics the equation of motion from Equation 2.2.4 is multiplied with the annihilation matrix $B^{\perp}(q)$ of the coupling matrix $B(q)$. This is specifically designed to remove the rows that contains the control input. The kinematic relations and its derivatives are then substituted into Equation 2.2.4.

$$\begin{aligned} B(\Phi(\theta))^{\perp}\Big\{M(\Phi(\theta))[\Phi''(\theta)\dot{\theta}^2 + \Phi'(\theta)\ddot{\theta}]+ \\ C(\Phi(\theta), \Phi'(\theta)\dot{\theta})\Phi'(\theta)\dot{\theta} + G(\Phi(\theta))\Big\} &= B(\Phi(\theta))^{\perp}B(\Phi(\theta))u \\ &= 0 \end{aligned} \tag{2.4.3}$$

Collecting terms and writing as a differential equation w.r.t. the motion generator θ

$$\Big\{B(\Phi)^{\perp}M(\Phi)\Phi'\Big\}\ddot{\theta} + \Big\{B(\Phi)^{\perp}\left[M(\Phi)\Phi'' + \frac{C(\Phi, \dot{\Phi}\dot{\theta})\Phi'}{\dot{\theta}}\right]\Big\}\dot{\theta}^2 + B(\Phi)^{\perp}G(\Phi) = 0 \tag{2.4.4}$$

Three new variables, α, β, γ can now be defined

$$\begin{aligned} \alpha(\theta) &= B(\Phi)^{\perp}M(\Phi)\Phi', \\ \beta(\theta) &= B(\Phi)^{\perp}\left[M(\Phi)\Phi'' + \frac{C(\Phi, \dot{\Phi}\dot{\theta})\Phi'}{\dot{\theta}}\right], \\ \gamma(\theta) &= B(\Phi)^{\perp}G(\Phi), \end{aligned} \tag{2.4.5}$$

which leads to the compact form

$$\alpha(\theta)\ddot{\theta} + \beta(\theta)\dot{\theta}^2 + \gamma(\theta) = 0 \tag{2.4.6}$$

Equation 2.4.6 will be referred to as the *alpha-beta-gamma*, or $\alpha\beta\gamma$ equation. Given a feedback controller that holds Equation 2.4.1 invariant, the solution of Equation 2.2.4 tends asymptotically to the solution of Equation 2.4.6.

Looking at Equation 2.4.6, some observations can be made. Clearly θ will have an asymptote whenever $\alpha(\theta)$ approach zero. Thus, any solution θ^* must satisfy $\alpha(\theta^*) \neq 0 \ \forall \theta^* \in \Theta^*$.

Another property of Equation 2.4.6 is that it is integrable. Consider a solution with initial conditions $(\varphi_0, \dot{\varphi}_0)$. Any point $(\varphi^*, \dot{\varphi}^*)$ along this trajectory will preserve the zero value of the function [2]

$$I = \dot{\theta}^2(t) - \exp\Big\{ -2 \int_{\theta_0}^{\theta(t)} \frac{\beta(\tau)}{\alpha(\tau)} \, d\tau \Big\} \Big[\dot{\theta}_0^2 - \int_{\theta_0}^{\theta(t)} \exp\Big\{ 2 \int_{\theta_0}^{s} \frac{\beta(\tau)}{\alpha(\tau)} \, d\tau \Big\} \frac{2\gamma(s)}{\alpha(s)} \, ds \Big], \tag{2.4.7}$$

which will be referred to as the *integral of motion*.

Partial Feedback Linearization

As mentioned previously, if a mechanical system is underactuated it is not possible to linearize the whole system through a variable transformation, as in Equation 2.2.7, such that the resulting system can be written as

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= v \end{aligned} \tag{2.4.8}$$

However, a partial feedback linearization can still be used [25]. A PFL will transform the system dynamics into two parts: the external dynamics and the internal dynamics, which will become the linear and the nonlinear dynamics respectively [25].

Consider a T-periodic motion of an n DOF mechanical system $q^*(t) = [q_1^*(t), ..., q_n^*(t)] = q^*(t+T)$. Given a scalar function θ and the n-smooth functions for the motions nested representation, $\phi_1(\cdot), ..., \phi_n(\cdot)$, a new set of $(n+1)$ excessive coordinates can then be introduced as the generalized coordinates to describe the dynamics in a vicinity of $q^*(t)$ [26]

$$\theta, \quad y_1 := q_1 - \phi_1(\theta), \quad ..., \quad y_n := q_n - \phi_n(\theta) \tag{2.4.9}$$

Suppose that in a vicinity of the orbit $\mathcal{O}_{x^*}$, the generalized coordinates for the mechanical system can be described by

$$\theta, \ y_1, \ ..., \ y_{n-1}, \tag{2.4.10}$$

Using Equation 2.4.9, the new set can be written w.r.t. q and its derivatives

$$q = \begin{bmatrix} y \\ \theta \end{bmatrix} + \begin{bmatrix} \phi(\theta) \\ 0 \end{bmatrix}, \quad \dot{q} = L(\theta, y) \begin{bmatrix} \dot{y} \\ \dot{\theta} \end{bmatrix}, \quad \ddot{q} = L(\theta, y) \begin{bmatrix} \ddot{y} \\ \ddot{\theta} \end{bmatrix} + N(\theta, \dot{\theta}, y, \dot{y}), \tag{2.4.11}$$

where $L(\cdot)$ and $N(\cdot)$ are some matrix functions. Through the coordinate transformation $(q, \dot{q}) \to (y, \theta, \dot{y}, \dot{\theta})$, we can insert Equation 2.4.11 into Equation 2.2.6 to obtain

$$L(\theta, y)\begin{bmatrix}\ddot{y}\\ \ddot{\theta}\end{bmatrix} + N(\theta, \dot{\theta}, y, \dot{y}) = M(\theta, y)^{-1}\Big(- C(\theta, y, \dot{\theta}, \dot{y})\begin{bmatrix}\dot{y}\\ \dot{\theta}\end{bmatrix} - G(\theta, y) + B(\theta, y)u\Big), \quad (2.4.12)$$

where it is assumed that $L(\theta, y)$ is non-singular in a vicinity of the solution of Equation 2.4.10 where $(\theta = \theta^*(t),\ \mathbf{y}_{n-1} = \mathbf{0}_{n-1})$, such that Equation 2.4.12 can be written w.r.t. acceleration vector $\begin{bmatrix}\ddot{y}\\ \ddot{\theta}\end{bmatrix}$

$$\begin{bmatrix}\ddot{y}\\ \ddot{\theta}\end{bmatrix} = L(\theta, y)^{-1}\Big\{M(\theta, y)^{-1}\Big(- C(\theta, y, \dot{\theta}, \dot{y})L(\theta, y)\begin{bmatrix}\dot{y}\\ \dot{\theta}\end{bmatrix} - G(\theta, y) + B(\theta, y)u\Big) - N(\theta, \dot{\theta}, y, \dot{y})\Big\} \quad (2.4.13)$$

Which gives the expression for $\ddot{y}$ as the $(n-1)$ dimensional subspace [4]

$$\ddot{y} = R(\theta, \dot{\theta}, y, \dot{y}) + K(\theta, y)u, \quad (2.4.14)$$

where

$$\begin{aligned} K(\theta, y) &:= [1 \quad 0]L(\theta, y)^{-1}M(\theta, y)^{-1}B(\theta, y) \\ R(\theta, \dot{\theta}, y, \dot{y}) &:= [1 \quad 0]L(\theta, y)^{-1}\Big\{M(\theta, y)^{-1}\Big[- G(\theta, y) \\ &\qquad - C(\theta, \dot{\theta}, y, \dot{y})L(\theta, y)\begin{bmatrix}\dot{y}\\ \dot{\theta}\end{bmatrix}\Big] - N(\theta, \dot{\theta}, y, \dot{y})\Big\} \end{aligned} \quad (2.4.15)$$

In the case where the mechanical system is underactuated of degree 1, i.e. $n - m = 1$, and assuming $K(\theta, y)$ is non-singular in a vicinity of the solution $(\theta = \theta^*(t),\ y \equiv 0)$, the PFL controller

$$u = K(\theta, y)^{-1}\big(v - R(\theta, \dot{\theta}, y, \dot{y})\big) \quad (2.4.16)$$

will reduce the external dynamics in Equation 2.4.14 to [4]

$$\ddot{y} = v, \quad (2.4.17)$$

where v is a control input to be defined. The dynamics should also be complimented by the remaining internal dynamics, w.r.t. the MG, θ. Lets rearrange Equation 2.4.12 and solve for $B(\theta, y)u$

$$B(\theta, y)u = M(\theta, y)\Big(L(\theta, y)\begin{bmatrix}\ddot{y}\\ \ddot{\theta}\end{bmatrix} + N(\theta, \dot{\theta}, y, \dot{y})\Big) + C(\theta, y, \dot{\theta}, \dot{y})\begin{bmatrix}\dot{y}\\ \dot{\theta}\end{bmatrix} + G(\theta, y) \quad (2.4.18)$$

Clearly, by multiplying both sides of Equation 2.4.18 with the annihilation matrix $B(,y)^{\perp}$, the left-hand side becomes zero. Recall that on a solution $y = \dot{y} = \ddot{y} = \mathbf{0}$, the $\alpha\beta\gamma$ equation is zero. Thus Equation 2.4.18 can be rewritten as [26]

$$\alpha(\theta)\ddot{\theta} + \beta(\theta)\dot{\theta}^2 + \gamma(\theta) = g(\theta, \dot{\theta}, \ddot{\theta}, y, \dot{y}, v), \tag{2.4.19}$$

where $g(\cdot)$ is a smooth function equal to zero on the orbit, and Equation 2.4.17 have replaced $\ddot{y}$ with v. Note that in order to avoid any asymptotes, $\alpha(\theta)$ must be separated from zero on the trajectory. One can now use Hadamard's Lemma [27] about the orbit such that $g(\cdot)$ can be rewritten as

$$g(\theta, \dot{\theta}, \ddot{\theta}, y, \dot{y}, \ddot{y}, v) = g_y(\theta, \dot{\theta}, \ddot{\theta}, y, \dot{y})y + g_{\dot{y}}(\theta, \dot{\theta}, \ddot{\theta}, y, \dot{y})\dot{y} + g_v(\theta, \dot{\theta}, \ddot{\theta}, y, \dot{y})v, \tag{2.4.20}$$

where $g_y(\cdot), g_{\dot{y}}(\cdot)$ and $g_v(\cdot)$ are smooth functions. The arguments above are summarized in the following theorem.

Theorem 2.4.1 ([26]) *Let $q^*(t)$ be a solution to Equation 2.2.4, which is underactuated of degree 1. Let also the generalized coordinates be written on the nested representation given by Equation 2.4.10 such that $N(\theta, y)$ in Equation 2.4.14 is non-singular, then the partial feedback linearization in Equation 2.4.16 implies that in a vicinity of the orbit $\mathcal{O}_{x^*}$ of Equation 2.2.4, the dynamics can be rewritten as*

$$\begin{aligned} \alpha(\theta)\ddot{\theta} + \beta(\theta)\dot{\theta}^2 + \gamma(\theta) &= g_y(\theta, \dot{\theta}, \ddot{\theta}, y, \dot{y})y + g_{\dot{y}}(\theta, \dot{\theta}, \ddot{\theta}, y, \dot{y})\dot{y} \\ &\quad + g_v(\theta, \dot{\theta}, \ddot{\theta}, y, \dot{y})v \\ \ddot{y} &= v \end{aligned} \tag{2.4.21}$$

Transverse Coordinates

Let a set of $(n+1)$ excessive coordinates be given by Equation 2.4.9. Clearly these are not all independent, meaning at least one of them is redundant. By assuming the variables $\theta, y = [y_1, ...y_{n-1}]$ are independent, it turns out that a natural choice of candidates for the transverse coordinates $x_\perp$ for the motion, assuming the conditions of Theorem 2.4.1 holds, is given by

$$x_\perp = \Big[I(\theta, \dot{\theta}, \theta^*(0), \dot{\theta}^*(0);\ \ y_1;\ \ ...;\ \ y_{n-1};\ \ \dot{y}_1;\ \ ...;\ \ \dot{y}_{n-1} \Big], \tag{2.4.22}$$

in a vicinity of the solution $y = \mathbf{0}$ and the integral of motion $I(\cdot)$ is defined in Equation 2.4.7 [4].

Transverse Linearization

Having found that for a system which is underactuated of degree one, the dynamics in a vicinity of the orbit can be written on the form in Equation 2.4.21, the transverse linearization

can be found. The dynamics of the transverse coordinates will be on the form

$$\dot{x}_\perp = [\dot{I};\ \dot{y};\ \ddot{y}], \tag{2.4.23}$$

which again can be linearized along the solution of Equation 2.4.21, given by

$$y_{1^*} = 0,\ \ ..., y_{(n-1)^*},\ \ \theta = \theta^*(t),\ \ v^* = 0 \tag{2.4.24}$$

The linearization of the integral function $I(\cdot)$ can be written on the form [26]

$$\frac{d\bar{I}}{dt} = a_{11}(t)\bar{I} + a_{12}(t)\bar{Y}_1 + a_{13}(t)\bar{Y}_2 + b_1(t)\bar{V}, \tag{2.4.25}$$

where

$$\begin{aligned} k(t) &= \frac{2\dot{\theta}^*}{\alpha(\theta^*(t))} \\ a_{11}(t) &= k(t)\cdot\beta(\theta^*(t) \\ a_{12}(t) &= k(t)\cdot g_y(\theta^*(t), \dot{\theta}^*(t), \ddot{\theta}^*(t), 0, 0) \\ a_{13}(t) &= k(t)\cdot g_{\dot{y}}(\theta^*(t), \dot{\theta}^*(t), \ddot{\theta}^*(t), 0, 0) \\ b_1(t) &= k(t)\cdot g_v(\theta^*(t), \dot{\theta}^*(t), 0, 0), \end{aligned} \tag{2.4.26}$$

where $g_y(\cdot)$,$_{\dot{y}}(\cdot)$ and $g_v(\cdot)$ are from Theorem 2.4.1. Similarly, the linearization of y and $\dot{y}$ will yield the remaining dynamics. Recall that for a system underactuated of degree one, the controller in Equation 2.4.16 will reduce the external dynamics to $\ddot{y} = v$. Thus the transverse linearization can be written on the form

$$\dot{x}_\perp = \mathcal{A}_\perp(t)x_\perp + \mathcal{B}_\perp(t)v, \tag{2.4.27}$$

where the T-periodic matrices $\mathcal{A}(t)$ and $\mathcal{B}(t)$ are given by

$$\mathcal{A}_\perp(t) = \begin{bmatrix} a_{11}(t) & a_{12}(t) & a_{13}(t) \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathcal{B}_\perp(t) = \begin{bmatrix} b_1(t) \\ 0 \\ 1 \end{bmatrix} \tag{2.4.28}$$

2.5 Stabilizing Controller Design

Given a linear time-varying system on the form

$$\dot{x}(t) = A(t)x(t) + B(t)u(t), \tag{2.5.1}$$

where $x(t) \in \mathbb{R}^n$ are the states of the system, $u(t) \in \mathbb{R}^m$ is the control inputs, $A(t) \in \mathbb{R}^{n\times n}$ and $B(t) \in \mathbb{R}^{n\times m}$.

In most cases, such a system do not have a stable origin when the system is unactuated, meaning there is no control input [13]. The task is therefor to design a controller $u(t)$ such that the closed-loop system becomes stable.

2.5.1 LQR

One basic optimization problem for linear control is the *linear regulator* problem, where the control function $u(t)$ from Equation 2.5.1 is assigned a quadratic cost. The task is then to minimize the cost function

$$J = \int_0^{t_f} \Big[x^\top(t)Q(t)x(t) + u^\top(t)R(t)u(t)\Big]dx + x^\top(t_f)S_f x(t_f), \tag{2.5.2}$$

where $Q(t)$ is positive semidefinite, $R(t)$ is positive definite and S_f is positive semidefinite. The integral term is the *running cost* where $x^\top Qx$ penalize deviation of the states from the origin, and $u^\top Ru$ penalize the actuator output. $x^\top Sx$ is the *end cost*, which penalize the final deviation of the states.

An optimal control input $u^*(t) \in \mathbb{R}^m$ to the suggested optimization problem is a control function that minimizes the cost of J over all possible control functions [4]. The corresponding solution is the optimal trajectory. It can be shown that if there exists a solution $H(t)$ to the *matrix Riccati Differential Equation* (RDE)

$$\mathcal{R}(H,t) = 0, \quad \forall t \geq 0, \tag{2.5.3}$$

where the differential Riccati operator is being defined as

$$\mathcal{R}(H,t) = \dot{H}(t) + A^\top(t)H(t) + H(t)A(t) - H(t)B(t)R^{-1}(t)B^\top(t)H(t) + Q, \tag{2.5.4}$$

where $A(t), B(t)$ comes from the dynamics of the system and $Q(t), R(t)$ comes from the running cost, then there exist an optimal feedback controller $u^*(t)$ [2] on the form

$$u^*(t) = -R^{-1}(t)B(t)H(t)x(t) \tag{2.5.5}$$

2.5.2 The Periodic Riccati Differential Equation

Consider a dynamical system on the form in Equation 2.5.1 with periodic coefficients, i.e. $A(t) = A(t+T)$ and $B(t) = B(t+T)$ for some $T > 0$ period. The optimal feedback controller for the system will have the same form as in Equation 2.5.5, where $R(t) = R(t+T)$ is the stabilizing solution of Equation 2.5.4, called the periodic Riccati differential equation (PRDE).

However, the task of finding such a solution $R(t)$ of the PRDE, and thus the optimal feedback control for a system with periodic coefficients, is non-trivial [33].

In [7] they proposed a method that do not involve finding a numerical solution to the differential equation, but rather a stabilizing solution found through approximating the solution of a constructed finite dimensional semidefinite programming (SDP) problem. The method is briefly presented here for convenience, as it will be used in the implementation. The rest of Section 2.5.2 is taken from [7].

Proposition 1 *Consider a system on the form in Equation 2.5.1 with A and B being continuous, T-periodic matrix function. The pair (A, B) is also stabilizable. Let also $Q > 0$ and $R > 0$ be continuous, T-periodic matrix functions. Then there exist a T-periodic, stabilizing solution H^+ of $\mathcal{R}(H, t) = 0$. Moreover, any T-periodic solution H of the Riccati inequality*

$$\mathcal{R}(H, t) \geq 0, \quad \forall t \in [0, T], \tag{2.5.6}$$

satisfy the inequality

$$H(t) \leq H^+(t) \tag{2.5.7}$$

The inequality in Equation 2.5.6 can be turned into an *linear matrix inequality* (LMI) through the Schur complement [29].

$$\mathcal{L}(H, t) \geq 0, \quad \forall t \in [0, T], \tag{2.5.8}$$

where

$$\mathcal{L}(H, t) = \begin{bmatrix} \dot{H}(t) + A^\top(t)H(t) + H(t)A(t) + Q(t) & H(t)B(t) \\ B^\top(t)H(t) & R(t) \end{bmatrix} \tag{2.5.9}$$

From Proposition 1 it follows that the stabilizing solution is unique and the maximal solution. The problem can then be reformulated into the following problem of maximizing the function

$$\mathcal{F}(H) = \int_0^T \operatorname{tr}(H(t))\, dt, \tag{2.5.10}$$

where T is the period and $\operatorname{tr}(H(t))$ is the trace of the matrix function $H(t)$. However, such an optimization problem is of infinite dimension The numerical solution is therefor replaced with the approximating stabilizing solution. Consider the finite dimensional version of Equation 2.5.8

$$\mathcal{L}(H, t_i) \geq 0, \quad t_i = (i-1)\frac{T}{N}, \quad i = 1, ..., N \tag{2.5.11}$$

To guarantee boundedness on H, the following system of LMI's are introduced

$$-dI_n \leq H(t_i) \leq dI_n, \quad i = 1, ..., N, \tag{2.5.12}$$

Lets define some necessary notation.

Definition 2 ([7]) *Let* $H = \sum_{k=-M}^{M} e^{\mathbf{i}k\omega t} F_k(H)$ *be a T-periodic trigonometric symmetric polynomial of dimension* $n \times n$*, where* $F_0(H)$ *is symmetric real matrix and* $F_k(H), k = 1, ..., M$ *are symmetric complex matrices.* $F_{-k}(H) = \bar{F}_k(H)$*. Denote* $\mathcal{H}_M$ *the vector space of all such polynomials of degree at most* M

The problem can be reformulated into the following problem

$(\mathcal{P}_{M,N})$: Maximize $\int_0^T \operatorname{tr}(H(t))\, dt$ over all $\mathcal{H} \in \mathcal{H}_M$ satisfying Equation 2.5.11 and Equation 2.5.12.

Note that maximizing a function $g(\cdot)$ is identical to minimizing $-g(\cdot)$. It was further shown in [7] that the error using the method above decreased exponentially as the approximation order M increased, until the round-off error where no longer negligible. The arguments are summarized in the following theorem.

Theorem 2.5.1 ([7]) Consider the twice differentiable, T-periodic matrix functions A, B, Q and R, where $Q(t) > 0$, $R(t) > 0 \quad \forall t \in [0, T]$. Let the pair (A, B) be stabilizible, and the constant d is known such that the stabilizing solution H^+ of the PRDE satisfy $\left\|H_c^+\right\| < d$. Denote $H_{M,N}^+$ a solution of $\mathcal{P}_{M,N}$. Then $\left\|H^+ - H_{M,N}^+\right\|_c \to 0$ as $M \to \infty$ and $\frac{N}{M^3} \to \infty$

2.5.3 Sliding Mode Control

Consider the single-input dynamical system given by

$$\dot{x} = f(x) + b(x)u, \tag{2.5.13}$$

where some of the parameters of $f(x)$ contain uncertainties, meaning the function is not exactly known, however, its bound is. Let also the function $b(x)$ contain uncertainties, but with a known sign. The control problem is to ensure that the state vector x tracks the desired trajectory, given by a time-varying state vector $x_d = [x_d, \dot{x}_d, ..., x_d^{(n-1)}]^\top$, when there are uncertainties present in in the functions $f(x)$ and $b(x)$.

Introducing the tracking error $\tilde{x}$ defined by $\tilde{x} = x - x_d = [\tilde{x}, \dot{\tilde{x}}, ..., \tilde{x}^{(n-1)}]^\top$. Also, let $S(t)$ define a surface that is time-varying. Introducing a scalar function

$$s(x, t) = (\frac{d}{dt} + \lambda)^{n-1} \tilde{x}, \tag{2.5.14}$$

where $\lambda > 0$ is a constant. The surface $S(t)$ is defined such that Equation 2.5.14 is zero on desired trajectory, i.e. on the surface S, $s(x, t) = 0$.

For a system given by Equation 2.5.13 with $n = 2$, the scalar function is given by

$$s = \dot{\tilde{x}} + \lambda \tilde{x}, \tag{2.5.15}$$

i.e. a weighted sum of the position and velocity tracking errors. It can be shown that such an nth-order *tracking problem* can be replaced by a first order *stabilization problem* [8]. The original problem of tracking x_d can therefor be achieved by choosing a control law u such that apart from the surface $S(t)$ we have

$$\frac{1}{2}\frac{d}{dt}s^2 \leq -\eta|s|\,, \tag{2.5.16}$$

where η is a positive design constant. Equation 2.5.16 is called the *reachability condition* or the *sliding condition*. This specific condition however, is called the η - reachability condition. Searching the literature will yield multiple different sliding conditions ([14], [15], [16]). Equation 2.5.16 can be interpreted as the squared "error", or the distance from the current trajectory to the surface S. Such a control law will therefor restrict the trajectory to head towards the surface, and remain there. Any trajectory that verifies Equation 2.5.16 is called a *sliding surface*, and the behavior that occurs on the surface is called the *sliding mode*. Figure 2.5.1 depicts how the trajectories away from the surface can "move" while still denoting the surface $S(t)$[8].

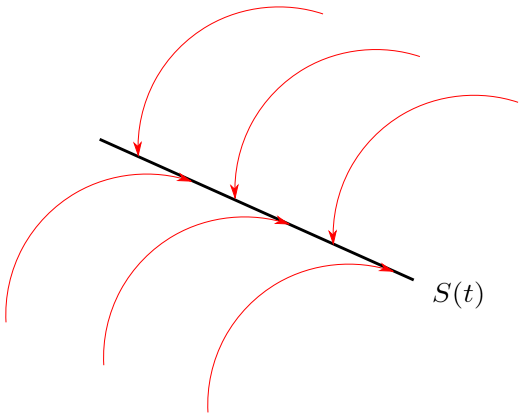


Figure 2.5.1: Trajectory being restricted to head towards the surface S

A typical system behavior under the control law from Equation 2.5.16 with $n = 2$ is shown in Figure 2.5.2. Here the sliding surface is represented by a straight line with slope $-\lambda$ in the phase plane of the system. The idea behind using the sliding mode control can be summarized by designing a feedback control law u that ensures the tracking error s can be used as a Lyapunov function of the closed loop, namely s^2. The paradigm have shown to be effective, even with the presence of uncertainties in the system parameters and external disturbances [28].

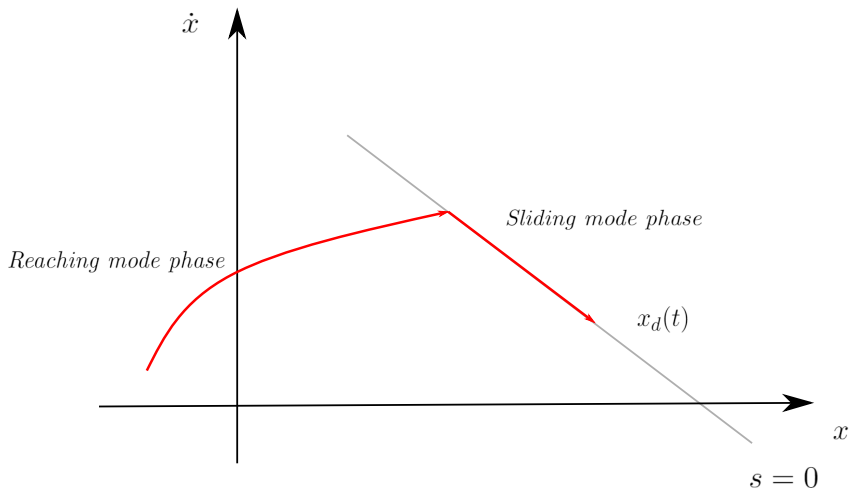


Figure 2.5.2: Typical desired behavior in sliding mode control

Note. As with everything in a physical world, the control switching is not perfect, i.e. it do not happen instantly fast, there are saturation etc. Together with the presence of modeling imprecision and disturbances, this effect can cause what is called *chattering*, see Figure 2.5.3. With a few important exceptions, this is undesired as it contains high control activity which can trigger neglected, high frequency dynamics in the system, and can cause unusual high wear and tear on equipment. To reach an optimal compromise between control bandwidth and tracking precision, the discontinuous control law u is smoothed accordingly in a vicinity of the sliding surface [8]. A switching action term u_{sw} defined by

$$u_{sw} = -Ksat(s) \tag{2.5.17}$$

is added to the control law. $K > 0$ is a design constant. The overall control law can therefor be written as

$$u = \hat{u} + u_{sw}, \tag{2.5.18}$$

where $\hat{u}$ deals with the low frequency control and u_{sw} deals with the high frequency control.

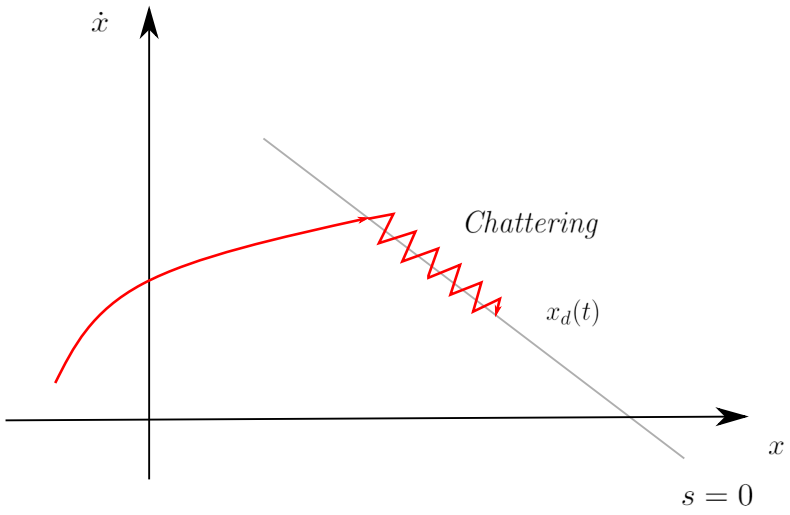


Figure 2.5.3: Illustration of the undesired "chattering" effect

Invariant Subspace-Based Switching Functions Design

In [9] they proposed a method based on the local equivalence between the stable invariant manifolds of a closed-loop, nonlinear mechanical system and the stable invariant subspace of its corresponding linearized system. The method is summed up in three parts

- Derive the linear periodic system of transverse linearization with a set of $(n-1)$ transverse coordinates.
- Transform the linear periodic system into an LTI system through a real Floquet-Lyapunov (FL) factorization of its state transition matrix.
- Construct a switching function for the LTI system, where one of its real invariant subspaces whose co-dimension equal the number of controls, is annihilated.

Considering the topics of transverse linearization along a nominal orbit already have been presented, the designing of such a switching function for an LTI will be presented first, before showing how one can transform a linear time periodic (LTP) system into an LTI system.

Invariant Subspace-Based Switching Function Design for LTI Systems
Consider the LTI system on the form

$$\dot{y} = Ay + B(u + \Delta(y, t)), \tag{2.5.19}$$

where A and B are constant, and $\Delta(y,t)$ is unkown, but with a known upper bound. Consider the problem of finding a full rank matrix $S \in \mathbb{R}^{m\times n}$ such that when restricted to the manifold

$$\sum := y \in \mathbb{R}^n : \sigma(y) := Sy \equiv 0 \tag{2.5.20}$$

the system in Equation 2.5.19 experiences the control $u_{eq} = Ky - \Delta(y,t)$, where K ensures $A^{cl} := A - BK$ is stable. It can be shown that if S satisfy $\det(SB) \neq 0$ and $Sy = 0 \Rightarrow SA^{cl}y = 0$, then S is a solution to the problem above. The following Lemma is based on Lemma 2 in [9]

Lemma 1 *Let S satisfy $det(SB) \neq 0$ and $Sy = 0 \Rightarrow SA^{cl}y = 0$. Suppose the control variable u is chosen as*

$$u = Ky + (SB)^{-1}v \tag{2.5.21}$$

in Equation 2.5.19 for some $v \in R^m$. Then the dynamics of $\sigma = Sy$ away from the manifold $\sum$ will be

$$\dot{\sigma} = \mathcal{A}_\sigma \sigma + v + SB\Delta(y,t), \tag{2.5.22}$$

where the constant matrix $\mathcal{A}_\sigma = SA^{cl}\hat{S}$ is Hurwitz, and $S\hat{S} = \mathbf{I}$

Proof is given in [9]. One can now use one of the several control strategies available in litterature to ensure the origin of Equation 2.5.21 is reached in finite time, even when considering the perturbation $\Delta(y,t)$ [9].

Linear Time Periodic Systems and Floquet-Lyapunov Theory
Consider a LTP system on the form

$$\dot{x} = A(t)x + B(t)(u + \Delta(x,t)), \tag{2.5.23}$$

where $A(t) = A(t+T)$ and $B(t) = B(t+T)$ are T-periodic, bounded matrix functions. $\Delta(x,t)$ is unknown, but with a known upper bound. Assume a continuus, T-period matrix function K is known, such that the closed loop, disturbance-free system given by

$$\dot{\mathcal{X}} = \mathcal{A}^{cl}(t)\mathcal{X}, \quad \mathcal{A}^{cl} := A(t) + B(t)K(t) \tag{2.5.24}$$

is exponentially stable. Then, any non-singular solution $\mathcal{X}(t)$ of Equation 2.5.24 is called a fundamental matrix to Equation 2.5.24 [18]. Further, let $\Phi_{\mathcal{A}^{cl}}(t,t_0)$, denoted the *state transition matrix*, be the unique solution to

$$\dot{\Phi}_{\mathcal{A}^{cl}}(t,t_0) = \mathcal{A}^{cl}(t)\Phi_{\mathcal{A}^{cl}}(t,t_0), \quad \Phi(t_0,t_0) = \mathbf{I}_n \tag{2.5.25}$$

which is equivalent to all the eigenvalues of the *Monodromy matrix*

$$\mathcal{M}_{\mathcal{A}^{cl}} := \Phi_{\mathcal{A}^{cl}}(T,0) \tag{2.5.26}$$

having a magnitude strictly lesser than one.

Similarly as with the LTI system, consider the problem of now finding a matrix function $S : \mathbb{R}_+ \to \mathbb{R}^{m\times n}$ such that the forward invariance of the relation $S(t)y(t) \equiv 0 \ \forall t \geq t_0$ corresponding to the system in Equation 2.5.23 experiencing the control $u_{eq}(t) = K(t)y(t) - \Delta(y,t) \ \forall t \geq t_0$. Consider the following Lemma, based on Lemma 3 in [9].

Lemma 2 *Let X_0 be of full rank and that the matrix function $S : \mathbb{R}_+ \to \mathbb{R}^{n\times(n-m)}$ is a left annihilator of $\Phi_{\mathcal{A}^{cl}}(t,0)X_0$ such that $\left\|S(t)\Phi_{\mathcal{A}^{cl}}(t,0)X_0 p\right\| \equiv 0$, where $p \in \mathbb{R}^{(n-m)}$. Then $S(t)$ is a solution to the problem above if rank$[S(t)B(t)] = m$. Moreover, if $S(t) = S(t+T)$, then X_0 is a basis of an invariant subspace of $\mathcal{M}_{\mathcal{A}^{cl}}$.*

Proof is given in [9]. This implies that if we are able to smoothly transform the LTP system into an LTI system, the theory presented previously could be used to generate a solution to the original problem of stabilizing the periodic transverse system.

To do so, consider the *Floquet-Lyapunov transformation* (FL transformation). Let a *linear time varying* (LTV) system be given by

$$\dot{y} = A(t)y, \tag{2.5.27}$$

where $A(t)$ is bounded. Denote Φ_A the STM of the system, and let there exist a real matrix $F \in \mathbb{R}^{(n\times n)}$ and a continuously differentiable, non-singular matrix function $L \in \mathbb{R}^{(n\times n)}$ such that the STM can be factorized as

$$\Phi_A(t,0) = L(t)e^{FT} \tag{2.5.28}$$

The following theorem can then be used.

Theorem 2.5.2 [30] *If there exist a T-periodic matrix $A(t) = A(t+T)$, then there also exist a real commuting matrices F and Y and a non-singular, continuously differentiable matrix function L(t) such that*

$$L(t) = L(t+2T), \quad L(t)Y = L(t+T), \quad Y^2 = I_n \tag{2.5.29}$$

such that Equation 2.5.28 holds for Equation 2.5.27

One can now use e.g. the method proposed in [18], where a boundary value problem was solved to derive the FL factorization of the system.

Transverse Coordinates-based Switching Function

Having already found the transverse linearized system along a solution $\mathcal{O}_{x^*}$, we define the matrix function $K_\perp$ as the nominal feedback that ensures

$$\dot{x}_\perp = [\mathcal{A}_\perp + \mathcal{B}_\perp K_\perp]x_\perp := \mathcal{A}_\perp^{cl} x_\perp \tag{2.5.30}$$

Denote $\Phi_\perp^{cl}$ the corresponding STM, i.e.

$$\dot{\Phi}_\perp^{cl} = \mathcal{A}_\perp^{cl}\Phi_\perp^{cl}, \quad \Phi_\perp^{cl}(x_0) = \mathbf{I}_{(n-1)} \tag{2.5.31}$$

All the arguments presented above are summarized in the following theorem, which is based on the main contribution of [9].

Theorem 2.5.3 *Suppose there exist a matrix function $K(\cdot)$ such that the STM of the closed loop system of the transverse linearization admits a real s_T-periodic FL factorization*

$$\Phi_\perp^{cl}(s) = L(s)e^{sF} \tag{2.5.32}$$

Let there also exist a full rank matrix $\hat{S}$ such that

- $\det[\hat{S}L^{-1}(s)B_\perp(s)] = 0$
- $\hat{S}z = 0 \Rightarrow \hat{S}Fz = 0$

are satisfied, then for any projector operator

$$\sigma(x) := S_\perp(p(x))x_\perp(x), \tag{2.5.33}$$

where $S_\perp(s) := \hat{S}L^{-1}(s)$, any x in a vicinity of the orbital solution $\mathcal{O}_{x^*}$ of the transverse linearization will be exponentially stabilized.

Proof is given in [9]

Chapter 3

Equation of Motion

In this chapter, the Euler-Lagrange equations for the Butterfly Robot will be derived. The section is outlined as follows. Section 3.1 will state the model assumptions that will be considered during the modeling of the mechanical system. In Section 3.2 the general coordinates will be derived, before being used to find the kinematics for the system in Section 3.3. In Section 3.4 the energy functions will be derived, and in Section 3.5 the dynamics of the Butterfly Robot will be derived as a set of differential equations.

3.1 Model Assumptions

Before deriving any dynamics of the system, it would be convenient to state the assumptions that will be made throughout this thesis. This is because the choice of such assumptions will largely influence the dynamics. The following assumptions will be made

- The frame do not move in any direction, but rotate around its center axis.
- The two plates that makes up the frame is perfectly identical and aligned.
- The ball and the frame are considered rigid bodies without any imperfections.
- The ball will never depart from the frame, leaving to a situation where it cannot be influenced by any control input.

3.2 Generalized Coordinates

As any object in a 3D space will have 6 degrees-of-freedom each, three related to translation (x, y, z) and three related to rotation *(Pitch, yaw, roll)*, the Butterfly Robot originally have 12 degrees-of-freedom. However, using the assumptions stated above, these can be reduced. Denote $\vec{e}^0, \vec{e}^1, \vec{e}^2$ as the world frame, body fixed frame of the frame, and the body fixed frame

of the ball, respectively. Note, as the word frame is used for both the frame of the robot and the coordinate frame, context would need to be taken into consideration. By placing the body fixed frame $\vec{e}^1$ in the origin of the world frame $\vec{e}^0$, the complexity of the frame is reduced with five degrees-of-freedom. As the first model assumption states that it can only rotate in one direction, around z, the frame can be modeled using only the angle θ_f.

The assumptions combined will ensure that the ball will only operate in a 2D plane, (x_b, y_b) with a rotation θ_b. The system can be modeled using the four variables $(x_b, y_b, \theta_b, \theta_f)$, see Figure 3.2.1. To further reduce the complexity, the coordinate system will be transformed from Cartesian to Polar Coordinates.

This change will not influence the DOF of the frame, which will still only have θ_f. Introducing the vector $\vec{\rho}$ as the vector from the origin of the frame to the center of the ball, leads to an expression for its position

$$\vec{r}_b = \vec{\rho} \tag{3.2.1}$$

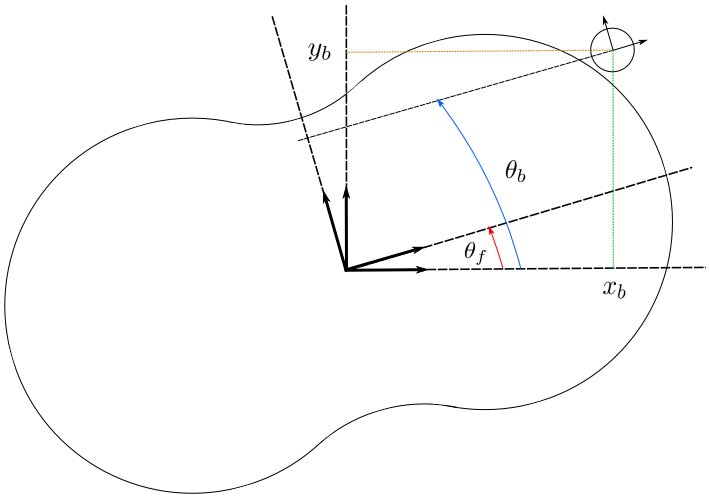


Figure 3.2.1: Illustration of the Butterfly Robot modeled with four degrees of freedom.

Let us also introduce a variable s, denoted the *arclength*. s describes the arc that is traced out by the ball's center point as it rolls around the curvature of the frame. As any vector $\vec{\rho}$ can be found through the arclength s if the shape of the frame is known, we re-write $\vec{\rho} = \vec{\rho}(s)$. To describe the motion of the ball, the *Frenet Frame* will be used. As the tangential vector of the ball $\vec{\tau}$ is defined as the derivative of the position $\vec{\rho}$ vector w.r.t. the arclength s, the Frenet frame can be defined

$$\vec{\tau} := \frac{d\vec{\rho}}{ds}, \qquad \vec{\kappa} := \frac{d^2\vec{\rho}}{ds^2}, \qquad \vec{n} := \hat{k} \times \vec{\tau}, \tag{3.2.2}$$

where $\hat{k} = [0\ 0\ 1]$ is the normal vector in the z-direction. In order to define the position of the ball along the frame, lets introduce a new variable φ. φ describes the angle between the body fixed frame of the frame $\vec{e}^1$ and the ball, see Figure 3.2.2. As any arclength s can be determined through any φ, we rewrite $s = s(\varphi)$ and $\vec{\rho}(s) = \vec{\rho}(\varphi)$.

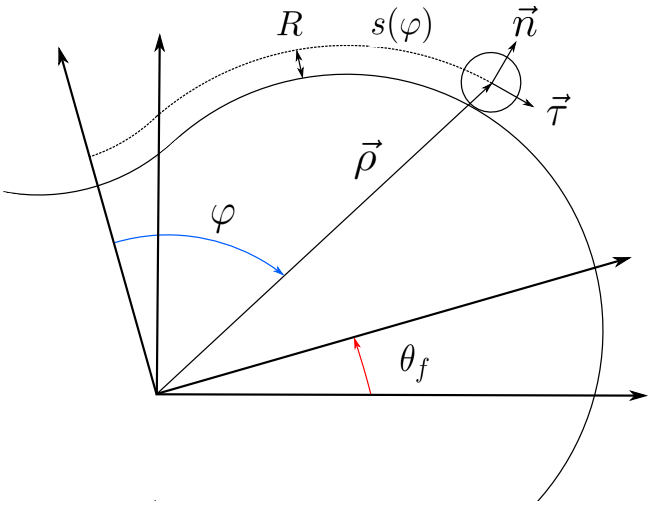


Figure 3.2.2: Illustration of the Butterfly Robot modeled with two degrees of freedom.

The last variable needed to describe the full configuration of the system is ψ, defined as the rotation of the ball w.r.t. $\vec{e}^1$. However, from the model assumptions stating that the ball rolls without slipping, clearly ψ is dependent on s, which again is dependent on φ. Note that the curve traced out by the balls center point and its contact point with the frame, will have an offset due to the distance between the two plates of the frame, see Figure 3.2.3. It can be seen that this curve do not have an offset equal to the radius of the ball R_b, but rather

$$R = \sqrt{R_b^2 - r_f^2}, \tag{3.2.3}$$

where R is the effective radius of the ball and $2r_f$ is the distance between the two plates of the frame.

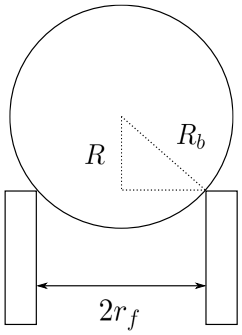


Figure 3.2.3: Illustration of the relation between the radius of the ball R_b and the effective rolling radius R.

The system can therefor be modeled using the two variables θ and φ, which will be chosen as the generalized coordinates of the system;

$$q = [\theta \;\; \varphi]^\top \tag{3.2.4}$$

3.3 Kinematics

Having found the generalized coordinates $q = [\theta \quad \varphi]^\top$, the next step in deriving the equation of motion is to obtain the kinematics. This includes the position, velocity and angular velocity for both the ball and the frame.

Frame

As the center point of the frame is placed at the world frame $\vec{e}^0$ its position and velocity will be identically zero

$$\vec{r}_f = \vec{v}_f = [0 \quad 0 \quad 0]^\top \tag{3.3.1}$$

Its angular velocity can be written as

$$\vec{\omega}_f = \dot{\theta}\hat{k} \tag{3.3.2}$$

with $\hat{k} = [0 \quad 0 \quad 1]$

Ball

As the vector $\vec{\rho}(\varphi)$ desicribes the position of the ball in the body attached frame $\vec{e}^1$, the position in world frame $\vec{e}^0$ can be expressed as

$$\vec{r}_b = \Pi(\theta)\vec{\rho}(\varphi) \tag{3.3.3}$$

where $\Pi(\theta)$ is the rotational matrix

$$\Pi(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3.4}$$

The velocity of the ball can be found through differentiation of the position

$$\begin{aligned} \vec{v}_b = \frac{d\vec{r}_b}{dt} &= \dot{\Pi}\vec{\rho} + \Pi\dot{\vec{\rho}} \\ &= (\dot{\theta}\hat{k}) \times (\Pi\vec{\rho}) + \Pi\frac{d\vec{\rho}}{ds}\frac{ds}{d\varphi}\frac{d\varphi}{dt} \\ &= \vec{\omega}_f \times (\Pi\vec{\rho}) + \vec{\tau}s'\dot{\varphi} \end{aligned} \tag{3.3.5}$$

To find the angular velocity of the ball, recall that it is proportional with the distance traveled. This can be expressed as

$$s_f = R\psi, \tag{3.3.6}$$

where s_f is the curve traced out by the contact between the frame and the ball. The total angular velocity of ball can therefor be written as

$$\vec{\omega}_b = \dot{\theta}\hat{k} - \dot{\psi}\hat{k} = \dot{\theta}\hat{k} - \frac{\dot{s_f}\hat{k}}{R} \tag{3.3.7}$$

The negative sign is of $\dot{\psi}$ due to the ball spinning in the opposite direction of the frame.

3.4 Energy Functions

Having found the kinematics of the system, Equation 2.2.2 can now be used to find the potential and kinetic energy.

Potential Energy
Inserting the position into the expression for potential energy, it can be found for both the frame and the ball

$$\begin{aligned} \mathcal{P}_f &= m_f\vec{g} \cdot \vec{r}_f = 0 \\ \mathcal{P}_b &= m_b\vec{g} \cdot \vec{r}_b = m_b\vec{g}(\Pi\vec{\rho}) \end{aligned} \tag{3.4.1}$$

The total potential energy for the system is then found by summing up the two functions

$$\mathcal{P} = \mathcal{P}_f + \mathcal{P}_b = m_b\vec{g}(\Pi\vec{\rho}) \tag{3.4.2}$$

Kinetic Energy

Similarly, the kinetic energy for the for the frame and the ball can be found.

$$\begin{aligned}
\mathcal{K}_f &= \frac{1}{2}m_f\vec{v}_f\cdot\vec{v}_f + \frac{1}{2}J_f\vec{\omega}_f\cdot\vec{\omega}_f \\
&= \frac{1}{2}J_f\dot{\theta}^2 \\
\mathcal{K}_b &= \frac{1}{2}m_b\vec{v}_b\cdot\vec{v}_b + \frac{1}{2}J_b\vec{\omega}_b\cdot\vec{\omega}_b \\
&= \frac{1}{2}m_b\left(\vec{\omega}_f\times(\Pi\vec{\rho}) + \vec{\tau}s'\dot{\varphi}\right)\left(\vec{\omega}_f\times(\Pi\vec{\rho}) + \vec{\tau}s'\dot{\varphi}\right) \\
&\quad + \frac{1}{2}J_b\left((\dot{\theta}+ps'\dot{\varphi})\hat{k}\right)\left((\dot{\theta}+ps'\dot{\varphi})\hat{k}\right) \\
&= \frac{1}{2}m_b\left(\dot{\theta}^2(\hat{k}\times(\Pi\vec{\rho})\cdot(\hat{k}\times(\Pi\vec{\rho})) + 2s'\dot{\theta}\dot{\varphi}(\Pi\tau)\cdot(\hat{k}\times(\Pi\vec{\rho})) + s'^2\dot{\varphi}^2\vec{\tau}^\top\Pi^\top\Pi\vec{\tau}\right) \\
&\quad + \frac{1}{2}(\dot{\theta}^2 + 2ps'\dot{\theta}\dot{\varphi} + p^2s'^2\dot{\varphi}^2)
\end{aligned} \tag{3.4.3}$$

And the total kinetic energy

$$\mathcal{K} = \mathcal{K}_f + \mathcal{K}_b \tag{3.4.4}$$

3.5 Equation of Motion

Following the procedure from the theory section, the Lagrangian can be defined.

$$\begin{aligned}
\mathcal{L} = \mathcal{K} - \mathcal{P} =& \frac{1}{2}(m_b\|\vec{\rho}\|^2 + J_f + J_b)\dot{\theta}^2 + (m_b\hat{k}\cdot(\vec{\rho}\times\vec{\tau}) + J_bp)s'\dot{\theta}\dot{\varphi} \\
&+ \frac{1}{2}(m_b + J_bp^2)s'^2\dot{\varphi}^2 - m_b\vec{g}\cdot(\Pi\vec{\rho})
\end{aligned} \tag{3.5.1}$$

To obtain the equation of motion on the compact form from Equation 2.2.4, Equation 2.2.5 will be used.

The Inertia Matrix $M(q)$

$$\begin{aligned}
m_{11} &= \frac{\partial}{\partial\dot{\theta}}\left(\frac{\partial\mathcal{K}}{\partial\dot{\theta}}\right) = m_b\|\vec{\rho}\|^2 + J_f + J_b \\
m_{12} &= \frac{\partial}{\partial\dot{\theta}}\left(\frac{\partial\mathcal{K}}{\partial\dot{\varphi}}\right) = (m_b\hat{k}\cdot(\vec{\rho}\times\vec{\tau}) + J_bp)s' \\
m_{21} &= \frac{\partial}{\partial\dot{\varphi}}\left(\frac{\partial\mathcal{K}}{\partial\dot{\theta}}\right) = m_{12} \\
m_{22} &= \frac{\partial}{\partial\dot{\varphi}}\left(\frac{\partial\mathcal{K}}{\partial\dot{\varphi}}\right) = (m_b + J_bp^2)s'^2
\end{aligned} \tag{3.5.2}$$

Which lead to the derivation of the inertia matrix

$$M(q) = \begin{bmatrix} m_b\|\vec{\rho}\|^2 + J_f + J_b & (m_b\hat{k}\cdot(\vec{\rho}\times\vec{\tau}) + J_b p)s' \\ (m_b\hat{k}\cdot(\vec{\rho}\times\vec{\tau}) + J_b p)s' & (m_b + J_b p^2)s'^2 \end{bmatrix} \tag{3.5.3}$$

The Coriolis and Centrifugal Matrix $C(q,\dot{q})$

Following the same procedure as in the inertia matrix, each elements of the Coriolis and centrifugal matrix can be computed as

$$C(q,\dot{q}) = \begin{bmatrix} m_b s'\vec{\rho}\cdot\vec{\tau}\dot{\varphi} & m_b s'\vec{\rho}\cdot\vec{\tau}\dot{\theta} + \left(\left(m_b\hat{k}\cdot(\vec{\rho}\times\vec{\tau}) + J_b p\right)s'' + m_b\hat{k}\cdot(\vec{\rho}\times\vec{\kappa})s'^2\right)\dot{\varphi} \\ -m_b s'\vec{\rho}\cdot\vec{\tau}\dot{\theta} & (m_b + J_b p^2)s's''\dot{\varphi} \end{bmatrix}, \tag{3.5.4}$$

where the computations are shown in Appendix A.1

The Gravity Matrix $G(q)$

Similarly the gravitational vector is found

$$\begin{aligned} g_1 &= \frac{\partial\mathcal{P}}{\partial\theta} = m_b\vec{g}\cdot\left(\frac{d\Pi}{d\theta}\vec{\rho}\right) = m_b\vec{g}\cdot(\Pi'\vec{\rho}) \\ g_2 &= \frac{\partial\mathcal{P}}{\partial\varphi} = m_b\vec{g}\cdot\left(\Pi\frac{d\vec{\rho}}{d\varphi}\right) = m_b\vec{g}\cdot(\Pi\vec{\tau}s') \end{aligned} \tag{3.5.5}$$

Where $\frac{d\vec{\rho}}{d\varphi} = \frac{dp}{ds}\frac{ds}{\varphi} = \vec{\tau}s'$ is found using Equation 3.2.2. The gravitation matrix is then given by

$$G(q) = \begin{bmatrix} m_b\vec{g}\cdot(\Pi'\vec{\rho}) \\ m_b\vec{g}\cdot(\Pi\vec{\tau}s') \end{bmatrix} \tag{3.5.6}$$

Expressing the Parameters

Considering that derivation of an analytical expression for the paramateres of the M, C, G matrices has been documented in multiple previous papers, it will not be repeated here. Instead, this can be found in e.g. this papers authors specialization project [31], or in [17].

Chapter 4

Motion Planning

In this chapter the theory presented in Section 2.4 will be used for the motion planning and deriving a feasible trajectory that can be used as the nominal trajectory of the system. The section is outlined as follows. In Section 4.1 the reduced dynamics of the system will be derived and written as a differential equation w.r.t. the motion generator. Section 4.2 will give a quick introduction to the desired trajectory that will be further investigated, while Section 4.3 will give an explanation as of how to choose a geometric relation. Section 4.4 will simulate the reduced dynamics before a quick discussion of the results are presented in Section 4.5.

As mentioned in the introduction, the Butterfly Robot is an underactuated mechanical system. This can be verified using the generalized coordinates $q = [\theta \quad \varphi]^\top$, i.e. $q \in \mathbb{R}^n$ with $n = 2$ and $u \in \mathbb{R}^m$ with $m = 1$. Hence, $m < n$, and the system is underactuaded of degree 1. This implies that the coupling matrix $B(q)$ do not have full rank, and no exact feedback linearization exist such that a control law can be defined as in Equation 2.2.7, effectively canceling the dynamics.

However, the introduction of a geometric relation between the generalized coordinates, also known as a VHC, has emerged as a valuable tool for solving motion control problems for underactuated systems and is a useful paradigm for control of oscillations [20].

4.1 Reduced Dynamics

Suppose that through some control input $u_\theta^*(t)$ there exist a motion $[\theta^*(t) \quad \varphi^*(t)]$ that satisfy the desired motion. Suppose there also exist some second order smooth function $\theta^*(t) = \Theta(\varphi^*(t))$. Here φ is chosen as the MG for θ. In other words, any motion of φ will create a motion for θ through the function Θ. Inserting the geometric relation into Equation 2.4.2,

the generalized coordinates are rewritten w.r.t. to φ.

$$\begin{aligned} q &= \begin{bmatrix} \Theta(\varphi) \\ \varphi \end{bmatrix} = \Phi(\varphi) \\ \dot{q} &= \begin{bmatrix} \Theta'(\varphi) \\ 1 \end{bmatrix} \dot{\varphi} = \Phi'(\varphi)\dot{\varphi} \\ \ddot{q} &= \begin{bmatrix} \Theta''(\varphi) \\ 0 \end{bmatrix} \dot{\varphi}^2 + \begin{bmatrix} \Theta'(\varphi) \\ 1 \end{bmatrix} \ddot{\varphi} = \Phi''(\varphi)\dot{\varphi}^2 + \Phi'(\varphi)\ddot{\varphi} \end{aligned} \tag{4.1.1}$$

Where

$$\Phi(\varphi) = \begin{bmatrix} \Theta(\varphi) \\ \varphi \end{bmatrix}, \quad \Phi'(\varphi) = \begin{bmatrix} \Theta'(\varphi) \\ 1 \end{bmatrix}, \quad \Phi''(\varphi) = \begin{bmatrix} \Theta''(\varphi) \\ 0 \end{bmatrix} \tag{4.1.2}$$

Defining the annihilator matrix $B^{\perp} = [0 \ 1]$, the system matrices $M(\cdot), G(\cdot)$ and $G(\cdot)$ can be inserted into Equation 2.4.5 to obtain the reduced dynamics

$$\alpha(\varphi)\ddot{\varphi} + \beta(\varphi)\dot{\varphi}^2 + \gamma(\varphi) = 0, \tag{4.1.3}$$

where

$$\begin{aligned} \alpha(\varphi) &= (m_b\hat{k} \cdot (\vec{\rho} \times \vec{\tau}) + J_b p)s'\Theta' + (m_b + J_b p^2)s'^2 \\ \beta(\varphi) &= (m_b\hat{k} \cdot (\vec{\rho} \times \vec{\tau}) + J_b p)s'\Theta'' - m_b s'\vec{\rho} \cdot \vec{\tau}\Theta'^2 + (m_b + J_b p^2)s's'' \\ \gamma(\varphi) &= m_b\vec{g} \cdot (\Pi\vec{\tau}s') \end{aligned} \tag{4.1.4}$$

4.2 Desired Trajectory

There are many different types of motion of interest in the Butterfly Robot. One can create continuous oscillations of the ball in one of the wells of the robot, or one can rotate the frame such that the ball goes from one well to the other. Here the well are referred to as the bump in the frame where the ball will naturally stay. In this thesis however, the desired motion is selected as a continuous, periodic perpetual rolling of the ball in one direction. Choosing this motion, a couple of observations can be made. It implies that the ball never stops rolling, meaning $\dot{\varphi}$ should never be zero. This also implies that its sign will always be the same.

The MG should be monotonically increasing, e.g. by choosing φ in this system. Further, from Figure 3.2.1 it can be seen that the frame of the robot is symmetric about both its $x - axis$ and the $y - axis$. This implies that motion of the ball, and thus the phase portrait of φ, will be periodic with a period of π.

4.3 Choosing a VHC

There are in theory many VHC that can be chosen as the geometric relation between the two generalized coordinates. Based on the observation from having chosen a desired trajectory, the conditions can be set on the behavior of the geometric relation. To show a few, in [2] they used

$$\Theta(\varphi) = \varphi + \arctan \frac{a \sin(2\varphi - \pi)[\Pi(\varphi)\vec{\tau}]_1 - [\Pi(\varphi)\vec{\tau}]_2}{a \sin(2\varphi - \pi)[\Pi(\varphi)\vec{\tau}]_2 + [\Pi(\varphi)\vec{\tau}]_1}, \tag{4.3.1}$$

where a is a constant and Π and $\vec{\tau}$ is from the dynamics of the Robot. In [17] they used the function

$$\Theta(\varphi) = \varphi - c \sin(2\varphi) \tag{4.3.2}$$

for some constant c.

In this thesis the chosen geometric relation is based on the pre-project of *Gustav Omberg Often* [32]. Here the relation $\theta = \Theta(\varphi)$ was used as

$$\Theta(\varphi) = c_1 \arctan(c_2 \sin(2\varphi) + c_3 \sin(4\varphi) + c_4 \sin(6\varphi) + c_5 \sin(8\varphi)) + \varphi, \tag{4.3.3}$$

where the constants $c_1 - c_5$ was found by solving an optimization problem of maximizing the centers in the phase plane of the reduced dynamics. The found constants are shown in Table 4.1

Table 4.1: Constants used in the geometric relation

Constant	Value
c_1	60.24
c_2	0.0091
c_3	0.0019
c_4	0.0017
c_5	0.000725

4.4 Simulating the Reduced Dynamics

Having selected a geometric relation between the two general coordinates, the dynamics of the motion generator φ can be simulated. The system parameters used in the simulations are shown in Table 4.2.

The reduced dynamics from Equation 4.1.3 was numerically simulated in MATLAB, resulting in the phase portrait shown in Figure 4.4.2, while Figure 4.4.2 shows the simulation with initial

Table 4.2: System parameters used in simulations

Constant	Value
m_b	$3.0e^{-3}$
J_b	$5.48e^{-7}$
J_f	$1.581e^{-3}$
R_b	$\sqrt{16.55e^{-3} - 12.5e^{-3}}$
a	0.1095
b	0.0405

conditions $(\varphi_0, \dot{\varphi}_0) = (0, 0.17)$. All code from chapter 4 are available on the Github link in [35].

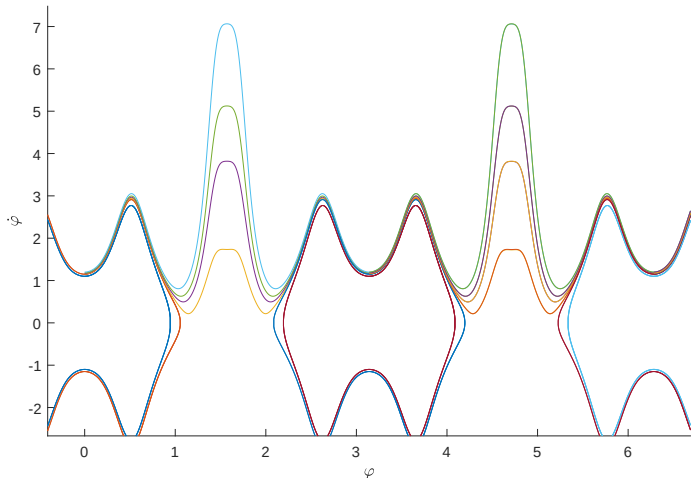


Figure 4.4.1: Phase portrait of the reduced dynamics with the geometric relation from Equation 4.3.3, with multiple different initial conditions.

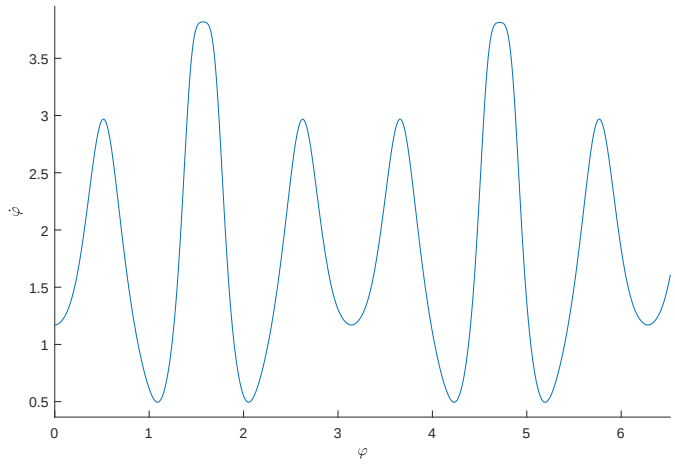


Figure 4.4.2: Phase portrait of the reduced dynamics with the geometric relation from Equation 4.3.3, with initial conditions $(\varphi, \dot{\varphi}) = (0, 0.17)$.

4.5 Discussion

In Figure 4.4.1 it can be seen that using the geometric relation in Equation 4.3.3, the initial condition for obtaining the desired trajectory must exceed $\approx 0.15\text{rad}/s$ in velocity. Considering that too high initial velocity will result in a high peak around $\frac{\pi}{2}$, the initial value will be chosen somewhere between. The trajectory seen Figure 4.4.2 will therefor be chosen as the desired trajectory, with initial conditions $(\varphi, \dot{\varphi}) = (0,\ \ 0.17)$.

Chapter 5

Controller Design

This chapter will present different controller designs for the Butterfly Robot. The section is outlined as follows. Section 5.1 will derive a feedforward controller as well as the performance of this using simulations. Section 5.2 will derive the transverse linearized system and show the steps in approximating a stabilizing solution to the PRDE. Section 5.3 will complement the feedforward controller with the stabilizing terms based on the principles of sliding mode control and present the results, before they are discussed in Section 5.4.

5.1 Feedforward Controller

Having found a nominal trajectory, the problem of developing a controller arises. The first and maybe intuitive task will therefor to develop a control variable u that will hold the geometric relation in Equation 4.3.3 invariant. Using Equation 2.4.14 and inserting Equation 2.4.15, such a controller can be defined

$$u^*(\varphi,\dot{\varphi}) = \frac{[1 \quad 0]L(\varphi)^{-1}\left\{M(\varphi)^{-1}\left(G(\varphi)+C(\varphi,\dot{\varphi})L(\varphi)\begin{bmatrix}0\\ \dot{\varphi}\end{bmatrix}\right)+N(\varphi,\dot{\varphi})\right\}}{[1 \quad 0]L(\varphi)^{-1}M(\varphi)^{-1}B(\varphi)}, \tag{5.1.1}$$

where $M(\cdot), C(\cdot)$ and $G(\cdot)$ are the system matrices found in Section 3.5, and $L(\cdot)$ and $N(\cdot)$ are defined as

$$L(\varphi) = \begin{bmatrix}1 & \phi'(\varphi)\\ 0 & 1\end{bmatrix}, \quad N(\varphi,\dot{\varphi}) = \begin{bmatrix}\phi''(\varphi)\dot{\varphi}^2\\ 0\end{bmatrix} \tag{5.1.2}$$

and u^* denotes that the system is on nominal trajectory.

The feedforward controller was then numerically simulated using MATLAB and Simulink. In Figure 5.1.1 it can be seen that the feedforward controller from Equation 5.1.1 is not able to stabilize the system. The system was simulated with the same initial conditions as for the

nominal trajectory. However, due to the presence of error and uncertainties in the numerical method, any small deviation from the nominal trajectory will result in divergence, which motivates for further stabilizing terms. All code from chapter 5 are available on the Github link in [35].

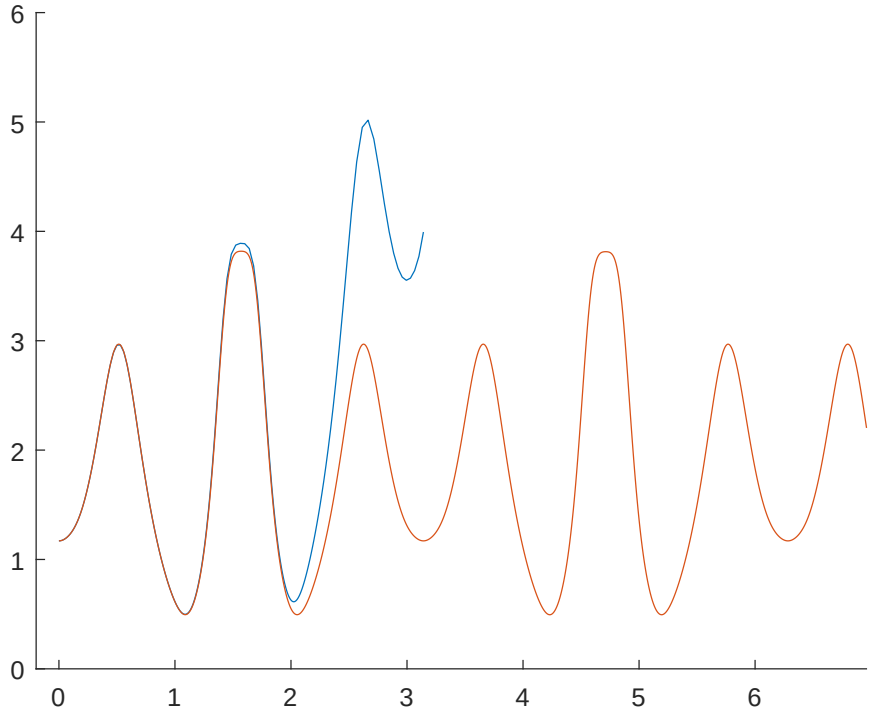


Figure 5.1.1: Phase portrait of the nominal trajectory vs. the feedforward output trajectory.

5.2 Stabilizing Controller

Having found that the feedforward controller for holding Equation 4.3.3 invariant is not able to stabilize the system alone, a stabilizing term will be added to the control variable in Equation 5.1.1. As it is well known, the stabilization of an orbit can be well interpreted by the stability of the transverse coordinates associated with the motion [2]. Thus, by designing a strategy for stabilizing the transverse linearized system, whose coordinates origin is the nominal orbit, the original nonlinear system should comply with a stable behavior as well. The dynamics will first be expressed in terms of the internal and external dynamics, before the transverse linearized system will be found. The technical detail around finding an approximating solution to the PRDE will then be explained.

Partial Feedback Linearization

Having chosen φ as the MG, the same procedure as in Section 2.4 can be followed to obtain the partial feedback linearization. The new set of general coordinates can be written as

$$\begin{aligned} q_1 &= y + \phi(\varphi) \\ q_2 &= \varphi \end{aligned} \tag{5.2.1}$$

yielding

$$\begin{aligned} \dot{q} &= \begin{bmatrix} 1 & \phi'(\varphi) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{y} \\ \dot{\varphi} \end{bmatrix} = L(\varphi) \begin{bmatrix} \dot{y} \\ \dot{\varphi} \end{bmatrix}, \\ \ddot{q} &= \begin{bmatrix} 1 & \phi'(\varphi) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \ddot{y} \\ \ddot{\varphi} \end{bmatrix} + \begin{bmatrix} \phi''(\varphi)\dot{\varphi}^2 \\ 0 \end{bmatrix} = Ł(\varphi) \begin{bmatrix} \ddot{y} \\ \ddot{\varphi} \end{bmatrix} + N(\varphi, \dot{\varphi}), \end{aligned} \tag{5.2.2}$$

where $L(\cdot)$ and $N(\cdot)$ are as defined in Equation 5.1.2. The PFL controller in Equation 2.4.14, where $K(\cdot)$ and $R(\cdot)$ are defined in Equation 2.4.15 will therefor bring the external dynamics to the linear system of double integrators $\ddot{y} = v$. What is left now is to compliment the dynamics with the remaining internal dynamics.

For this we will use Equation 2.4.19. By multiplying Equation 2.4.18 with the annihilation matrix $B^{\perp}$ and inserting Equation 5.2.2, we are left with

$$M_{21}(\ddot{y} + \Phi''\dot{\varphi}^2 + \Phi'\ddot{\varphi}) + M_{22}\ddot{\varphi} + C_{21}\dot{y} + C_{22}\dot{\varphi} + G_2 \tag{5.2.3}$$

Observe that by adding and subtracting the term $C_{21}\Theta'\dot{\varphi}$, some of the terms will be equal to α, β and γ. Also, considering the l.h.s of the equation is zero, we can move the remaining term over to the other side and collected as the function $g(\cdot)$

$$\ddot{\varphi}(M_{21}\Theta' + M_{22}) + \dot{\varphi}^2\left(M_{21}\Theta'' + \frac{C_{22}}{\dot{\varphi}} + \frac{C_{21}\Theta'}{\dot{\varphi}}\right) + G_2 = g(\varphi, \dot{\varphi}, y, \dot{y}, v) \tag{5.2.4}$$

Clearly the equation matches the internal dynamics from Theorem 3.4.1, where the left hand side is the $\beta\gamma$ equation. Thus we can now use Hadamard's Lemma such that the right hand side of Equation 5.2.4 becomes

$$g_y(\varphi, \dot{\varphi}, y, \dot{y})y + g_{\dot{y}}(\varphi, \dot{\varphi}, y, \dot{y})\dot{y} + g_v(\varphi, \dot{\varphi}, y, \dot{y}, v)v, \tag{5.2.5}$$

where

$$\begin{aligned} g_v &= -m_b s' \left(\vec{\rho} \times \vec{\tau} - \frac{J_b}{m_b R_b} \right) \\ g_y &= m_b s' \vec{\rho}^{\top} \vec{\tau} (\dot{y} + 2\Theta' \dot{\varphi}) \\ g_{\dot{y}} &= -m_b s' g \big[\cos\big(\frac{y}{2} + \varphi\big), \ -\sin\big(\frac{y}{2} + \varphi\big) \big] \vec{\tau} \mathrm{sinc}\big(\frac{y}{2}\big) \end{aligned} \tag{5.2.6}$$

Transverse Dynamics

To linearize the system, which is now on the same form as in Theorem 3.4.1, one can simply use Equation 2.4.25, Equation 2.4.26 and Equation 2.4.28 such that the transverse linearization can be written on the form

$$\dot{x}_{\perp} = \mathcal{A}(t)x_{\perp} + \mathcal{B}(t)w, \tag{5.2.7}$$

where

$$\mathcal{A}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ \frac{\bar{g}_y}{\bar{\alpha}} & \frac{\bar{g}_{\dot{y}}}{\bar{\alpha}} & \frac{\bar{\gamma} - \bar{\beta}\dot{\varphi}^2}{\bar{\alpha}\dot{\varphi}} \end{bmatrix}, \quad \mathcal{B}(t) = \begin{bmatrix} 0 \\ 1 \\ \frac{\bar{g}_v}{\bar{\alpha}} \end{bmatrix} \tag{5.2.8}$$

The bar indicates that the function is evaluated on nominal trajectory, i.e. $\bar{\alpha} = \alpha(\varphi^*)$.

Solving the Periodic Ricatti Differential Equation

To calculate the approximation of the stabilizing solution of the PRDE in Equation 2.5.4, the method presented in Section 2.5.2 was used. The reduced dynamics was simulated using the geometric relation from Section 4.3, before the matrices A and B corresponding to the transverse linearized system was obtained. The optimization problem was then defined using Equation 2.5.9, Equation 2.5.11 and Equation 2.5.12, and solved using the MATLAB SDPT3 Package [33].

The matrices Q and R was set to

$$Q = \begin{bmatrix} 20 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 20 \end{bmatrix}, \quad R = 1, \tag{5.2.9}$$

and the sample size N and polynomial size M from Theorem 2.5.1 was set to 100 and 40 respectively. After increasing both the step size and the polynomial size, there seemed to be no improvement in the method, which also was the case in the example used in the original paper [33].

5.3 Sliding Mode Control

The sliding mode control section consists of two different designs for the sliding surface, or the *switching function*. Design I consist of constructing a control law u that ensures the sliding condition in Equation 2.5.16 is upheld, while Design II will construct a switching surface based on the annihilation of a real invariant subspace of the linearized system.

SMC Design I

In this section a sliding mode controller will be developed based on the theory presented in Section 2.5.3.

Having found the linearized system in Equation 5.2.7 based on the transverse coordinates, the following scalar function $s(x_\perp, t)$ was defined as

$$s(x,t) = y + \dot{y} + z \tag{5.3.1}$$

Now, in order for the sliding condition in Equation 2.5.16 to be valid, clearly the condition

$$s\dot{s} < 0, \tag{5.3.2}$$

needs to be upheld. Differentiating s yields

$$\dot{s} = \dot{y} + \ddot{y} + \dot{z} = \dot{y} + w(1 + b_3) + a_{31}y + a_{32}\ddot{y} + a_{33}, \tag{5.3.3}$$

Where the parameters a_i and b_i are the elements of the transverse linearization matrices. Thus the condition in Equation 5.3.2 will be satisfied if

$$\begin{aligned} (y + \dot{y} + z) < 0: &\quad w > \frac{1}{1+b_3}(-a_{31}y - \dot{y}(1 + a_{32}) - a_{33}z) \\ (y + \dot{y} + z) > 0: &\quad w > \frac{1}{1+b_3}(a_{31}y + \dot{y}(1 + a_{32}) + a_{33}z) \end{aligned} \tag{5.3.4}$$

Which can be written in the compact form

$$w = g \cdot sgn(y + \dot{y} + z), \tag{5.3.5}$$

where $g = \frac{1}{1+b_3}(a_{31}y + \dot{y}(1 + a_{32}) + a_{33}z)$.

Implementation and Results

The method presented above was implemented in MATLAB and simulated using Simulink. The method however proved to be unsuccessful at first.

The sliding surface s used in the $sgn(\cdot)$ function was further complemented by the approximation of the stabilizing solution of the PRDE in Equation 2.5.4, which vastly improved the stabilization of the controller. The system was then tested with both nominal parameters and perturbed parameters.

Nominal System Parameters

The sliding mode controller was first simulated on the system using the nominal system parameters given in Table 4.2. The results are shown in Figure 5.3.1 - 5.3.4. In Figure 5.3.1 the plot is showing the nominal trajectory, φ^*, given by the reduced dynamics vs. actual output trajectory, φ, of the system. Given that the initial conditions of the simulated system is $(\varphi, \dot{\varphi}) = (0, 0)$, the controller seems to quickly catch up with the dynamic trajectory, before converging to some stationary value which will be discussed soon. In Figure 5.3.2 the input torque of the controller is compared with the nominal output given by the model. Clearly the controller suffers from the well-known downside of the SMC called chattering during the first period, before vanishing as the trajectory have converged. It should be mentioned that it was not a priority to completely remove this effect during the work presented here, as there are several strategies available in literature to do so [9].

Figure 5.3.3 shows the transverse coordinates throughout the simulation. As mentioned previously, the transverse coordinates will, in a vicinity of the nominal trajectory, give a measure of error between the current and the nominal trajectory. Clearly the error is quite large in the beginning of the run, considering that the system is initiated at zero. However, as we can see in Figure 5.3.4, which is just zoomed in on the previous plot, the transverse coordinates have converged to a periodic trajectory whose error doesn't exceed $|y|, |\dot{y}|, |z| < 0.5e^{-4}$

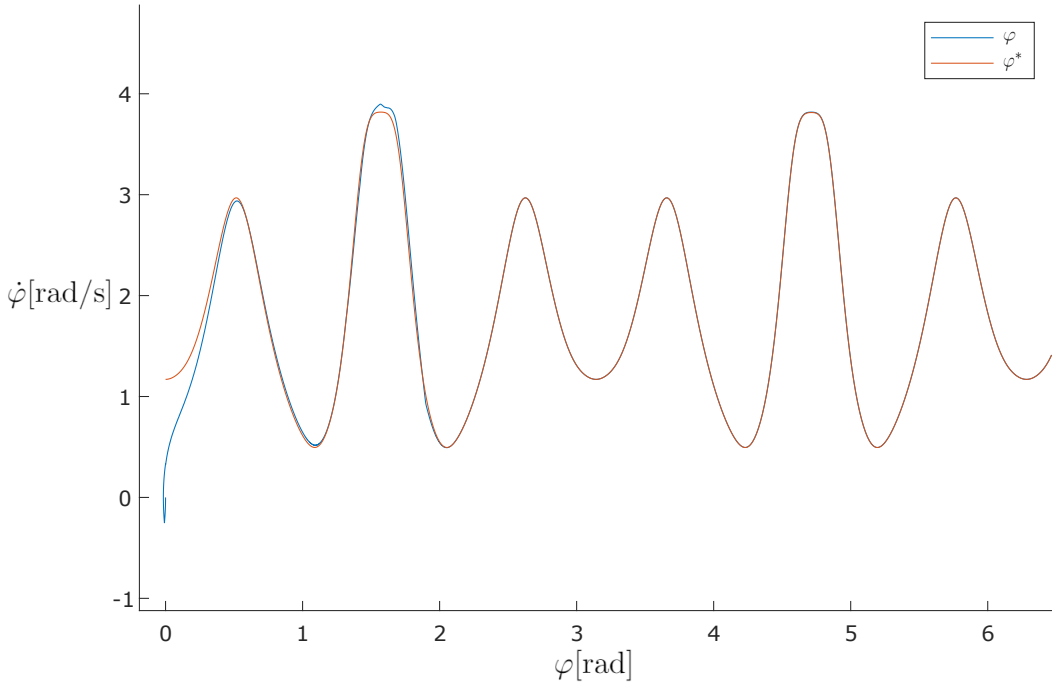


Figure 5.3.1: Phase portrait of the nominal trajectory vs. the sliding mode control output trajectory, simulated with nominal system parameters.

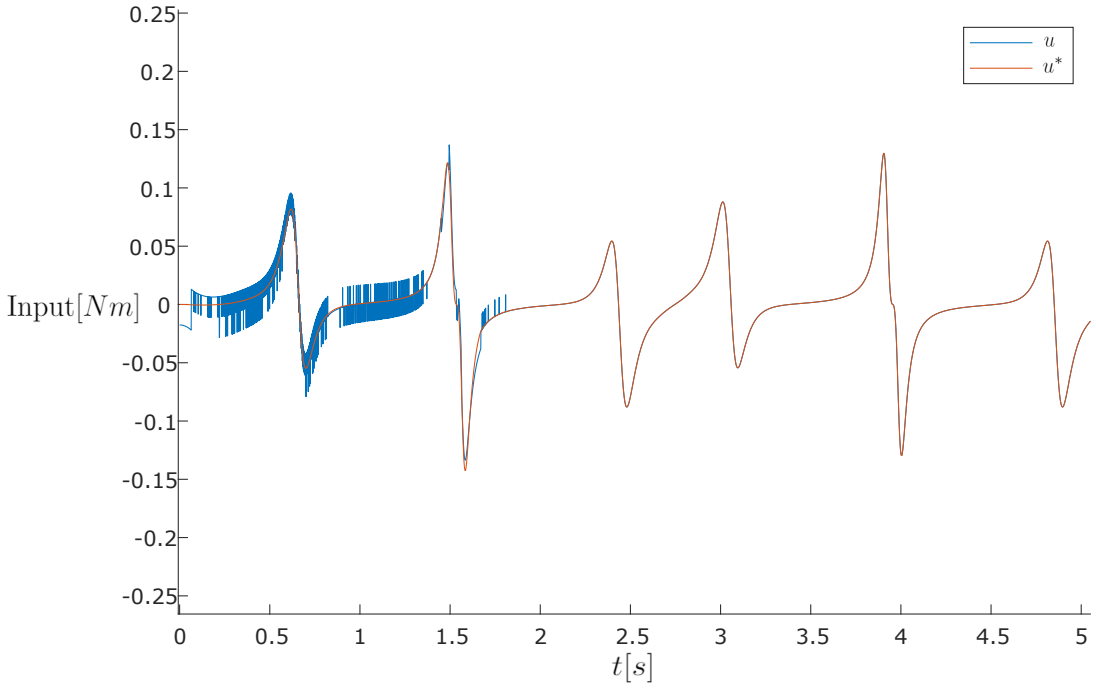


Figure 5.3.2: Nominal input vs. actual input torque of the sliding mode controller, simulated with nominal system parameters.

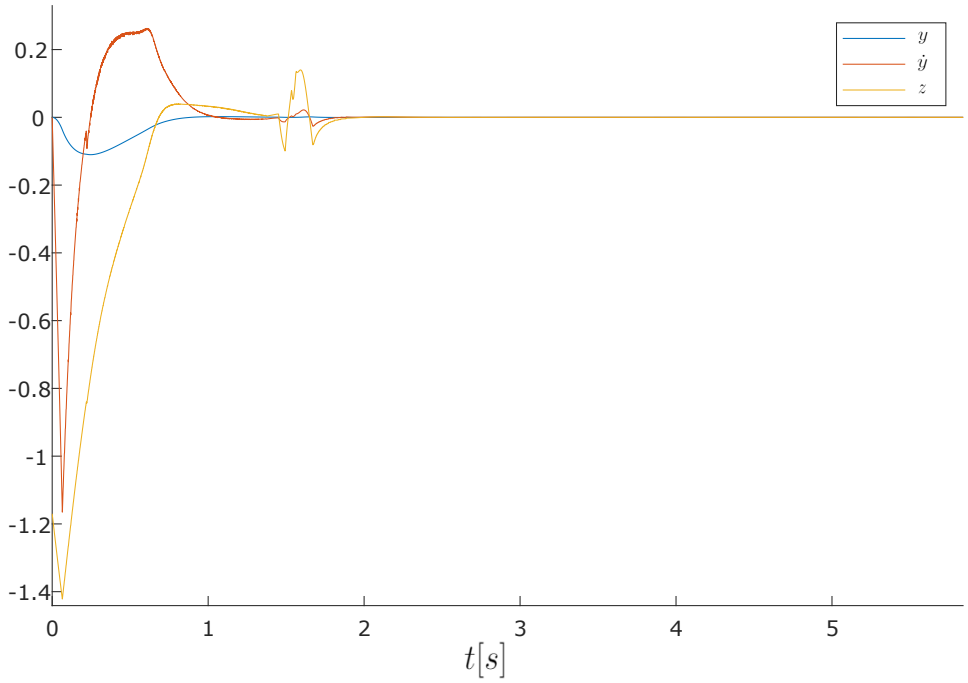


Figure 5.3.3: The transverse coordinates under the sliding mode controller, simulated with nominal system parameters.

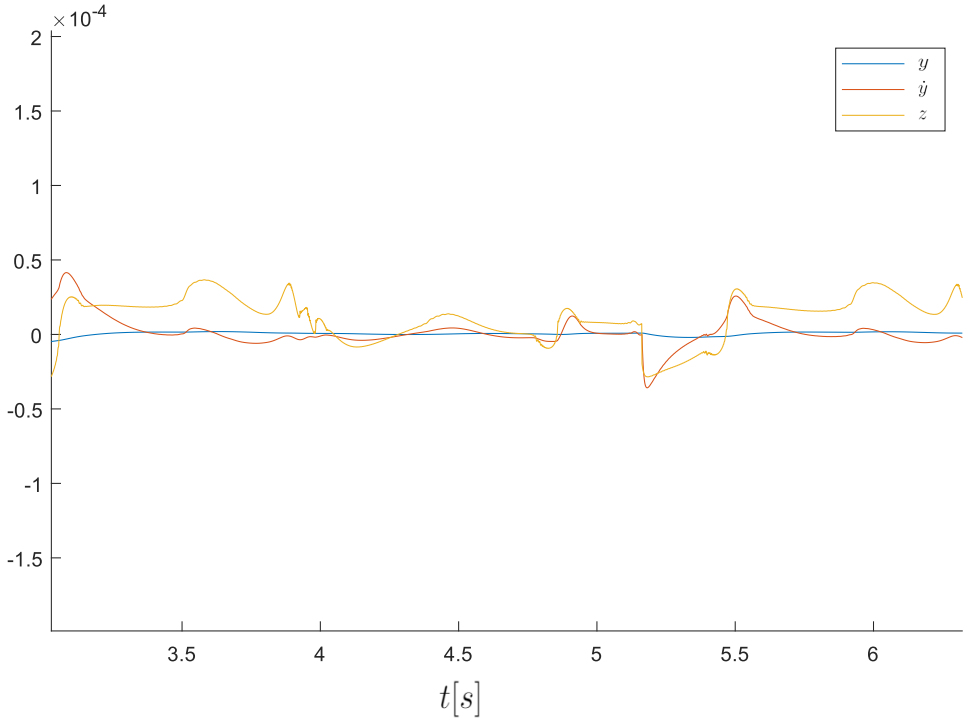


Figure 5.3.4: Zoomed in version of Figure 5.3.3. Note the axis values.

Perturbed System Parameters

As with everything in the real world, the values in Table 4.2 are measured using physical instruments on physical objects. This naturally implies there are some uncertainties associated with the precision of the method. This is one of the main motivations for implementing a sliding mode control, namely its ability to reject uncertainties and imprecision in the modeling. To test the controller for robustness, the system parameters in Table 4.2 are scaled up by 10% and the system is simulated. The results can be seen in Figure 5.3.5 - 5.3.7.

Figure 5.3.5 shows that the output trajectory do converge towards the nominal trajectory, however with a considerable larger error as with the nominal case. It seems that the controller seems to struggle the most around the local maxima and minima of the trajectory. To maintain readability, the simulation was stopped after two periods, and it should be mentioned that the output trajectory did not converge any closer than what can be seen in the plot. In Figure 5.3.6 the control torque also reach a somewhat poorer stationary error than with the nominal case. However, the chattering effect is considerable lesser in this case. One explanation for this may be that the stationary error of the perturbed system is larger than the boundary in which chattering occurs.

As for the transverse coordinates which can be seen in Figure 5.3.7, the first dip for all three coordinates seems to match that of the nominal case, before converging to a periodic behavior. The error in the perturbed simulation is noticeable larger, especially for z, which have an maximum error approximately 4000 times larger than that of the nominal case.

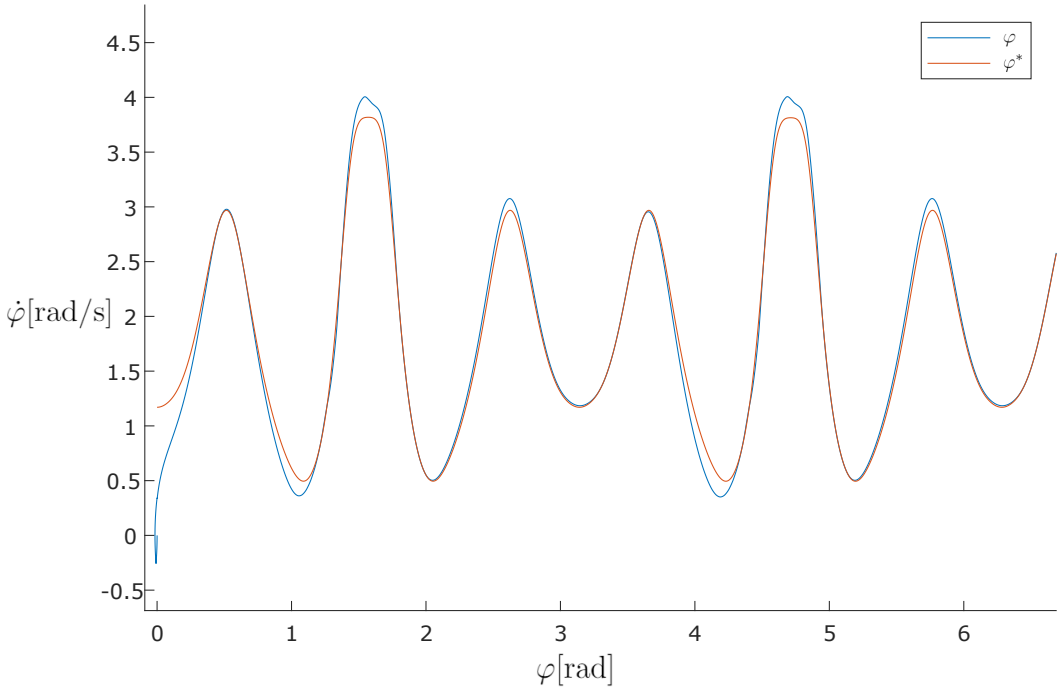


Figure 5.3.5: Phase portrait of the nominal trajectory vs. the sliding mode control output trajectory, simulated with system parameters scaled up by 10%.

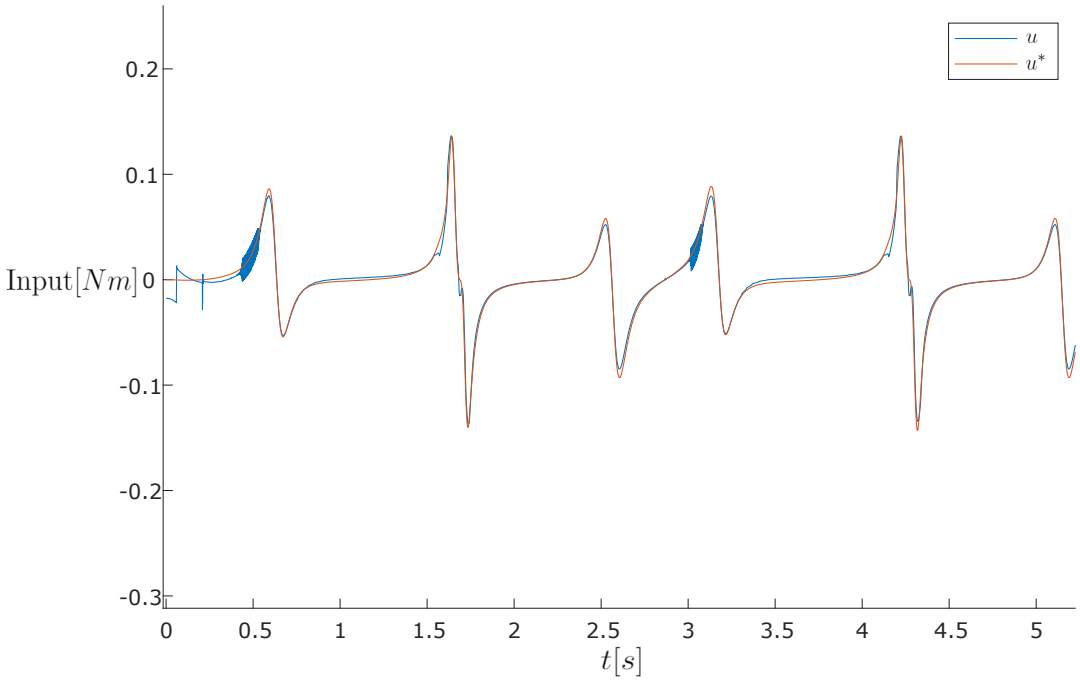


Figure 5.3.6: Nominal input vs. actual input torque of the sliding mode controller, simulated with system parameters scaled up by 10%.

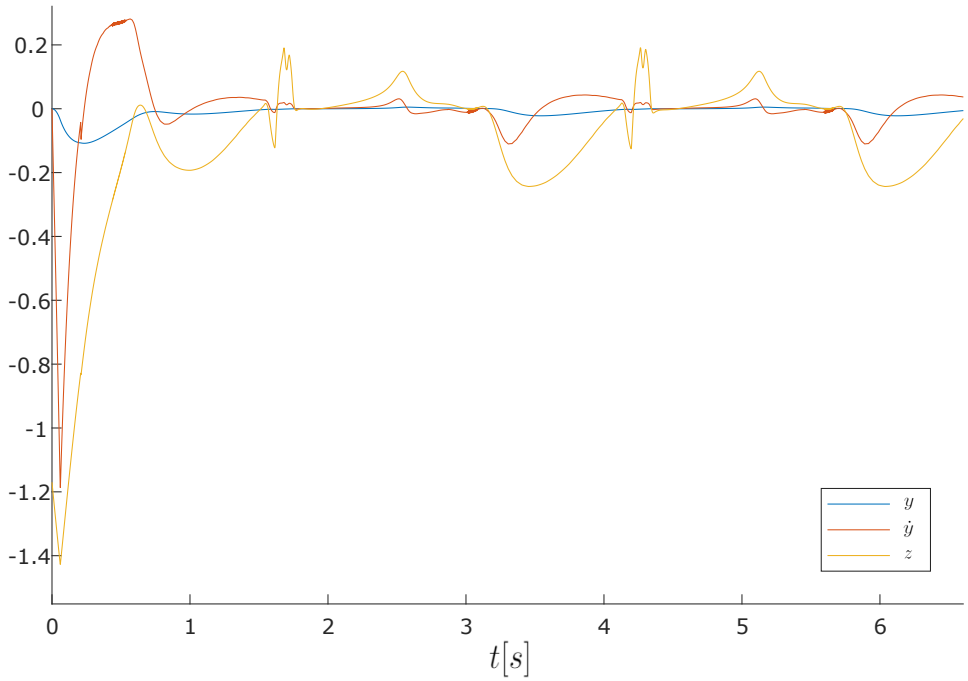


Figure 5.3.7: The transverse coordinates under the sliding mode controller, simulated with system parameters scaled up by 10%.

The system was then simulated again with system parameters now being scaled down by 10%. The results did not seem to visually change from the plots shown in Figure 5.3.5 - 5.3.7, and will therefor be omitted.

SMC Design II

In this section, a switching function for the SMC will be designed based on the subspace of the linearized system. By the Grobman-Hartman theorem, it is well known that the behavior of a nonlinear system around one of its equilibirum points can be analyzed by the linearized system around that same point. One important note is that the locally invariant manifolds to the original nonlinear system, corresponds to the invariant subspace of linearized system [9]. Thus, by designing a switching function such that a sliding mode controller converge the system to a set of stable invariant subspace of the linearized system, the behavior of the nonlinear system should comply with this stable behavior.

Finding an FL Factorization

In order to design a switching function based on the invariant subspace from the theory, an FL factorization of the STM in Equation 2.5.31 was needed. To do so, a boundary value problem (BVP) was solved. The problem formulation was set up in the same manner as in [18] where it was shown that $L(\cdot)$ satisfy the first order matrix differential equation

$$\forall t \in \mathbb{R}^+, \quad \dot{\mathbf{L}}(t) = \mathbf{A}(t)\mathbf{L}(t) - \mathbf{L}(t)\mathbf{F}, \tag{5.3.6}$$

while F is constant, meaning an initial guess was needed. To obtain this, Equation 2.5.25 was integrated and inserted into Equation A.2.1, yielding the F matrix

$$F \approx \begin{bmatrix} 4.9898 & -1.4548 & -2.4611 \\ 4.4042 & -1.1158 & -2.2925 \\ -13.1731 & 3.6652 & 6.6227 \end{bmatrix}, \tag{5.3.7}$$

An initial guess for L was also needed, and was set to

$$\mathbf{L}_0 = \sin(x) \cdot \mathbf{I}_3 + \mathbf{I}_3, \tag{5.3.8}$$

for some x. In order to simplify the calculations, the following boundary conditions was added to the system

$$\mathbf{L}(0) = \mathbf{L}(s_T) = \mathbf{I}_3 \tag{5.3.9}$$

The boundary value problem was then solved using the MATLAB function *BVP5C* ([34]) with the initial guesses, the function to solve and the boundary conditions were defined as above.

Designing a Switching Function

To find a real invariant subspace of co-dimension one of F, the corresponding eigenvalues and eigenvectors was found

$$\begin{aligned}(\lambda_1, v_1) &\approx \left(-2.1155e^{-6}, \quad \begin{bmatrix} 0.70155 \\ 0.71454 \\ 1 \end{bmatrix} \right), \quad (\lambda_2, v_2) \approx \left(0.24164, \quad \begin{bmatrix} 0.14916 \\ -1.20486 \\ 1 \end{bmatrix} \right), \\ (\lambda_3, v_3) &\approx \left(10.25505, \quad \begin{bmatrix} -0.37191 \\ -0.34566 \\ 1 \end{bmatrix} \right),\end{aligned} \tag{5.3.10}$$

yielding the three subspaces

$$\text{span}\{v_1, v_2\}, \;\; \text{span}\{v_1, v_3\}, \;\; \text{span}\{v_2, v_3\}$$

The task is then to design $\hat{S}$ such that $\hat{S}v_i \equiv 0$ will be used in the final matrix function for the switching function; $S_\perp(s) = \hat{S}L^{-1}(s) = [S_y, S_{\dot{y}}, S_z]$. Note that the span defined by the vectors above must be chosen so that the non-zero determinant condition from Theorem 2.5.3 is satisfied.

Implementation and Results

Having already found the approximating stabilizing solution H to the PRDE, the optimal LQR controller $u_\perp^* = K_\perp(s)$ was found using Equation 2.5.5, such that

$$K_\perp(s) = -R^{-1}(s)B_\perp(s)H(s) \tag{5.3.11}$$

The elements of $K_\perp = [K_y \;\; K_{\dot{y}} \;\; K_w]$ are shown in Figure 5.3.8.

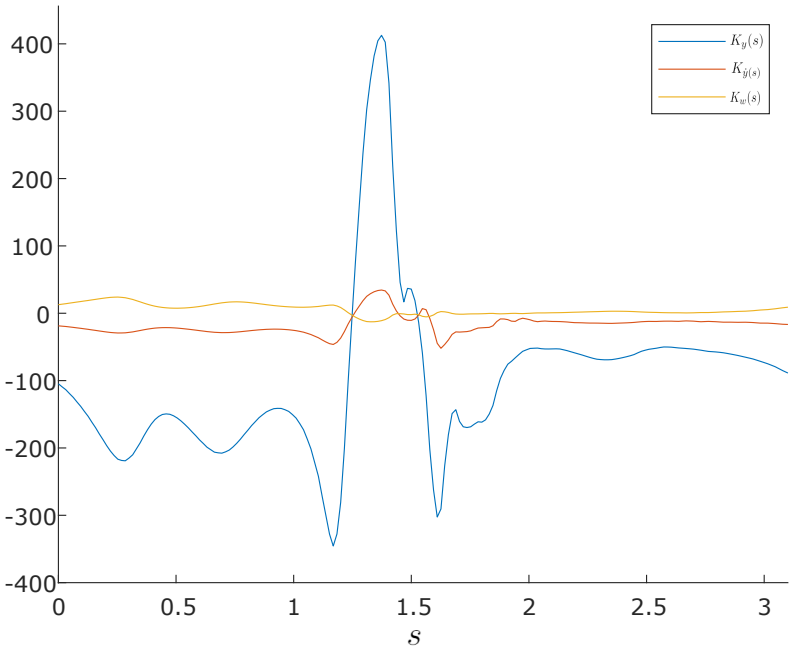


Figure 5.3.8: The nominal feedback matrix $K_\perp(s)$ given by Equation 5.3.11.

The vector $\hat{S}$ was then calculated based on the subspace span$\{v_2, v_3\}$, and used together with the periodic $L(\cdot)$ matrix to obtain $S_\perp(s) = \hat{S}L^{-1}(s) = [S_y, S_{\dot{y}}, S_w]$. Figure 5.3.9 shows the matrix function $S_\perp(s)$, while Figure 5.3.10 shows $S_\perp(s)B_\perp(s)$.

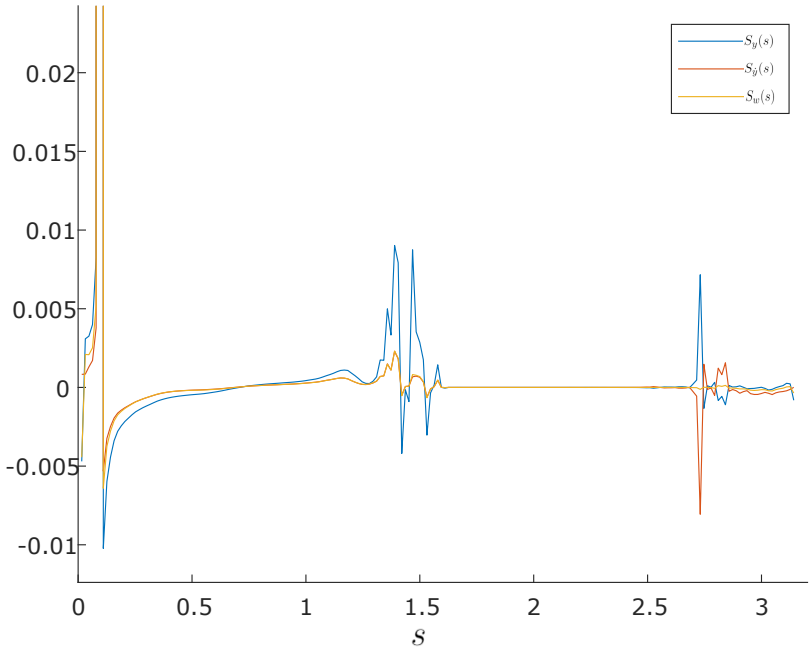


Figure 5.3.9: The elements of the switching function $S_\perp(s)$

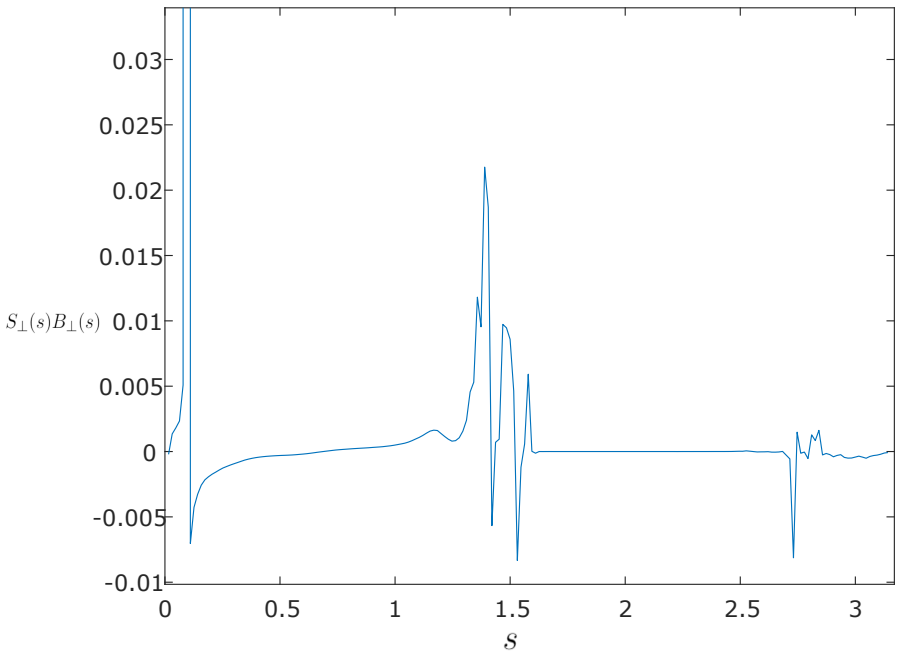


Figure 5.3.10: The matrix function $S_\perp(s)B_\perp(s)$

The first observation that can be made from Figure 5.3.10 is that $S_\perp(s)B_\perp(s)$ is not separated from zero, meaning the conditions of Theorem 2.5.3 are not satisfied. The method was then repeated for the other two subspaces span$\{v_1, v_2\}$ and span$\{v_1, v_3\}$ which unfortunately did not change the outcome, suggesting that there was something wrong in the implementation.

As mentioned as a note in the origin of Theorem 2.5.3 from [9], it is assumed that the FL factorization is s_T periodic, which is not guaranteed to exist, rather than a $2s_T$ periodic factorization, which is guaranteed. Simulations showed that this was the case indeed, and $L(s)$ was 2π periodic. Motivated to solve the problem, an effort was put into obtaining the *Yakubovich* matrix Y, which may be used when encountering the problem above [18].

The new boundary value problem was identical to the previous, however where the following boundary condition was swapped

$$\mathbf{L}(s_T) = \mathbf{I}_3 \quad \Rightarrow \quad \mathbf{L}(s_T) = \mathbf{Y}^{-1},$$

Meaning that the more correct notation for the condition in Equation 5.3.9 would be $\mathbf{L}(2s_T) = \mathbf{I}_3$. However, the solution of the new boundary value problem gave even poorer results than the BVP suggested at first. As can be seen in Figure 5.3.9 and Figure 5.3.10, there seems to be an asymptotic behavior at first. This behavior repeated itself several times in the newly suggested BVP, and the matrix function $S_\perp B_\perp$ still was not seperated from zero, meaning the necessary conditions of Theorem 2.5.3 was still not satisfied. Considering the amount of time and effort that had been put into this problem, both theoretically and practical, no further effort was put into trying to find a solution.

5.4 Discussion

In this chapter, two different sliding mode controller designs was presented and simulated on the Butterfly Robot. The results are discussed here.

SMC Design I

The unstable feedforward controller from section 5.1 was complemented with a stabilizing term, which is based on a control law that ensures invariance of the sliding condition in Equation 2.5.16. As mentioned in the first part of the results, this did not result in any stabilizing controller for the system, motivating for further work. Considering that the stabilizing solution of the PRDE have previously been used in a stabilizing term similarly to the one in Equation 2.5.5, the sliding surface in Equation 5.3.1 was complemented with the approximating stabilizing solution H^+. The resulting controller managed to stabilize the system when it was subjected to the nominal system parameters used in the modeling. When the parameters were scaled up/down by as much as 10%, the controller was still able to track the nominal trajectory, although a considerable larger error from the nominal curve occurred. Although scaling all the system parameters up by 10% can be considered a rather extreme case, considering the accuracy of modern instruments, it would still be of interest to obtain a better convergence than what was obtained. In a physical experiments the smallest error could send the ball flying of the frame. And considering that there are a lot of neglected forces like friction, air resistance and simplifications in the shape of both the ball and the frame, it would be reasonable to say the perturbation in a real world may be challenging for this controller. It is possible that the simple design may be a little too simple for such a complicated system.

One motivation for designing a control law that did not depend on the Ricatti equation was due to the uncertainties that comes along with it. Solving the PRDE for a highly unstable system may cause trouble down the line, as the solution found in the beginning will be used throughout the run. Thus, any imprecision in the modeling will be used in the stabilizing controller later. However, the upside of using H^+ in the manner being done in this paper, is that instead of being directly used in the control law (as in [2]), it is rather be used as a term in the Lyapunov function, thus having the possibility that the controller will still be stabilizing even if the Lyapunov function is wrong.

SMC Design II

Although a substantial effort was put into both understanding and implementation of the invariant-based SMC, the method did not yield any significant results, other than the fact that the problem remains unsolved. It would be reasonable to say that this do not prove that such a method do not work for the Butterfly Robot, but rather that it was not implemented in the

right way. Having encountered several struggles regarding both way too long computational time, and the encountering of singular Jacobians during the solving of the boundary value problem, it may indicate that (one of) the problem(s) originated already here. Considering the complexity of the robotic system as well, the task of troubleshooting quickly became non-trivial. When it comes to the encountering of the singular Jacobian, a quick search in literature yields several possible things that may have been the problem. The initial guess may not be sufficient to produce a convergence, which often seems to be the case for complex systems. Although the system was initiated with several different initial guesses, it is important to consider that the nonlinear solver is just as effective as its initial guess, making the BVP feel more like art than science. Another reason may have been that the solver was able to converge for some iterations, but at some point became unable to find the next guess needed for a solution.
As was mentioned in the introduction, due to the approximation of the system parameters, any solution should in fact hold for a family of different systems in order to account for the possible variations in the parameters. This will add another level of complexity to the problem, as one solution may only work for a particular set of parameters. To the authors knowledge, no such method has previously been implemented on the Butterfly Robot which is without a doubt a complicated system. This raises the question of how complicated a solution may be. It would therefor be reasonable to suggest that further work is needed in order to evaluate such a method on the robotic system.

It is worth mentioning that the stabilizing methods presented both here and in [2] relies on the solution of the Ricatti equation. Consider that one is able to design a controller and bring the linearized system to a closed loop in terms of the transverse coordinates. The stability of this is equivalent with the fact that there exist a solution to the Ricatti differential equation such that the Monodromy matrix has eigenvalues within the unit circle. Thus, any stabilizing controller design will have to solve the RDE. Recognizing this indeed is a challenging task motivates for the development of alternative solutions.

Chapter 6

Conclusion and Further Recommendations

6.1 Conclusion

This paper aimed to develop a stabilizing controller based on the sliding mode control synthesis for a system experiencing perturbations in system parameters. The mechanical system to which the controller was subjected is currently known as the Butterfly Robot. Given that such a robotic system is underactuated, meaning that the system has more degrees of freedom than it has control inputs, it will be unstable by nature. To overcome such a problem, several essential steps were made during the deriving of the generalized coordinates, as well as in the motion planning section. To find a feasible motion for the system, a geometric relation known as a virtual holonomic constraint was added to couple the generalized coordinates together using a scalar function known as a motion generator. The geometric relation was then held invariant by a feedback controller, essentially lowering the complexity to just one configuration variable. The dynamics of the motion generator, known as the reduced dynamics, could then be used as a noble candidate for the nominal trajectory.

The main contribution of this paper was to introduce two different designs of a sliding mode controller to stabilize the transverse linearized dynamics of the Butterfly Robot. Both methods exploited the well-known fact that the stabilization of a set of transverse coordinates, whose origin corresponds to the nominal trajectory, corresponds to the stabilization of the nonlinear system. As a part of the stabilizing terms, an approximation of the stabilizing solution of the periodic Ricatti differential equation was found through semi-definite programming, where the Ricatti differential equation was solved by writing the differential equation as a linear matrix inequality by taking the Schur complement.

The first design was based on a notation simplification such that an n-th order problem could be replaced by a first-order problem. Lyapunov theory was then used to design a control law that provides finite-time convergence towards a so-called sliding surface. This surface could then be designed to match the nominal trajectory given by the reduced dynamics. As the

transverse coordinates by nature are zero on the nominal trajectory, they were directly used in the sliding surface.

The second design is built on the equivalence between the stable invariant manifolds of the nonlinear system and the stable invariant subspace of the first-order approximation system. Here, a time-varying subspace of the linearized system was designed before its annihilator was used in the switching function for the sliding mode controller. To construct such a subspace, a real Floquet-Lyapunov transformation of the state transition matrix of the transverse linearized system was to be found. Here, the F factor was found through integration of the system, and the L factor was found by solving a boundary value problem.

Both controllers were implemented and numerically simulated using MATLAB and Simulink. The first design was able to converge the system trajectory towards the nominal trajectory, with an absolute maximum error of approximately $0.5e^{-4}$ in the transverse coordinates using the nominal system parameters. The system then experienced a perturbation of $\pm 10\%$ in the system parameters, resulting in a system that still converged towards the nominal trajectory, however with an absolute maximum error of approximately 0.2 in the transverse coordinates. The second design was not able to properly stabilize the system. Considering that the constructed switching function failed to satisfy the necessary conditions, a great effort was put into overcoming the problem without success. Due to the complexity of the system as well, troubleshooting became non-trivial. It should be mentioned that a real FL transformation where L was π-periodic was not found, suggesting the problem originated in solving the boundary value problem.
As the Butterfly Robot is a practical example with physically measured system parameters, any solutions to the stabilization problem should hold for a family of different systems to account for the variations in parameters. As the complexity of such a solution is questionable, further work is needed.

6.2 Recommendations for Further Work

In this section, some recommendations for further work is presented.

Construction of a switching function based on the invariant subspace
Considering that the work presented in this paper did not find any real invariant subspace of the linearized transverse system that satisfied all the necessary conditions, this would be a natural next step. The switching function could then be designed based on its annihilator, and later used in a sliding mode controller to evaluate such a controller using numerical simulations.

Implementation on the physical system

Clearly mathematics is not useful unless it can be implemented in the real world. Considering that the SMC in design 1 was able to converge towards the nominal trajectory, even with a perturbation of $\pm 10\%$ in the system parameters, it would be interesting to see how the controller would perform on the physical robot where there are real perturbations. Given that some of the model assumptions made in this paper are not valid in the real world, the real robustness of the controller could be tested. This also applies for the second design, if one were to be able to construct such a switching function.

Bibliography

[1] Fabio Ruggiero, Vincenzo Lippiello, Bruno Siciliano, (2018) *Nonprehensile Dynamic Manipulation: A Survey*, IEEE Robotics and Automation Letters 3, DOI: 10.1109/LRA.2018.2801939

[2] Maksim Surov, Anton Shiriaev, Leonid Freidovich, Sergei Gusev, Leonid Paramonov *Case study in non-prehensile manipulation: planning and orbital stabilization of one-directional rollings for the "Butterfly" robot*, 2015 IEEE International Conference on Robotics and Automation (ICRA)

[3] K.M. Lynch, N. Shiroma, H. Arai and K. Tanie, *The Roles of Shape and Motion in Dynamic Manipulation: The Butterfly Example*, IEEE International Conference on Robotics and Automation, 1998

[4] Christian Fredrik Sætre, (2016), *Stable Gaits for an Underactuated Compass Biped Robot with a Torso*, Master Thesis, Norwegian University of Science and Technology, Department of Engineering Cybernetics

[5] A. Shiriaev, J.W. Perram, C. Canudas-de Wit (2005) *Constructive tool for orbital stabilization of underactuated nonlinear systems: Virtual constraints approach*, Automatic Control, IEEE Transactions

[6] Anton S. Shiriaev, Leonid B. Freidovich, Sergei V. Gusev (2010) *Transverse Linearization for Controlled Mechanical Systems With Several Passive Degrees of Freedom*, IEEE Transactions on Automatic Control, DOI: 10.1109/TAC.2010.2042000

[7] Sergei V. Gusev, Anton S. Shiriaev and Leonid B. Freidovich, (2016), *SDP-based approximation of stabilising solutions for periodic matrix Riccati differential equations*, International Journal of Control, 89:7, 1396-1405, DOI: 10.1080/00207179.2015.1131850

[8] İhsan Ömür Bucak, *An In-Depth Analysis of Sliding Mode Control and Its Application to Robotics*, Intechopen, DOI: 10.5772/inte-

chopen.93027 https://www.intechopen.com/books/automation-and-control/an-in-depth-analysis-of-sliding-mode-control-and-its-application-to-robotics

[9] Christian Fredrik Sætre, Anton Shiriaev, Leonid Freidovich, Sergei V. Gusev, Leonid Fridman, (2020) *Robust Orbital Stabilization: A Floquet Theory-based approach*, arXiv preprint arXiv:2011.13674

[10] H.R.Harrison, T.Nettleton, (1997) *Advanced Engineering Dynamics*, Elsevier Ltd., ISBN: 978-0-340-64571-0

[11] *Encyclopedia of Physical Science and Technology*, Chapter 13 2001 Elsevier Science Ltd., ISBN 978-0-12-227410-7

[12] Guangren Duan (2021), *High-order fully actuated system approaches: Part I. Models and basic procedure*, International Journal of Systems Science, 52:2, 422-435, DOI: 10.1080/00207721.2020.1829167

[13] Edward Walter Kamen (2010) *Fundamentals of Linear Time-Varying Systems*, The Control Systems Handbook, Second Edition, CRC Press, DOI: 10.1201/b10384-5

[14] Jiang B, Gao C, Kao Y, Liu Z, (2016), *Sliding mode control of Markovian jump systems with incomplete information on time-varying delays and transition rates*, Applied Mathematics and Computation, DOI: 10.1016/j.amc.2016.05.038

[15] Leu VQ, Choi HH, Jung JW (2012), *LMI-based sliding mode speed tracking control design for surface-mounted permanent magnet synchronous motors*, Journal of Electrical Engineering and Technology, DOI: 10.5370/JEET.2012.7.4.513

[16] Huynh VV, Tsai YW, Duc PV (2015), *Adaptive output feedback sliding mode control for complex interconnected time-delay systems*, Mathematical Problems in Engineering, DOI: 10.1155/2015/239584

[17] Oskar Rømyhr Lund, (2018) *Case study research: the Butterfly Robot*, Master Thesis, Norwegian University of Science and Technology, Department of Engineering Cybernetics

[18] Pierre Montagnier, Raymond J. Spiteri, Jorge Angeles, (2004) *The control of linear time-periodic systems using Floquet-Lyapunov theory*International Journal of Control, March 2004, DOI: 10.1080/00207170410001667477

[19] Bogusław Radziszewski, Krzysztof Gajewski *On orbital stability of impact motion*, Nonlinear Vibration Problems, (1991)

[20] Manfredi Maggiore, Luca Consolini, *Virtual Holonomic Constraints for Euler–Lagrange Systems*, IEEE XPLORE, DOI: 10.1109/TAC.2012.2215538

[21] Leonid B. Freydovich, Sergei V. Gusev, *Method, system and computer program for controlling dynamic manipulations by a robot*, PRV, Swedish Patent Database

[22] Marco Capotondi, Giulio Turrisi, Claudio Gaz, Valerio Modugno, Giuseppe Oriolo, Alessandro De Luca, *Learning Feedback Linearization Control Without Torque Measurements*, Dipartimento di Informatica Automatica e Gestionale, Sapienza Universita di Roma.

[23] Anton Shiriaev, TTK6 Lecture 2, *Tools for Analysis of Second Order Systems*, August 31, 2020, NTNU Trondheim

[24] Ian R. Manchester, *Transverse Dynamics and Regions of Stability for Nonlinear Hybrid Limit Cycles*, CSAIL, Massachusetts Institute of Technology

[25] J. A. Acosta, M. Lopez-Martinez, (2005), *Constructive feedback linearization of underactuated mechanical systems with 2-DOF*, 44th IEEE Conference on Decision and Control, and the European Control Conference

[26] Anton S. Shiriaev, Leonid B. Freidovich, Sergei V. Gusev, (2009), *Transverse Linearization for Mechanical Systems with Several Passive Degrees of Freedom with Applications to Orbital Stabilization*, American Control Conference

[27] Nestruev, Jet, (2002), *Smooth manifolds and observables*, Berlin: Springer, ISBN 0-387-95543-7

[28] Mohamed Harmouche, (2013) *Contribution to the theory of higher order sliding mode control and the control of underactuated mechanical systems*, Other, Université de Technologie de Belfort-Montbeliard, English. NNT : 2013BELF0214. tel-01503945

[29] Zhang, Fuzhen, (2005), *The Schur Complement and Its Applications*, Numerical Methods and Algorithms, 4, Springer. doi:10.1007/b105056, ISBN 0-387-24271-6

[30] V. Yakubovich, V. Starzhinskii, (1975), *Linear Differential Equations with Periodic Coefficients*, John Wiley Sons

[31] Ruben Bjarnesen Brecke, (2020) *Developing Dynamics and Motion Planning for the Butterfly Robot*, Pre-project, Norwegian University of Science and Technology, Department of Engineering Cybernetics

[32] Gustav Omberg Often, (2020) *Non-prehensile Manipulation, the Butterfly Robot*, Preproject, Norwegian University of Science and Technology, Department of Engineering Cybernetics

[33] K. C. Toh, R. H. Tütüncü, M. J. Todd *SDPT3 - a MATLAB software package for semidefinite-quadratic-linear programming*, `https://www.math.cmu.edu/~reha/sdpt3.html`

[34] Mathworks *bvp5c*, `https://se.mathworks.com/help/matlab/ref/bvp5c.html`

[35] Github *Uploaded code*, `https://github.com/rubenbb/TTK4900`

Appendices

Appendix A

Various Calculations and Equations

A.1 The Christoffel Symbols

Using Equation 2.2.5 from Section 2.2 in the theory, each element of the $C(\cdot)$ matrix can be easily computed

$$
\begin{aligned}
c_{111} &= \frac{1}{2}\left[\frac{\partial m_{11}}{\partial \theta}\right] = 0 \\
c_{112} &= \frac{1}{2}\left[2 \cdot \frac{\partial m_{21}}{\partial \theta} - \frac{\partial m_{11}}{\partial \varphi}\right] = -m_b s' \vec{\rho} \cdot \vec{\tau} \\
c_{121} &= \frac{1}{2}\left[\frac{\partial m_{11}}{\partial \varphi}\right] = m_b s' \vec{\rho} \cdot \vec{\tau} \\
c_{122} &= \frac{1}{2}\left[\frac{\partial m_{22}}{\partial \theta}\right] = 0 \\
c_{211} &= \frac{1}{2}\left[\frac{\partial m_{11}}{\partial \varphi}\right] = m_b s' \vec{\rho} \cdot \vec{\tau} \\
c_{212} &= \frac{1}{2}\left[\frac{\partial m_{22}}{\partial \theta}\right] = 0 \\
c_{221} &= \frac{1}{2}\left[2 \cdot \frac{\partial m_{12}}{\partial \varphi} - \frac{\partial m_{22}}{\partial \theta}\right] = \left(m_b \hat{k} \cdot (\vec{\rho} \times \vec{\tau}) + J_b p\right) s'' + m_b \hat{k} \cdot (\vec{\rho} \times \vec{\kappa}) s'^2 \\
c_{222} &= \frac{1}{2}\left[\frac{\partial m_{22}}{\partial \varphi}\right] = (m_b + J_b p^2) s' s''
\end{aligned}
\tag{A.1.1}
$$

which will yield the four elements

$$\begin{aligned}
c_{11} &= c_{111}\dot{\theta} + c_{211}\dot{\varphi} = m_b s'\vec{\rho}\cdot\vec{\tau}\dot{\varphi} \\
c_{12} &= c_{121}\dot{\theta} + c_{221}\dot{\varphi} = m_b s'\vec{\rho}\cdot\vec{\tau}\dot{\theta} + \left(\left(m_b\hat{k}\cdot(\vec{\rho}\times\vec{\tau}) + J_b p\right)s'' + m_b\hat{k}\cdot(\vec{\rho}\times\vec{\kappa})s'^2\right)\dot{\varphi} \\
c_{21} &= c_{112}\dot{\theta} + c_{212}\dot{\varphi} = -m_b s'\vec{\rho}\cdot\vec{\tau}\dot{\theta} \\
c_{22} &= c_{122}\dot{\theta} + c_{222}\dot{\varphi} = (m_b + J_b p^2)s's''\dot{\varphi}
\end{aligned} \tag{A.1.2}$$

which gives the final form

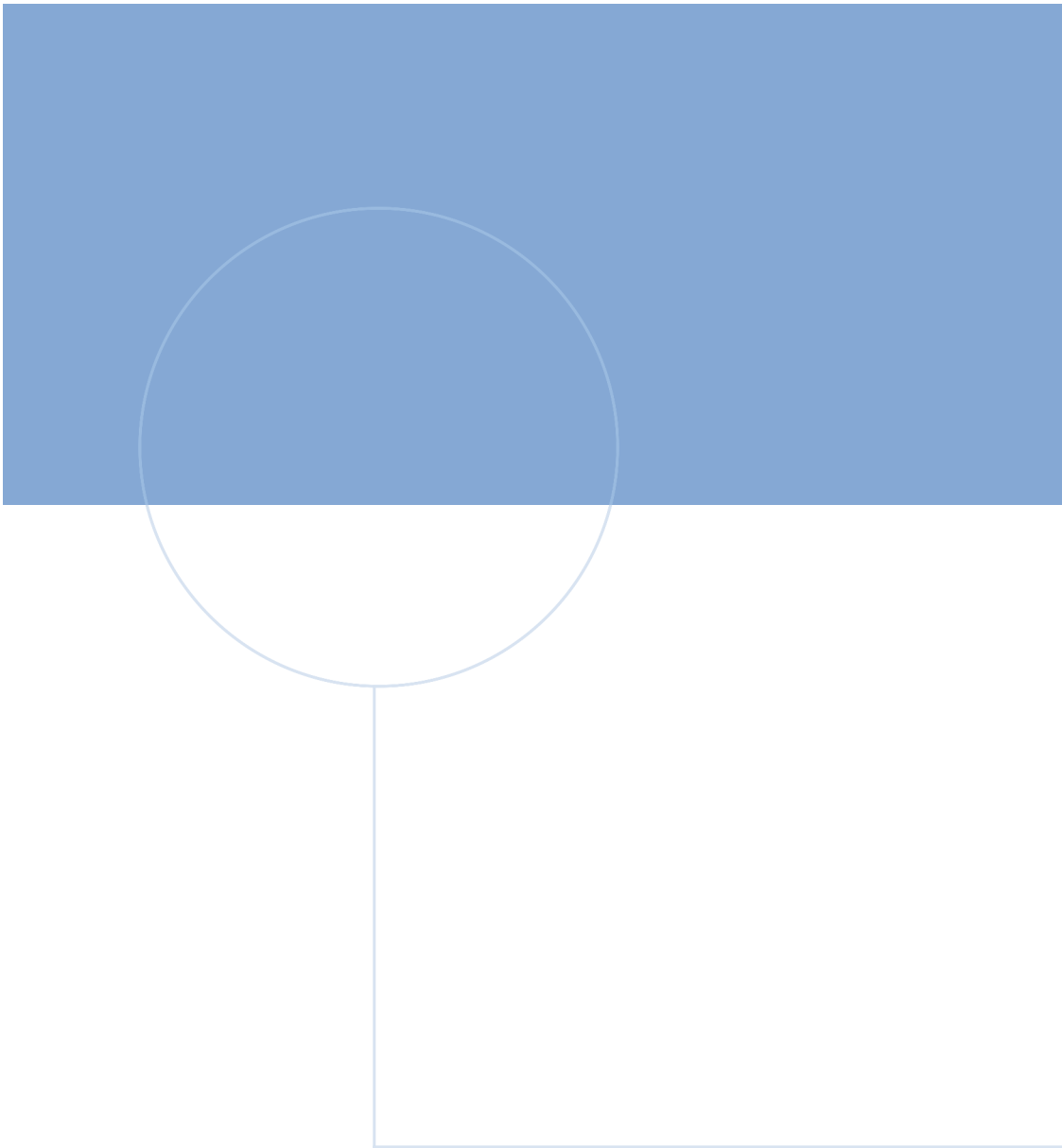
$$C(q,\dot{q}) = \begin{bmatrix} m_b s'\vec{\rho}\cdot\vec{\tau}\dot{\varphi} & m_b s'\vec{\rho}\cdot\vec{\tau}\dot{\theta} + \left(\left(m_b\hat{k}\cdot(\vec{\rho}\times\vec{\tau}) + J_b p\right)s'' + m_b\hat{k}\cdot(\vec{\rho}\times\vec{\kappa})s'^2\right)\dot{\varphi} \\ -m_b s'\vec{\rho}\cdot\vec{\tau}\dot{\theta} & (m_b + J_b p^2)s's''\dot{\varphi} \end{bmatrix} \tag{A.1.3}$$

A.2 Constructing a Floquet-Lyapunov Factorization

There are several methods available for computing a real FL factorization for LTP systems. These are mainly categorized in direct and indirect methods. As for the direct method, one can use knowledge of the state transition matrix. In [9] it was shown that the matrix F can be calculated using the Monodromy matrix;

$$\mathcal{M}_A^2 = \bar{\mathcal{M}}_A\mathcal{M}_A = e^{TB}e^{\bar{T}\bar{B}} = e^{2TF}, \tag{A.2.1}$$

i.e. $F = (B + \bar{B})/2$, where B may be complex.



NTNU

Norwegian University of
Science and Technology