NTNU
Norwegian University of
Science and Technology

Stable Gaits for an Underactuated Compass Biped Robot with a Torso

Trajectory Planning and Control Design using
the Virtual Holonomic Constraints Approach

Christian Fredrik Sætre

Master of Science in Cybernetics and Robotics
Submission date: June 2016
Supervisor: Anton Shiriaev, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Abstract

The versatility of two-legged walking can make bipedal robots ideal for operations in rough terrain and environments designed for human beings. In order to make such robots able to operate over long periods of time and handle unexpected external disturbances, it is crucial that their gaits are energy efficient and stable with respect to perturbations from the nominal locomotion.

This thesis considers gait planing and the development of stabilizing controllers for a planar compass biped with a torso using the virtual holonomic constraints approach. The biped considered has three degrees of freedom and two actuators, and is thus underactuated by a degree of one. It is further considered as a hybrid dynamical system with a continuous phase and a discrete phase arising from the sudden change in velocities occurring from the impact of the swing leg with the ground at the end of a step.

Feasible gaits for the biped are found by assuming that a set of geometrical relations, called virtual holonomic constraints, between the configurations variables and a scalar variable, referred to as the motion generator, are kept invariant by feedback control. This constrains the evolution of the dynamical system to a two-dimensional submanifold of the state space, describing the evolution of the motion generator. For a given motion generator, virtual holonomic constraints which describes a feasible gait are then found by solving a non-linear optimization problem which minimizes a desired cost criteria. It is shown that highly energy efficient gaits with low torque requirements can be found with the method presented. Moreover, it is shown that the choice of the motion generator heavily influences which feasible gaits that can be found, and that obvious candidates for the motion generator, as the angle of the stance leg, can greatly restrict the search space.

Stabilizing controllers are developed by stabilizing a linear, time-varying system corresponding to the linearization of a set of transverse coordinates which are exactly zero on the nominal trajectory. Two different types of controllers based on transverse linearization are developed. The first uses the angle of the stance leg as the motion generator and five transverse coordinates, which is minimum number of independent transverse coordinates. The second method approximates the motion generator through a projection from the configuration variables of the biped, such that it can be used to stabilize gaits where the motion generator is not directly known as a function of the configuration. Transverse linearization is then carried out for six excessive transverse coordinates. For both methods, a stabilizing controller is then developed which exponentially stabilizes the linear time-varying system corresponding to the linearized transverse dynamics by solving the Riccati differential equation with one jump. This is done by transforming the Riccati differential equation to a linear matrix inequality and solving a semi-definite programming problem. The found controllers then renders the trajectory of the non-linear system locally, exponentially orbitally stable. It is shown in numerical simulations that both control methods can stabilize gaits with respect to small perturbations from the nominal motion and changes in the system parameters from those used in the development of the controllers.

Sammendrag

(Norwegian translation of the Abstract)
Allsidigheten ved tobeint gange kan gjøre tobeinte roboter ideelle for drift i vanskelig terreng og i omgivelser som er designet for mennesker. For å gjøre slike roboter i stand til å operere over lange tidsperioder og håndtere uventet ytre forstyrrelse, så er det avgjørende at deres ganglag er energieffektive og stabile med hensyn til avvik fra det ønskede ganglaget.

Denne avhandlingen omhandler planlegging av ganglag og utviklingen av stabiliserende kontrollere for en tobeint robot med overkropp som er betraktet i det vertikale plan ved hjelp av en metode som bruker virtuelle holonomiske begrensninger. Den tobeinte roboten har tre frihetsgrader og to aktuatorer, og er dermed et underaktuert system av èn grad. Roboten er videre betraktet som et hybrid dynamisk system med en kontinuerlig fase og en diskret fase som er en følge av den plutselige endringen i hastigheter som oppstår fra kollisjonen mellom svingbenet til roboten og bakken ved slutten av et steg.

Mulige ganglang for roboten blir funnet ved å anta at et sett av geometriske relasjoner, kalt virtuelle holonomiske begrensninger, mellom konfigurasjonsvariablene og en skalar variabel kalt bevegelsesskaperen, kan opprettholdes ved bruk av tilbakekobling. Dette tvinger det dynamiske systemet til å utvikle seg på en todimensjonal submanifold av tilstandsrommet som beskriver utviklingen av bevegelsesskaperen. For en gitt bevegelsesskaper finner man så virtuelle holonomiske begrensninger som beskriver et ganglag ved å løse et ulineært optimaliseringsproblem som minimerer en ønsket konstnadsindeks. Det blir vist at den presenterte metoden kan finne svært energieffektive ganglag med lave krav for dreiemoment. Videre blir det vist at valget av bevegelsesskaperen kan ha store påvirkninger på hvilke ganglag som kan bli funnet, og at åpenbare kandidater, som vinkelen til standfoten, kan i stor grad begrense rommet av mulige ganglag.

Stabiliserende kontrollere blir utviklet ved å stabilisere et lineært, tidsvarierende system som tilsvarer linearisering av et sett av tverrgående koordinater som er nøyaktig lik null på det gitte ganglaget. To kontrollere blir utviklet som begge er basert på linearisering av tverrgående koordinater. Den første benytter vinkelen til standfoten som bevegelsesskaperen og bruker fem tverrgående koordinater, som er minimum antall uavhengige tverrgående koordinater. Den andre metoden tilnærmer bevegelsesskaperen gjennom en projeksjon fra konfigurasjonsvariablene til roboten, slik at den kan brukes til å stabilisere ganglag hvor bevegelsesskparen ikke er direkte kjent som en funksjon av konfigurasjonen. Linearisering av de tverrgående koordinatene blir så utført for seks tverrgående koordinater. For begge fremgangsmåter er en stabiliserende kontroller funnet som eksponentielt stabiliserer det lineære, tidsvarierende systemet som svarer til linearisering av det tverrgående dynamiske systemet ved å løse en Riccati differensialligningen med ett hopp. Dette gjøres ved å modellere differensialligningen som en lineær matrise-ulikhet og løse et semidefinitt optimaliseringsproblem. De derav funnede kontrollerne resulterer i at banen i tilstandsrommet som tilsvarer ganglaget for det ulineære system blir lokalt, eksponentielt orbitalt stabilt. Det blir vist i numeriske simuleringer at begge kontrollmetodene kan stabilisere ganglag for små avvik fra det ønskede ganglaget og for endringer i systemparametrene fra de som brukes i utviklingen av kontrollerene.

Preface

This thesis documents my work done on motion planning and stabilization of an underactuated, planar compass biped during the Spring semester of 2016. It also concludes my five year Master of Science program in Engineering Cybernetics at the Norwegian University of Science and Technology (NTNU).

Acknowledgements

I would like to thank my supervisor, Professor Anton Shiriaev, for all his guidance during the last two semesters and for taking the time to patiently explain difficult topics which gave me greater knowledge and appreciation for my field of study. I am very grateful for him giving me the opportunity to work on such an interesting and highly exiting topic as bipedal walking. I would also like to thank Dr. Leonid Paramonov and Torleif Anstensrud for all their help and advice during the semester.

Christian Fredrik Sætre
Trondheim, June 2016

Table of Contents

Abstract i

Sammendrag ii

Preface iii

Table of Contents vii

List of Tables ix

List of Figures ix

Abbreviations and Acronyms xi

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

2 Theoretical Framework and Preliminaries 5
- 2.1 Classical Mechanics and Equations of Motion . . . 5
  - 2.1.1 Generalized Coordinates and Holonomic Constraints . . . 5
  - 2.1.2 The Euler-Lagrange Equation . . . 6
  - 2.1.3 Fully Actuated vs Underactuated Systems . . . 8
- 2.2 Hybrid Mechanical Systems . . . 9
- 2.3 Periodic Trajectories, Limit Cycles and Stability . . . 10
  - 2.3.1 Orbital Stability . . . 12
  - 2.3.2 Poincaré first-return map and Poincarè Sections . . . 13
- 2.4 The Virtual Holonomic Constraints Approach . . . 15
  - 2.4.1 The Reduced Dynamics . . . 16
  - 2.4.2 Partial Feedback Linearization . . . 19
  - 2.4.3 Transverse Coordinates and Transverse Linearization . . . 22
  - 2.4.4 Extension to Impulsive Mechanical Systems with One Jump . . . 25

2.5 Stabilizing Linear Time-Varying Systems 29

3 Equations of Motion and Impact Models for the Three-Link Biped 35
3.1 Equations of Motion in the Swing Phase 36
3.2 Impact models 42
3.2.1 Algebraic Method with Newton's Model of Restitution 43
3.2.2 Conservation of Angular Momentum 45

4 Procedure for finding Periodic Gaits 49
4.1 Finding Feasible Gaits on the Full Model 50
4.2 Finding Feasible Gaits from the Reduced Dynamics 51
4.2.1 Finding the Reduced Dynamics 52
4.2.2 Choosing Synchronization Functions 54
4.2.3 Constraints 57
4.2.4 Summary of the optimization problem 62
4.3 Results 63
4.3.1 Full dynamical model 63
4.3.2 Reduced dynamics with stance leg as the motion generator 64
4.3.3 Reduced dynamics with an unknown motion generator 71
4.4 Discussion 76

5 Development of Stabilizing Controllers 77
5.1 Motion Generator as the Angle of the Stance Leg 77
5.1.1 Partial Feedback Linearization 78
5.1.2 Remaining Internal Dynamics 80
5.1.3 The Hybrid Linearized Transverse Dynamics 81
5.1.4 Construction of a Stabilizing Controller 83
5.1.5 Implementation and Results from Numerical Simulations 85
5.2 Motion Generator found through a Projection from the Configuration . . . 94
5.2.1 Excessive Transverse Coordinates 94
5.2.2 Transverse Linearization 95
5.2.3 Implementation and Results from Numerical Simulations 99
5.3 Discussion 108

6 Conclusion and Recommendations for Further Work 111
6.1 Conclusion 111
6.2 Recommendations for Further Work 112

Bibliography 114

A Various Proofs and Theorems 121
A.1 Proof of Theorem 2.4.1 121
A.2 Proof of Theorem 2.4.2 122
A.3 Hadamard's Lemma 123
A.4 Proof of Theorem 4.2.1 123
A.5 Proof of Proposition 4.2.2 125

B Full Function Expressions **127**

C Implementations and Additional Results **131**
C.1 The Integral Function using the Trapezoidal Rule 131
C.2 Stabilizing a Gait with the Motion Generator as the Angle of the Stane Leg using the Projection Method 132

List of Tables

4.1 Physcial parameters of the compass biped rounded to three decimal places. 63

List of Figures

1.1 The minimum cost of transport as function of body mass for a variety of robots, animals, and vehicles. 2

2.1 A bouncing ball for different values of the coefficient of restitution. 10
2.2 Illustration of a periodic trajectory Γ lying inside the annular region enclosed by the closed curves C_1 and C_2. 11
2.3 The van der Pol oscillator. 12
2.4 Illustration of a Poincarè map. 14
2.5 Illustration of a moving Poincarè section. 15
2.6 Moving Poincarè surfaces and switching surfaces. 27
2.7 Projection from a point $\Delta x(T_h) \in TS(T_h)$ to a point $\delta x_- \in T\Gamma_-$ along the direction of $\vec{n}(T_h)$. 29

3.1 Phases of the three-link biped with torso shown in the saggital plane. . . . 35

3.2 Masses, inertias and link lengths of the three-link biped. 37
3.3 Coordinate conventions. 38
3.4 Coordinate and frame convention using relative angles. 39

4.1 Gait found on the full dynamical model. 65
4.2 Energy of the biped for the gait found on the full dynamical model over two steps. 66
4.3 Gait found from the reduced dynamics with the motion generator chosen as $\theta = q_1$ and with the virtual holonomic constraints as 6th-order regular polynomials. 67
4.4 Energy of the biped for the gait found from the reduced dynamics over with 6th-order regular polynomials as virtual holonomic constraints. . . . 68
4.5 Gait found from the reduced dynamics with the motion generator chosen as $\theta = q_1$ and with the virtual holonomic constraints as 3rd-order trigonometric polynomials. 69
4.6 Gait found from the reduced dynamics with the motion generator chosen as $\theta = q_1$ and with the virtual holonomic constraints as 6th-order trigonometric polynomials. 70
4.7 Gait found from the reduced dynamics with the MG chosen as $\theta = s \in [0, 1]$ and with the VHCs as 6th-order Bezièr curves. 72
4.8 Energy of the biped for the gait found from the reduced dynamics with $\theta = s \in [0, 1]$ as the MG and 6th-order Bezièr curves as VHCs. 73
4.9 Energy of the biped for the gait found by minimizing (4.56) with $\theta = s \in [0, 1]$ as the MG and 6th-order Bezièr curves as VHCs. 73
4.10 Gait found by minimizing (4.56) with the MG chosen as $\theta = s \in [0, 1]$ and with the VHCs as 6th-order Bezièr curves. 74
4.11 Gait found by minimizing (4.56) for a step length 0f $10\,\mathrm{cm}$ and lean angle of $-0.0015\,\mathrm{rad}$ with the MG chosen as $\theta = s \in [0, 1]$ and with the VHCs as 6th-order Bezièr curves. 75

5.1 Relation between the projection operator and time over two steps. 86
5.2 The gait in open loop with only the PFL controller and unperturbed initial conditions. 87
5.3 The gait with unperturbed initial conditions and the controller generated with $\delta = 1$. 88
5.4 The gait with the parameters of the system scaled up by 10% compared to the design phase and a controller with $\delta = 2$. 89
5.5 The gait with the parameters of the system scaled down by 10% compared to the design phase and a controller with $\delta = 3$. 90
5.6 The gait with the initial perturbation $\dot{q}_1(t_0) = \dot{\theta}_*(t_0) + 0.015$ and $\delta = 3$. . 91
5.7 The gait with the initial perturbation $\dot{q}_1(t_0) = \dot{\theta}_*(t_0) - 0.06$ of the stance leg and controller with $\delta = 2$ with enabled saturation of the actuators at $\pm 5\,\mathrm{N\,m}$. 92

5.8 The gait with the initial conditions perturbed such that, measured in absolute angles, the swing and stance leg have zero velocity, while the torso is initialized with $\dot{q}_2^a(t_0) = -1.25\,\mathrm{rad\,s^{-1}}$. The controller was generated with $\delta = 1$ and saturation was set to $\pm 10\,\mathrm{N\,m}$. 93
5.9 Accuracy of the projection $P(q)$ over the nominal trajectory. 100
5.10 Norms of the expansions of f_i over the nominal trajectory. 101
5.11 Time vs $\theta \cdot T$ over the step. 101
5.12 The gait in open loop with only the semi-PFL controller and unperturbed initial conditions. 102
5.13 The gait in closed loop and unperturbed initial conditions. 103
5.14 The gait with the parameters of the system scaled up by 6% compared to the design phase 104
5.15 The gait with the parameters of the system scaled down by 8% compared to the design phase 105
5.16 The gait with the initial perturbation $\dot{q}_1(t_0) = \dot{\theta}_*(t_0) + 0.04$. 106
5.17 The gait with the initial perturbation $\dot{q}_1(t_0) = \dot{\theta}_*(t_0) - 0.08$. 107
5.18 The gait with the initial conditions perturbed such that, measured in absolute angles, the swing and stance leg have zero velocity, while the torso is initialized with $\dot{q}_2^a(t_0) = -0.75\,\mathrm{rad\,s^{-1}}$. 108

C.1 The gait simulated in closed loop with zero initial perturbation. 133
C.2 Shows the evolution of the system with the parameters scaled up by 10%. 134
C.3 The gait with the initial perturbation $\dot{q}_1(t_0) = \dot{\theta}_*(t_0) + 0.015$. 135

Abbreviations and Acronyms

COT	Energetic cost of transport
DOF	Degrees of freedom
LQR	Linear-quadratic regulator
MG	Motion generator
PFL	Partial feedback linearization
RDE	Riccati differential equation
SPD	Symmetric positive definite

SPSD	Symmetric positive semidefinite
VHC	Virtual holonomic constraint
w.r.t	With respect to

Chapter 1

Introduction

1.1 Background and Motivation

The study of walking for two-legged robots, often referred to as bipeds, is a field of increasing popularity. Studying bipedal walking can give insight into how humans walk, while the versatility of two-legged walking can make such robots ideal for operation in rough terrain and environments designed for human beings.

Some important criteria for bipedal walking, and locomotion in general, are energy efficiency and the stability of the motion w.r.t. perturbations from the nominal locomotion. In order to evaluate the energy efficiency, it is common to measure the *energetic cost of transport* (COT). The COT is a dimensionless quantity, defined as the energy consumed to move a unit weight a unit distance [25]. The COT plotted against body mass for a variety of well known robots, animals and vehicles is shown in Figure 1.1. One of the most energy efficient bipedal robots that have been made, the Cornell biped [6], has a COT of approximately 0.2, which is about the same as the minimum for an average human being. Although the Cornell biped can perform highly efficient walking, it can not do other simple takes like standing still [25].

In order to ensure stability for bipedal walking, there exists a variety of different strategies. A common strategy is the zero moment point (ZMP) approach [53]. By ensuring that the whole area of the sole of stance foot is in contact with the walking surface, i.e, the robot has flat feet which does not roll, the robot can be regarded as fully actuated throughout the waling cycle such that motion planning and control can be largely based on the kinematics rather than the dynamics of the robot. This approach is employed by many well known bipedal robots, like for example Honda's ASIMO [40]. ASIMO can – in addition to stable walking – run, climb stairs and even kick a ball. Similar strategies were also employed by several bipedal robots from different teams competing in the recent DARPA Robotics Challenge [38], where the robots had to perform a variety of tasks, from opening doors and drilling holes, to walking across difficult terrain as to emulate operating in a disaster area.

Although these robots and others which are employing the ZMP approach or similar

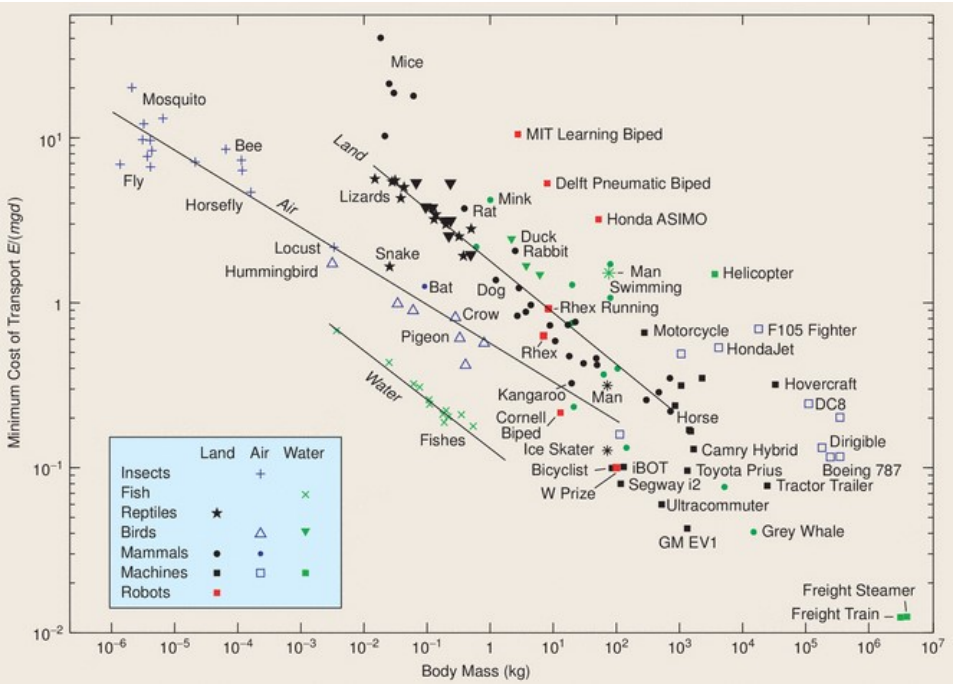


Figure 1.1: The minimum cost of transport as function of body mass for a variety of robots, animals, and vehicles. Reprinted from IEEE Robotics and Automation Magazine, Vol. 14, No. 2, A.D Kuo, *Choosing your steps carefully,* pages 18-29, Copyright © 2007 IEEE.

methods are highly advanced and can perform a variety of tasks, the restriction that the feet can not roll results in gaits which look very non-human like and consume far more energy than that of a walking human being. Indeed, ASMIO has a minimum COT of approximately 3.2, which over ten times that of an average human being [6].

For this reasons, strategies for creating and stabilizing gaits with low energetic cost by drawing inspiration from humans are becoming widely studied. One such approach is *dynamic walking*, also called *limit cycle walking*, where the walking motion is analysed as a limit cycle. This approach, first presented in [20], harnesses the natural dynamics of the system in order to generate energy efficient and often quite human-like walking [25]. The field gained a lot of traction after the seminal paper [31] on passive walking by McGeer. In this paper, McGeer demonstrated that a completely unactuated bipedal robot could walk down a shallow slope only powered by gravity, resulting in a very human-like gait. From this observation, strategies to recreate such motion on varying slopes and level ground using control action have been studied. One such strategy is the method of passivity-based control presented in [50]. This method takes an already known passive gait found for a given slope and transforms and stabilizes it for a differently angled slope using energy shaping techniques [50].

Some other methods that are gaining traction within the scope of dynamic walking, are those employing the notion of so-called *virtual holonomic constraints* (VHC). These

methods introduces a set of geometric relations, called virtual holonomic constraints, between the configuration variables of the system and some scalar variable inherent in the system, which is referred to as the *motion generator* (MG). By assuming that some control action can make these relations invariant, the dynamics of the system is restricted to evolve on a two-dimensional submanifold of the state space describing the time evolution of the motion generator. Thus, feasible motions of the system can be found from the geometric curves defined by the virtual holonomic constraints and the evolution of the motion generator. For underactuated system, the evolution of the motion generator can not be chosen arbitrarily[1], as is the case for fully actuated systems, but must comply with the passive degrees of freedom and the choice of the geometric relations.

The method of using VHCs for dynamic walking was first demonstrated on the planar bipedal robot RABBIT [5], which had five degrees of freedom (DOF) and one degree of underactuation. In order to generate a stabilizing controller, the authors derived the so called *hybrid zero dynamics* corresponding to reduced dynamics found by assuming that the virtual holonomic constraints were perfectly enforced by feedback. Stability could then be determined by the contraction of the Poincarè map over a step using a two-dimensional Poincarè surface found from the hybrid zero dynamics [5]. For further details on this approach, see, e.g., [16, 55, 54].

In [46], virtual holonomic constraints were used for orbital stabilization of periodic solution for mechanical system with one degree of underactuation, and later extended to several degrees of underactuation in [42]. It was shown that there exists a generic choice of *transverse coordinates*, and that the corresponding *transverse dynamics* could be linearized analytically such that an orbitally stabilizing controller could be generated by stabilizing the origin of the linearized transverse dynamics [46, 42].

In this thesis, the virtual holonomic constraint approach from [46] and [42] will be used to generate and stabilize gaits for a planar, three-link compass biped with a torso. In order to account for the sudden change of velocities due to the impact between the swing leg and the ground at the end of a step, the extension of this method to hybrid systems presented in [41] will be used. Stabilizing controllers will then be generated by solving the *Riccati differential equation* (RDE) corresponding to the linearized transverse dynamics with a method based on [18, 17] with some additional constraints to account for the dynamics of the system being hybrid.

1.2 Outline of the Thesis

The outline of this thesis is as follows. In Chapter 2, the theoretical framework of the thesis is presented. This includes some basic concepts in classical mechanics, like the Euler-Lagrange equation; the concept of periodic trajectories and orbital stability; a brief introduction to hybrid systems; the virtual holonomic constraint approach; as well as stabilization of linear time-varying systems using LQR based on the solution of the Riccati differential equation. In Chapter 3, the mathematical model of the three-link biped is presented, with the continuous dynamics modelled using Euler-Lagrange and the derivation of two discrete impact maps. Chapter 4 presents the suggested motion planning procedure

[1] The evolution of the motion generator must of course also comply with the limitations of the system, like restrictions actuator torques, as well as constraints on joint velocities and accelerations.

for finding symmetric gaits using the reduced dynamics derived from the virtual holonomic constraints approach. Chapter 5 regards the development of stabilizing controllers based on the solution of the Riccati differential equation with one jump. The first method that is presented can be used for gaits found with the motion generator chosen as the angle of the stance leg, while for the second, the motion generator is approximated from a projection of the configuration and uses excessive transverse coordinates. Lastly, some concluding remarks and suggestions for further work are given in Chapter 6.

Chapter 2

Theoretical Framework and Preliminaries

In this chapter, the theoretical framework of this thesis is presented, as well as complementary theory meant to help the reader better understand the topics presented. The chapter is outlined as follows. In Section 2.1, a brief overview of classical mechanics is presented, giving a brief introduction to topics as generalized coordinates, holonomic constraint, the Euler-Lagrange equation, as well as the difference between fully actuated and underactuated systems. The notion of hybrid systems is then presented in Section 2.2. Section 2.3 brings up the topics of periodic trajectories, limit cycles and orbital stability, as well as Poincarè sections, which is a useful tool in order to determine orbital stability. In Section 2.4, the theory of virtual holonomic constraints is presented, showing how it can be used for motion planning and determining orbital stability through transverse linearization. Lastly, Section 2.5 shows how linear, time-varying systems can be stabilized using the linear-quadratic regulator approach by solving the Riccati differential equation.

2.1 Classical Mechanics and Equations of Motion

2.1.1 Generalized Coordinates and Holonomic Constraints

Consider a system of K particles where the position of the particles are given by the position vectors $\vec{r}_i$, $i = 1, \ldots, K$, describing the *configuration* of the system. Suppose that each particle of mass m_i is free to move unconstrained in a three-dimensional Cartesian coordinate system such that the configuration of the system is described by $\vec{r} = [\vec{r}_1; \ldots; \vec{r}_K] \in \mathbb{R}^{3K}$.[1] Then, the system is said to have $3K$ *degrees of freedom*

[1] The notation $y = [y_1; \ldots; y_n]$ will used throughout this text to denote that y is a column vector, i.e., $[y_1; \ldots; y_n] = [y_1, \ldots, y_n]^T$.

(DOF) and the motion of each particle is described by Newton's second Law of motion:

$$m_i \frac{d^2\vec{r}}{dt^2} = m_i \vec{a}_i = \sum_{j=1}^{k} \vec{F}_{j,i} + \vec{F}_i^{(e)} \tag{2.1}$$

where $\vec{a}_i$ is the second derivative of $\vec{r}_i$ w.r.t. to time, i.e., the acceleration of particle i, $\vec{F}_i^{(e)}$ is the external force acting upon it and $\vec{F}_{j,i}$ is the internal force acting upon the particle due to the particle j. Note that $\vec{F}_{j,i} = 0$ for $j = i$.

Now, suppose that the particles are constrained in some fashion. Then, one must also take into the forces needed for the constraints to hold in (2.1), called the *constraint forces*. Some constraints, called *holonomic constraints*, have the useful property that they make it possible to reduce the degrees of freedom of the system. A constraint is called holonomic if it can be written as an equality constraint in terms of the position vectors:

$$h(\vec{r}_1, \ldots, \vec{r}_K, t) = 0, \tag{2.2}$$

and *nonholonomic* otherwise [51]. Such constraints can be thought of as either restricting movement of a particle in certain directions or by coupling the particles in the system such that they are not fully independent, e.g., two particles linked by a non-flexible rod. Since these are equality constraints only in terms of the position vectors, one can, for a system with C holonomic constraints

$$h_j(\vec{r}_1, \ldots, \vec{r}_K, t) = 0, \quad j = 1, \ldots, C, \tag{2.3}$$

reduce the degrees of freedom to $N = 3K - C$ by introducing a new set of N independent coordinates, $q_1, \ldots, q_N$, such that the position, $\vec{r}_i$, and velocity, $\vec{v}_i$, of each particle, $i = 1, \ldots, K$, can be written in terms of the coordinates as

$$\begin{aligned} \vec{r}_i &= \vec{r}_i(q_1, \ldots, q_N, t), \\ \vec{v}_i &= \frac{d\vec{r}}{dt} = \sum_{n=1}^{N} \frac{\partial \vec{r}_i}{\partial q_n} \dot{q}_n + \frac{\partial \vec{r}_i}{\partial t}. \end{aligned} \tag{2.4}$$

These new coordinates, $q = [q_1; \ldots; q_N] \in Q \subseteq \mathbb{R}^N$, are called *generalized coordinates* and evolve on the configuration manifold, denoted Q, describing all feasible configurations of the system. The generalized coordinates, $q_1, \ldots, q_N$, are called *minimal coordinates* if they are the minimum number of coordinates needed to describe the configuration of the system, and *excessive coordinates* otherwise. The states of the system are described by $[q; \dot{q}] \in TQ$, where TQ is the $2N$-dimensional tangent bundle of Q called the *state space* of the dynamical system.

2.1.2 The Euler-Lagrange Equation

For systems subject to holonomic constraints, it is often convenient using the Euler-Lagrange equation for finding the equations of motion, i.e., the dynamics of the system, as one can

use generalized coordinates, $q \in Q$, that contain the holonomic constraints (2.3) conditions implicitly in the transformation equation (2.4) [15]. The Euler-Lagrange equation can be derived using either D'Alembert's principle:

$$\sum_{i=1}^{K} \left(\vec{F}_i^{(a)} - m_i \frac{d^2 \vec{r}_i}{dt^2} \right) \cdot \delta \vec{r}_i = 0 \tag{2.5}$$

where $\vec{F}_i^{(a)}$ is the applied forces and $\delta \vec{r}_i$ is a *virtual displacement*; or Hamilton's principle of least action. However, such derivations are outside the scope of this text; see, e.g., [9] or [15].

In order to derive the equations of motion using the method of Euler-Lagrange, one starts by defining the Lagrangian, $\mathcal{L}(q, \dot{q})$, of the system given by the kinetic energy, $\mathcal{K} := \frac{1}{2} \dot{q}^T M(q) q$, and the potential energy, $\mathcal{P} := V(q)$, as

$$\mathcal{L}(q, \dot{q}) := \mathcal{K} - \mathcal{P} = \frac{1}{2} \dot{q}^T M(q) \dot{q} - V(q), \tag{2.6}$$

where $M(q)$ is the symmetric, positive definite *inertia matrix*, whose elements are given by

$$m_{i,j}(q) := \frac{\partial}{\partial \dot{q}_i} \left(\frac{\partial \mathcal{K}}{\partial \dot{q}_j} \right), \quad \forall i, j \in \{1, 2, \ldots, n\}. \tag{2.7}$$

The Euler-Lagrange equation is then defined as follows

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}} \right) - \frac{\partial \mathcal{L}(q, \dot{q})}{\partial q} = B(q) u, \tag{2.8}$$

where $u \in \mathbb{R}^m$ is a vector of independent control inputs and $B(q)$ is an $n \times m$ matrix function of full rank. Thus, the equations of motion of the system are described by (2.8), which can be written on the equivalent form [51]

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) = B(q) u, \tag{2.9}$$

where $G(q) = [\partial V(q)/\partial q_1, \ldots, \partial V(q)/\partial q_n]^T$ is a vector of gravitational terms, referred to as the gravity vector, and the matrix $C(q, \dot{q})$ contains the centrifugal and Coriolis terms, whose elements are given by [51]

$$c_{i,j} = \sum_{k=1}^{n} \Gamma_{i,j}^k(q) \dot{q}_k, \quad \forall i, j \in \{1, 2, \ldots, n\}. \tag{2.10}$$

Herein, $\Gamma_{i,j}^k(q)$ are the Christoffel symbols of the first kind, which are defined as

$$\Gamma_{i,j}^k(q) := \frac{1}{2} \left\{ \frac{\partial m_{i,j}(q)}{\partial q_k} + \frac{\partial m_{i,k}(q)}{\partial q_j} - \frac{\partial m_{k,j}(q)}{\partial q_i} \right\}. \tag{2.11}$$

As the inertia matrix, $M(q)$, is positive positive definite, and thus invertible, one can solve (2.9) in terms of the acceleration vector $\ddot{q}$:

$$\ddot{q} = M(q)^{-1} \Big[- C(q, \dot{q}) \dot{q} - G(q) + B(q) u \Big]. \tag{2.12}$$

By defining the variables $x_1 := q$ and $x_2 := \dot{q}$, one can write the system as a set of $2n$ ordinary differential equations (ODE) as follows

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= M(x_1)^{-1}\Big[- C(x_1, x_2)x_2 - G(x_1) + B(x_1)u\Big], \end{aligned} \tag{2.13}$$

which, by defining $x = [x_1; x_2]$, can be written on the standard form:

$$\dot{x} = f(x) + g(x)u \tag{2.14}$$

with

$$f(x) = \begin{bmatrix} x_2 \\ M(x_1)^{-1}\big(- C(x_1, x_2)x_2 - G(x_1) + \big) \end{bmatrix} \quad \text{and} \quad g(x) = \begin{bmatrix} 0 \\ M(x_1)^{-1}B(x_1) \end{bmatrix}.$$

2.1.3 Fully Actuated vs Underactuated Systems

A mechanical system is said to be underactuated if it has fewer actuators than degrees of freedom. Thus, the system (2.9) is fully actuated if $m = \text{rank}B(q) = n$, and underactuated by degree $n - m$ otherwise. If the system is fully actuated, i.e., $\text{rank}B(q) = n$, then $B(q)$ is invertible, meaning that for any fully actuated system one can set the input to

$$u = B(q)^{-1}\big[M(q)v + C(q, \dot{q}) + G(q)\big] \tag{2.15}$$

such that (2.9) reduces to

$$\ddot{q} = v.$$

This means that it is possible to generate arbitrary accelerations in any direction on Q.[2] Writing the system on the form (2.13), it takes the form of a double integrator:

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= v, \end{aligned} \tag{2.16}$$

which can be written on standard linear form as

$$\dot{x} = Ax + bv = \begin{bmatrix} \mathbf{0}_n & I_n \\ \mathbf{0}_n & \mathbf{0}_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \mathbf{0}_n \\ I_n \end{bmatrix} v, \tag{2.17}$$

where I_n is the $n \times n$ identity matrix and $\mathbf{0}_n$ is an $n \times n$ matrix of zeros.

For underactuated systems, i.e., $\text{rank}B(q) = m < n$, no such exact feedback linearization as (2.15) exists. However, the method of *partial feedback linearization* (PFL) presented in [48] (see also [23]) makes it possible to linearize m of the second order equations of (2.9) using nonlinear feedback, where the $n - m$ remaining nonlinear equation are termed the *internal dynamics*. From the internal dynamics, one can derive the so called *zero-dynamics*. Stability of the system can then be determined by the stability of the zero-dynamics [48].

[2] This is of course not the case if there are limitations on the actuator inputs or joint velocities.

2.2 Hybrid Mechanical Systems

Consider a mechanical system where the continuous dynamics are of the form of the Euler-Lagrange equation (2.9). Suppose the system is subject to a discrete event defined as an instantaneous jump in the states of the system, such as the impact between the swing leg of a bipedal robot and the ground at the end of a step. The jump consists of the pair of hypersurfaces, $\{\Gamma_-, \Gamma_+\}$, in the state space of the system and instantaneous mappings F. Whenever the states of the system hits the surface Γ_-, the states are immediately mapped by F onto Γ_+ such that the update law is defined by the set of triples

$$\{\Gamma_-, \Gamma_+\}, \quad F : \Gamma_- \to \Gamma_+. \tag{2.18}$$

The system's dynamics are thus of the form

$$\begin{aligned} M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = B(q)u, \qquad & [q_-; \dot{q}_-] \notin \Gamma_-, \\ \Gamma_+ \ni [q_+; \dot{q}_+] = F\big([q_-; \dot{q}_-]\big), \qquad & [q_-; \dot{q}_-] \in \Gamma_-, \end{aligned} \tag{2.19}$$

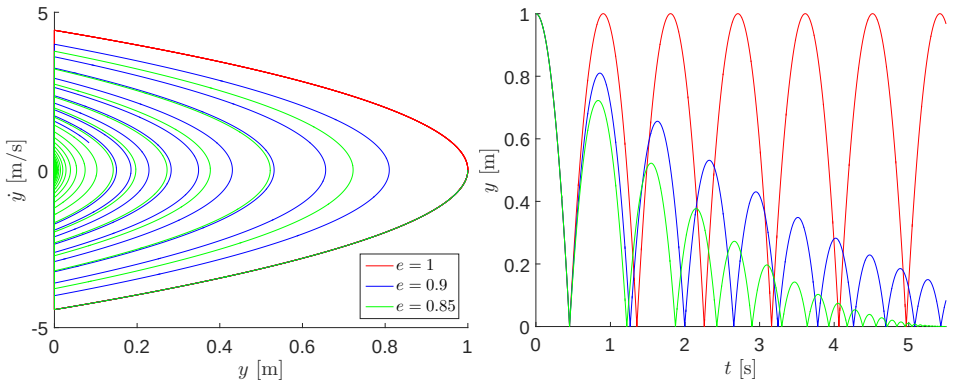
with $[q; \dot{q}] \in TQ \subseteq \mathbb{R}^{2n}$. Here the notation $[q_-; \dot{q}_-]$ and $[q_+; \dot{q}_+]$ denotes the states of the system immediately prior to and after the jump, respectively. Such a mechanical system, which embodies both continuous and discrete behaviours, is referred to as a *hybrid mechanical system* [14]. The discrete parts of the dynamics arises from events where abrupt changes in the states of the system takes place over a sufficiently small timespan such that the events can be modelled as impulsive, i.e., instantaneous jumps in the states of the system.

An example of such a system, is a ball dropped from some initial height that falls towards an immovable horizontal plane, which will be referred to as the ground. The dynamics of the ball in free fall are simply found by Newton's second law as $ma = F^r$, where m is its mass, a its acceleration and F^r are the forces acting upon it. When the ball is falling, its dynamics are thus simply $a = -g$, where g is the gravitational acceleration. However, when the ball hits the ground, and assuming it can not be deformed, its velocity v immediately changes while its vertical position y stays the same. The velocity v_+ of the ball immediately after the impact with the ground is given by $v_+ = -ev_-$, where v_- is its velocity immediately prior to the impact, and $e \in [0, 1]$ is the *coefficient of restitution*, which depends on properties like the materials of the bodies in contact and their geometry, the approach velocity, as well as the duration of contact [13]. Thus, the hybrid dynamics of the ball is given by

$$\begin{aligned} a = -g, \qquad & [y_-; v_-] \notin \Gamma_-, \\ \Gamma_+ \ni v_+ = -ev_-, \qquad & [y_-; v_-] \in \Gamma_-, \end{aligned} \tag{2.20}$$

where Γ_- denotes the impact surface, defined as $\Gamma_- = \{[y; v] : \ y = 0, \ v < 0\}$. If the coefficient of restitution, e, is equal to one such that $v_+ = -v_-$, then the ball will continue to bounce with the same amplitude forever (assuming no air resistance). Such impacts are called *elastic* and the energy of the ball is conserved. For the case when $e < 1$, there is some dissipation of energy at each impact, meaning that the amplitude will be smaller for each impact, eventually reaching zero. Such impacts are called *inelastic*, and *perfectly inelastic* if $e = 0$, meaning that the ball stops after the first impact. Figure 2.1

shows the evolution of a ball dropped from 1 meter for different values of the coefficient of restitution.



(a) Phase portrait. **(b)** Time evolution.

Figure 2.1: A bouncing ball for different values of the coefficient of restitution.

Most bipedal robots are modelled as hybrid systems due to events during walking like knee-locking and the impact of the swing leg with the ground. Although discretizing such events is not necessarily the best representation of these events found in nature, it is the most the most common in the literature under the assumption of completely rigid bodies; see, e.g., [54, 50, 12]. It has also been shown in [20] that such discrete impacts can play a vital role in the stability of bipedal locomotion.

2.3 Periodic Trajectories, Limit Cycles and Stability

In many systems, like bipedal walking, one wants the behaviour of the system to be oscillatory, i.e., repeating its behaviour with some period T. A system is said to have a *periodic solution*[3] if

$$x(t+T) = x(t), \quad \forall t \geq 0,$$

with period $T > 0$. On the phase portrait, such periodic solutions form closed trajectories referred to as *periodic orbits* or *periodic trajectories*. For a second-order autonomous system

$$\dot{x} = f(x), \quad x \in \mathbb{R}^n, \tag{2.21}$$

where $f(x)$ continuously differentiable, one can use the Poincaré-Bendixson Theorem to determine if some closed bounded subset of the plane spanned by the vector field of (2.21) contains a periodic orbit. A corollary of the theorem, given in [23], called the Poincaré-Bendixson Criterion is stated bellow (see, e.g., [21] for the theorem).

Theorem 2.3.1 (Poincaré-Bendixson Criterion, [23])**.** *Consider the system* (2.21) *and let* $\mathcal{M}$ *be a closed bounded subset of the plane such that*

[3] We exclude constant solutions arising from equilibrium points whenever we refer to periodic solutions or periodic orbits throughout this text.

- *$\mathcal{M}$ contains no equilibrium points, or contains only one equilibrium point such that the Jacobian matrix $\left[\partial f(x)/\partial x\right]$ at this point has eigenvalues with positive real parts. (Hence, the equilibrium point is unstable focus or unstable node.)*
- *Every trajectory starting in $\mathcal{M}$ stays in $\mathcal{M}$ for all future time.*

Then, $\mathcal{M}$ contains a periodic orbit of (2.21).

One can think of the Criterion in the following way. Imagine that $\mathcal{M}$ is the closed bounded region enclosed by the closed curves C_1 and C_2 given in Figure 2.2. If there are no equilibrium points in $\mathcal{M}$ and the gradient of the vector field points into $\mathcal{M}$ for all points on C_1 and C_2 such that every trajectory that enters $\mathcal{M}$ stays in $\mathcal{M}$ for all time. Then, there must be at least one periodic orbit Γ in $\mathcal{M}$.

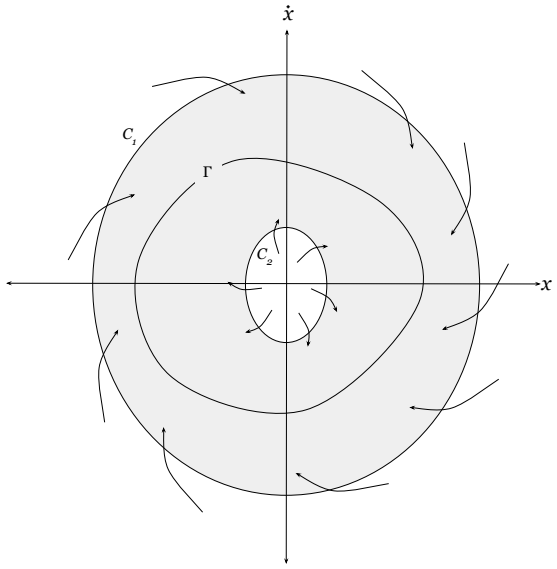


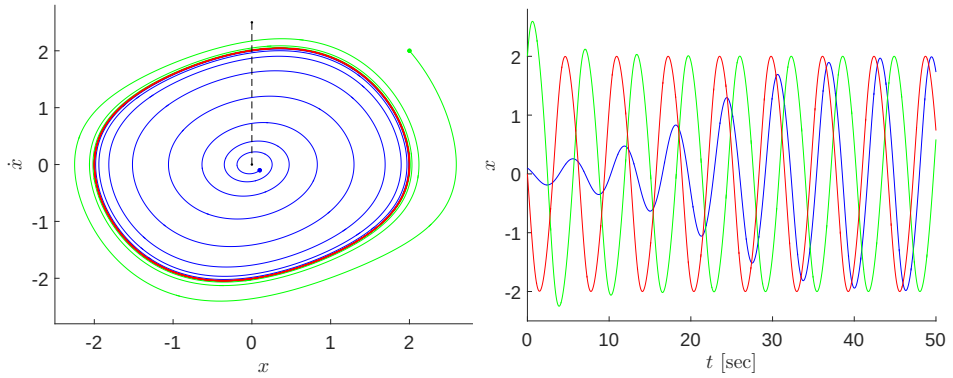
Figure 2.2: Illustration of a periodic trajectory Γ lying inside the annular region enclosed by the closed curves C_1 and C_2.

If a periodic orbit is isolated, i.e., there are no other periodic orbits in the vicinity of the orbit, it is referred to as a *limit cycle*. A limit cycle is said to be stable if all solutions starting in some non-zero vicinity of the cycle, referred to as the *region of attraction* of the cycle, will converge to the cycle as $t \to \infty$, and unstable otherwise. A well known example of a limit cycle, is van der Pol's oscillator [21]

$$\ddot{x} + \mu(x^2 - 1)\dot{x} + x = 0, \tag{2.22}$$

which is stable for $\mu > 0$ and unstable for $\mu < 0$. The phase portait and time evolution of the van der Pol oscillator can be seen in Figure 2.3 for $\mu = 0.5$.

As mentioned in the introduction, dynamic walking is often referred to as limit cycle walking. However, the limit cycles resulting in a walking gait differs from that found



(a) Phase portrait.

(b) Time evolution.

Figure 2.3: The van der Pol oscillator with $\mu = 0.5$ where the red line corresponds to the limit cycle. The black, dotted line in (a) corresponds to the Poincaré section $\mathcal{S}_{vdP} = \{(x, \dot{x}) \in \mathbb{R}^2 : \ x = 0, \ \dot{x} > 0\}$.

for the van der Pol oscillator due to the discrete events of the system from knee-locking and impacts between the feet and the ground. Yet, calling such gaits limit cycles are still appropriate as they can be considered as closed periodic trajectories in the state space, which can be both inherently stable or unstable. Such gaits might, however, not exist on the unactuated mechanical system and can only be realized using control action.

2.3.1 Orbital Stability

The reason all solutions starting in a vicinity of a limit cycle must eventually converge to it for it to be stable, stems from the definition of a limit cycle, i.e., if a non-constant solution starting in a vicinity of a limit cycle stays in the vicinity of the cycle for all time and never converges, then the limit cycle is not an isolated periodic orbit and thus not a limit cycle. Therefore, a stable limit cycle is said to be *orbitally asymptotically stable*. The term *orbitally* refers to the fact that periodic orbits can not be asymptotically stable in the normal sense, i.e., in the same way that equilibrium points are asymptotically stable. To illustrate this, consider the van der Pol oscillator (2.22). Every trajectory that is perturbed a small distance from the limit cycle will eventually converge to the limit cycle, but different perturbations will not all converge to a given point on the cycle. This is clearly illustrated in Figure 2.3b where the perturbations clearly converge to the limit cycle but not in phase with the nominal cycle.

The definition for a periodic orbit to be *orbitally stable*, *orbitally asymptotically stable* and *orbitally exponentially stable* is given next and is based on Definition 8.2 in [23] and [43].

Definition 1. *Let $x(t) = [q(t); \dot{q}(t)]$ be a trajectory with initial conditions $x(t_0) = x_0$. Then, the orbit $\mathcal{O}_*$ is said to be*

- *Orbitally stable if for each* $\epsilon > 0$ *there is* $\delta = \delta(\epsilon) > 0$ *such that*

$$\text{dist}(x_0, \mathcal{O}_*) < \delta \;\Rightarrow\; \text{dist}(x(t), \mathcal{O}_*) < \epsilon, \quad \forall t > t_0.$$

- *Unstable if not orbitally stable.*

- *Orbitally asymptotically stable if it is orbitally stable and* δ *can be chosen such that*

$$\text{dist}(x_0, \mathcal{O}_*) < \delta \;\Rightarrow\; \text{dist}(x(t), \mathcal{O}_*) \to 0 \quad \text{as} \quad t \to \infty.$$

- *Orbitally exponentially stable if there exists some* $c_1 > 0$*,* $c_2 > 0$ *and* δ *can be chosen such than such that*

$$\text{dist}(x_0, \mathcal{O}_*) < \delta \;\Rightarrow\; \text{dist}(x(t), \mathcal{O}_*) \leq c_1 \text{dist}(x_0, \mathcal{O}_*) \exp(-c_2(t - t_0)), \quad \forall t \geq t_0.$$

Here, $\text{dist}(x(t), \mathcal{O})$ is the shortest distance between the orbit $\mathcal{O}$ and the vector $x(t)$, i.e.,

$$\text{dist}(x, \mathcal{O}) = \inf_{y \in \mathcal{O}} \|x - y\|. \tag{2.23}$$

2.3.2 Poincaré first-return map and Poincarè Sections

One way of determining orbital stability of a periodic orbit $\mathcal{O}_*$ of (2.21), is using Poincarè's first-return map [21, 43]. One start by introducing a $(n-1)$-dimensional hyper-surface S, called a Poincaré section, transversal to the flow of $\mathcal{O}_*$, i.e., $n(x) \cdot f(x) \neq 0$, $\forall x \in S$, where $n(x)$ is the normal vector of S at the point x. An example of a Poincarè section for the van der Pol oscillator is $\mathcal{S}_{vdP} = \{[x; \dot{x}] \in \mathbb{R}^2 : \; x = 0, \; \dot{x} > 0\}$ shown in Figure 2.3a.

Given a sufficiently small region $S_0 \subset S$ which is open relative to S and containts the point of intersection between the orbit $\mathcal{O}_*$ and S, the Poincarè map is then defned as $P : \; S_0 \mapsto S$, mapping the initial point of intersection onto the next point the orbit $\mathcal{O}_*$ first intersects S again; see Figure 2.4. For the system (2.21), the Poincarè map is given as the time-discrete system

$$x_p[n+1] = P(x_p[n]) \tag{2.24}$$

where $x_p[n]$ is the point of the nth intersection with the surface $\mathcal{S}$ and $x_p[n+1]$ is the first-return point where the trajectory intersects the Poincarè section again. If the first-return point is the same as the initial point, i.e.,

$$x_p[n+1] = x_p[n] = P(x_p[n]),$$

then $x_p[n]$ is a fixed point of of the map $P(\cdot)$. If this point corresponds to the intersection between the periodic orbit $\mathcal{O}_*$ and the surface S, then the stability of $\mathcal{O}_*$ can be studied by studying the stability of the map $P(\cdot)$. Indeed, an orbit corresponding to the fixed point $x_p[n]$ of the map $P(\cdot)$ is (asymptotically) orbitally stable if and only if $x_p[n]$ is a (asymptotically) stable fixed point of the the map $P(\cdot)$ [52, Lemma 12.2]. Orbital exponential stability can be verified by linearizing the map, $dP : \; TS \mapsto TS$, around the orbit and checking if the eigenvalues of dP lies inside the unit circle; see, e.g., [43] and references therein.

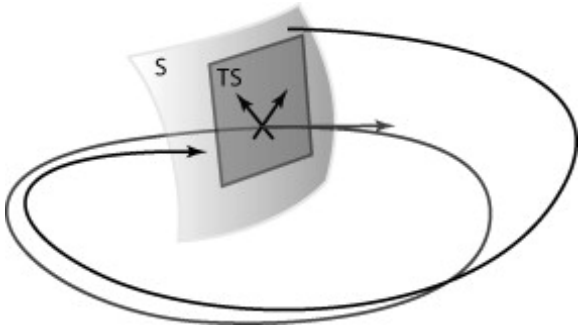


Figure 2.4: Illustration of a Poincarè map $P: \ S \mapsto S$ on the section S and its tangent plane TS. Reprinted from Annual Reviews in Control Vol. 32, Issue 2, A.S. Shiriaev, L.B. Freidovich, I.R. Manchester, *Can we make a robot ballerina perform a pirouette? Orbital stabilization of periodic motions of underactuated mechanical systems,* pages 200-211, Copyright © 2008, with permission from Elsevier.

The use of Poincarè maps for determining stability for bipedal robots is commonly used in the literature; see, e.g., [54]. However, the Poincaré map does not give a continuous representation of the dynamics of the system transverse to the target orbit, but focuses only at one point on the orbit. This means it has limited use for constructive control design [30].

Another way investigating stability of periodic orbits is by introducing a family of Poincarè sections, $\{S\}_{t\in[0,T]}$, called *moving Poincarè sections* [28]. The definition of a moving Poincarè section given in [42] is stated next; see also [28].

Definition 2 ([42])**.** *Let $q_*(t)$ for $t \in [0,T]$ be a solution of the n-degrees-of-freedom Euler-Lagrange ststem* (2.9) *driven by a control input $u_*(t) \in \mathcal{C}^1([0,T])$ with the initial conditions at $q(0) = q_0$, $\dot{q}_*(0) = \dot{q}_0$, such that $\left(|\dot{q}_*(t)|^2 + |\ddot{q}_*(t)|^2\right) > 0$ for all $t \in [0,T]$. The orbit of the corresponding trajectory is*

$$\mathcal{O}_* := \left\{[q;\dot{q}] \in TQ: \ q = q_*(\tau), \ \dot{q} = \dot{q}_*(\tau), \ \tau \in [0,T]\right\} \tag{2.25}$$

and its tublar neighborhood, the set of all points on distance not bigger than som $\epsilon > 0$, is

$$\mathcal{O}_\epsilon(q_*) = \{[q;\dot{q}] \in TQ: \ \min_{\tau\in[0,T]} \|[q - q_*(\tau); \dot{q} - \dot{q}_*(\tau)]\| \leq \epsilon\}. \tag{2.26}$$

Then, a family of $(2n-1)$-dimensional $\mathcal{C}^1$-smooth surfaces $\{S\}_{t\in[0,T]}$ is called a moving Poincarè section associated with the solution $q_*(t)$, $t \in [0,T]$, if

1. *The Surfaces are locally disjoint, i.e., $\exists \epsilon > 0: \ S(\tau_1) \cap S(\tau_2) \cap \mathcal{O}_\epsilon(q_*) = \emptyset$, for all $\tau_1 \in [0,T)$, $\tau_2 \in (0,T]$, $\tau_1 \neq \tau$.*
2. *Each of the surfaces $S(\cdot)$ locally intersects the orbit only in one point, i.e., $\exists \epsilon > 0: \ S(\tau) \cap \{[q_*(t); \dot{q}_*(t)], |t-\tau| < \epsilon\} \cap \mathcal{O}_\epsilon = \{[q_*(t); \dot{q}_*(t)]\}$ for each $\tau \in [0,T]$.*
3. *The surfaces $S(\cdot)$ are smoothly parametrized by time, i.e., $\exists f_s(q,\dot{q},t) \in \mathcal{C}^1(\mathbb{R}^n, \mathbb{R}^n, \mathbb{R}): \ S(t) \cap \mathcal{O}_\epsilon(q_*) = \{[q;\dot{q}] \in \mathbb{R}^n: \ f_s(q,\dot{q},t) = 0\} \cap \mathcal{O}_\epsilon(q_*)$.*

4. The surfaces $S(\cdot)$ are transversal to the orbit (2.25)*, i.e.,*

$$\left[\frac{\partial f_s}{\partial q}\Big|_{\substack{q=q_*(t)\\ \dot{q}=\dot{q}_*(t)}}\right]^T \dot{q}_*(t) + \left[\frac{\partial f_s}{\partial \dot{q}}\Big|_{\substack{q=q_*(t)\\ \dot{q}=\dot{q}_*(t)}}\right]^T \ddot{q}_*(t) \neq 0, \quad \forall t \in [0,T].$$

Given a moving Poincarè section $\{S(t)\}_{t\in[0,T]}$ assosicated with the motion $q_*(t)$ of the system (2.9), one can perform the (local) coordinate transformation $(q,\dot{q}) \mapsto (\psi, x_\perp)$ where the scalar variable $\psi(t)$ parametrizes position along the periodic orbit (2.25) and the $(2n-1)$-dimensional vector $x_\perp(t)$ defines the location on the surface $S(t)$ [42]. The vector $x_\perp$ is refereed to as a vector of *transverse coordinates*, while the dynamics of (2.9) rewritten in terms of $(\psi, x_\perp)$ are called *transverse dynamics*. Linearizing the transverse dynamics along the solution $q_*(t),\ t \in [0,T]$, gives rise to a linear time-varying control system of $2n$-dimensions defined on $t \in [0,T]$. The $(2n-1)$-dimensional subsystem that corresponds to linearization of the dynamics of the transverse coordinates $x_\perp$ is called a *transverse linearization* and evolve on the tangent plane $TS(t)$ of the moving Poincarè sections $S(t)$; see Figure 2.5.

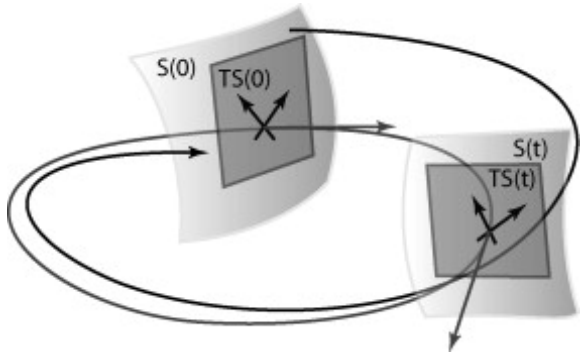


Figure 2.5: Illustration of a moving Poincarè section $S(\tau)$ and its tangent plane $TS(\tau)$. Reprinted from Annual Reviews in Control Vol. 32, Issue 32, A.S. Shiriaev, L.B. Freidovich, I.R. Manchester, *Can we make a robot ballerina perform a pirouette? Orbital stabilization of periodic motions of underactuated mechanical systems,* pages 200-211, Copyright © 2008, with permission from Elsevier.

2.4 The Virtual Holonomic Constraints Approach

The virtual holonomic constraints approach is a useful tool for both planning and stabilizing motions for underactuated mechanical systems. As mentioned in the introduction, by introducing a set of geometric relations between the generalized coordinates and a scalar variable, referred to as the *motion generator* (MG), which are kept invariant by feedback control, the evolution of the mechanical system is restricted to a two-dimensional submanifold of the state space. Thus, the virtual holonomic constraints approach reduces the complexity motion planning. The definition of a virtual holonomic constraint (VHC) which is used in this thesis is given next.

Definition 3 (Virtual holonomic constraints)**.** *Suppose there exists a solution $q_*(t) \in \mathbb{R}^n$ of the Euler-Lagrange system* (2.9) *which can be represented by the geometric relation*

$$q_*(t) = \Phi\big(\theta_*(t)\big), \quad \theta_*(t) \in \Theta_* \subset \mathbb{R}, \tag{2.27}$$

where $\Phi(\theta) = [\phi_1(\theta); \dots; \phi_n(\theta)]$ with $\phi_i : \Theta_* \to \mathbb{R}$, $i = 1, \dots, n$, being $\mathcal{C}^2$-smooth functions. Then the relations (2.27) *are called*

- *holonomic (geometric) constraints if they represent a restriction on the generalized coordinates physically imposed on the system;*
- *virtual holonomic (geometric) constraints if the relations are not physically imposed on the system but rather are kept invariant by some control action.*

Herein, the scalar variable θ is the motion generator which synchronizes the motion of the mechanical system through the synchronization functions $\phi_i(\theta)$, which are the VHCs enforced upon the system by feedback control. The MG θ can often be chosen as one of the generalized coordinates, i.e., $\theta(t) = q_i(t)$ for some $i \in \{1, \dots, n\}$, or some parameter of the system like the arc length along the orbit. In the literature, it is commonly chosen as a passive coordinate [55, 12].

In this section, the virtual holonomic constraints approach presented in [46] and [42] will be presented. First, it will shown that enforcing a set of virtual holonomic constraints upon the system reduces evolution of the system down to that of a second-order differential equation, which will be referred to as the *reduced dynamics*. The reduced dynamics are then shown to have an inherent integral of motion which preserves its zero value along the solution of interest, and whose derivative w.r.t. time can be computed analytically. It is then shown that there exists a generic choice of transverse coordinates, which allows an analytic procedure to linearize the the transverse dynamics such that stability can be assessed on the corresponding linear system. The method is then extended to hybrid systems with one jump using the method presented in [41].

2.4.1 The Reduced Dynamics

Let $q_*(t)$ be a solution of the Euler-Lagrange equation (2.9) driven by a $\mathcal{C}^1$-smooth control input $u_*(t)$ such that the set of geometric relations

$$q_1^* = \phi_1(\theta_*, P), \dots q_n^* = \phi_n(\theta_*, P), \quad \theta_* \in \Theta_* \subset \mathbb{R} \tag{2.28}$$

are valid along the orbit (see (2.25))

$$\mathcal{O}_* := \big\{[q; \dot{q}] \in TQ : \; q = q_*(\tau), \; \dot{q} = \dot{q}_*(\tau), \; \tau \in [0, T]\big\}$$

for the $\mathcal{C}^2$-smooth synchronization functions $\phi_i : \Theta_* \to \mathbb{R}$, $i = 1, \dots, n$ for a set of parameters P. Using (2.28) and assuming that some control signal $u_*(t)$ can keep the relations invariant, one can rewrite (2.9) in terms of the motion generator θ_*, using that

$$\dot{q} = \Phi^{'}(\theta_*)\dot{\theta}_* \quad \text{and} \quad \ddot{q} = \Phi^{'}(\theta_*)\ddot{\theta}_* + \Phi^{''}(\theta_*)\dot{\theta}_*^2,$$

where

$$\begin{aligned}\Phi(\theta) &:= \Phi(\theta, P) = [\phi_1(\theta, P); \ldots; \phi_n(\theta, P)],\\ \Phi^{'}(\theta) &:= \Phi^{'}(\theta, P) = \frac{\partial}{\partial\theta}\Phi(\theta, P) = [\phi_1^{'}(\theta, P); \ldots; \phi_n^{'}(\theta, P)],\\ \Phi^{''}(\theta) &:= \Phi^{''}(\theta, P) = \frac{\partial^2}{\partial\theta^2}\Phi(\theta, P) = [\phi_1^{''}(\theta, P); \ldots; \phi_n^{''}(\theta, P)],\end{aligned} \tag{2.29}$$

such that one obtains

$$\begin{aligned}M\big(\Phi(\theta_*)\big)\Big[\Phi^{''}(\theta_*)\dot{\theta}_*^2 + \Phi^{'}(\theta_*)\ddot{\theta}_*\Big] + C\big(\Phi(\theta_*), \Phi^{'}(\theta_*)\dot{\theta}_*\big)\Phi^{'}(\theta_*)\dot{\theta}_*\\ +G\big(\Phi(\theta_*)\big) = B\big(\Phi(\theta_*)\big)u_*.\end{aligned} \tag{2.30}$$

If the system (2.9) has $m < n$ independent control variables, i.e., it is underactuated by a degree $n - m$, such that the smooth matrix function $B(q) \in \mathbb{R}^{n\times m}$ has rank m, then there exists a smooth matrix function $B^{\perp}(q) \in \mathbb{R}^{(n-m)\times n}$ of rank $n - m$ such that $B^{\perp}(q)B(q) = 0$, called the annihilator of $B(q)$. Multiplying both sides of (2.30) by $B^{\perp}\big(\Phi(\theta_*)\big)$, making it independent of u_*, yields

$$\begin{aligned}&B^{\perp}\big(\Phi(\theta_*)\big)\Big\{M\big(\Phi(\theta_*)\big)\Big[\Phi^{''}(\theta_*)\dot{\theta}_*^2 + \Phi^{'}(\theta_*)\ddot{\theta}_*\Big]\\ &+ C\big(\Phi(\theta_*), \Phi^{'}(\theta_*)\dot{\theta}_*\big)\Phi^{'}(\theta_*)\dot{\theta}_* + G\big(\Phi(\theta_*)\big)\Big\}\\ &= B^{\perp}\big(\Phi(\theta_*)\big)B\big(\Phi(\theta_*)\big)u_* = 0.\end{aligned} \tag{2.31}$$

This in turn can be written as $n - m$ second order differential equations of the form

$$\alpha_i(\theta_*, P)\ddot{\theta}_* + \beta_i(\theta_*, P)\dot{\theta}_*^2 + \gamma_i(\theta_*, P) = 0, \quad i = 1, \ldots, n - m, \tag{2.32}$$

where

$$\begin{bmatrix}\alpha_1(\theta_*)\\ \vdots\\ \alpha_{n-m}(\theta_*)\end{bmatrix} = B^{\perp}\big(\Phi(\theta_*)\big)M\big(\Phi(\theta_*)\big)\Phi^{'}(\theta_*), \tag{2.33a}$$

$$\begin{bmatrix}\beta_1(\theta_*)\\ \vdots\\ \beta_{n-m}(\theta_*)\end{bmatrix} = B^{\perp}\big(\Phi(\theta_*)\big)\Big[M\big(\Phi(\theta_*)\big)\Phi^{''}(\theta_*) + C\big(\Phi(\theta_*), \Phi^{'}(\theta_*)\big)\Phi^{'}(\theta_*)\Big], \tag{2.33b}$$

$$\begin{bmatrix}\gamma_1(\theta_*)\\ \vdots\\ \gamma_{n-m}(\theta_*)\end{bmatrix} = B^{\perp}\big(\Phi(\theta_*)\big)G\big(\Phi(\theta_*)\big). \tag{2.33c}$$

For the case of $m = n - 1$, assuming a feedback controller can make the relations (2.28) invariant, the solutions of the system (2.9) tend asymptotically to the solution of the second order equation

$$\alpha(\theta_*, P)\ddot{\theta}_* + \beta(\theta_*, P)\dot{\theta}_*^2 + \gamma(\theta_*, P) = 0, \tag{2.34}$$

which results in a family of 2-dimensional manifolds $Z(P) \subset TQ$ defined by

$$Z(P)) = \Big\{ [q; \dot{q}] \in TQ : \; q = \Phi(\theta, P), \; \dot{q} = \Phi^{'}(\theta, P)\dot{\theta} \Big\}. \tag{2.35}$$

Herein, equation (2.34) and $Z(p)$ will be referred to as the *reduced dynamics* and the *constraint manifold*, respectively, throughout this text[4].

Thus, one can study the system (2.9) by only looking at the solutions $(\theta_*(t), \dot{\theta}_*(t))$ of (2.34) and the relations (2.28) constrained to $Z(P)$. Furthermore, the equation (2.34) have a few important properties which will be stated next.

The first observation one can from (2.34) is that it has an asymptote at $\alpha(\theta) \equiv 0$, i.e., the acceleration $\ddot{\theta}$ tends to infinity. Thus any solution θ_* must be such that $\alpha(\theta_*) \neq 0$ for all $\theta_* \in \Theta_*$ on the interval of interest.

The second observation one can make is that it only has two possible types of equilibrium points, namely *centers* and *saddles*, which are found by solving for $\bar{\theta}$ such that $\gamma(\bar{\theta}) \equiv 0$ (see, e.g., [23] for an introduction to equilibrium points). The method of how to determine the type of equilibrium can be found in [47]. Since the reduced dynamics is quadratic in the velocity $\dot{\theta}$, it is clear that the vector field is mirrored about the horizontal θ-axis on the phase portrait. Thus, it is clear that for any solution that crosses the horizontal θ-axis will return to its initial state θ_0 and continue, i.e., the solution is periodic and thus orbiting a center equilibrium point.

Another interesting and very useful property of (2.34) is that it is integrable, which makes it possible to find an *integral of motion* which preserves its zero value along any solution of the reduced dynamics. This function will be referred to as the *integral function* throughout this text and is defined in the next theorem.

Theorem 2.4.1 (Integral of motion of the reduced dynamics, [45, 46])**.** *For a solution* $(\theta_*(t), \dot{\theta}_*(t))$ *of* (2.34) *with initial conditions* $(\theta_0, \dot{\theta}_0)$ *on which* $\alpha(\theta_*) \neq 0$*, the function*

$$\begin{aligned} I(\theta_*(t),\dot{\theta}_*(t), \theta_0, \dot{\theta}_0) = \dot{\theta}_*^2 - \exp\left\{ -\int_{\theta_0}^{\theta_*} \frac{2\beta(\tau)}{\alpha(\tau)} d\tau \right\} \\ \times \left[\dot{\theta}_0^2 - \int_{\theta_0}^{\theta_*} \exp\left\{ \int_{\theta_0}^{s} \frac{2\beta(\tau)}{\alpha(\tau)} d\tau \right\} \frac{2\gamma(s)}{\alpha(s)} ds \right] \end{aligned} \tag{2.36}$$

preserves it zero value along the solution irrespective of the boundedness of $(\theta_*(t), \dot{\theta}_*(t))$.

Proof. The proof is given in Appendix A.1 □

It turn out that the time derivative of this function has a further useful property as its time-derivative, even though the right-hand side of (2.34) is not zero, is well defined and given in the next theorem.

Theorem 2.4.2 ([46])**.** *For some constants* x *and* y*, the derivative w.r.t. time of the function* $I(\theta, \dot{\theta}, x, y)$ *defined in* (2.36)*, calculated along a solution* $(\theta(t), \dot{\theta}(t))$ *of the system*

$$\alpha(\theta)\ddot{\theta} + \beta(\theta)\dot{\theta}^2 + \gamma(\theta) = W, \tag{2.37}$$

[4] In the litterature, (2.34) has been referred to as the *virtual limit system* [46] or *zero-dynamics* [16, 23], while $Z(P)$ has been referred to as the *zero dynamics manifold* [16].

can be computed as

$$\dot{I}(\theta,\dot{\theta},x,y)=\frac{2\dot{\theta}}{\alpha(\theta)}\Big(W-\beta(\theta)I\Big). \tag{2.38}$$

Proof. The proof is given in Appendix A.2 □

The integral function and its time derivative will prove to be useful when the topics of transverse coordinates and transverse linearization are presented.

2.4.2 Partial Feedback Linearization

Introducing a new set of $(n+1)$ excessive coordinates

$$y_1=q_1-\phi_1(\theta),\dots,y_n=q_n-\phi_n(\theta)\quad\text{and}\quad\theta\in\Theta\subseteq\mathbb{R}, \tag{2.39}$$

where $y_1,\dots,y_n$ are the deviations from the virtual holonomic constraints (2.28) and θ is the synchronization variable, such that

$$\begin{aligned}\dot{y}_i&=\dot{q}_i-\phi_i^{'}(\theta)\dot{\theta},\\ \ddot{y}_i&=\ddot{q}_i-\phi_i^{''}(\theta)\dot{\theta}^2-\phi_n^{'}(\theta)\ddot{\theta},\end{aligned} \tag{2.40}$$

for $i=1,\dots,n$. Herein, it is possible to locally rewrite one of the generalized coordinates in terms of $n-1$ of the y coordinates and the motion generator θ using the he relations (2.39); e.g.

$$q_1=\phi_1(\theta)+h(y_2,\dots,y_n,\theta) \tag{2.41}$$

for a $\mathcal{C}^2$-smooth scalar function $h:\ \mathbb{R}^{n-1}\times\Theta\to\mathbb{R}$ which can be found using the implicit function theorem and which is identically zero on the solution $q_*(t)$.

Using the relation (2.41), one can rewrite (2.40) in terms of the time derivatives of the generalized coordinates as

$$\dot{q}=L(\theta,y)\begin{bmatrix}\dot{y}\\ \dot{\theta}\end{bmatrix}\quad\text{and}\quad\ddot{q}=L(\theta,y)\begin{bmatrix}\ddot{y}\\ \ddot{\theta}\end{bmatrix}+N(\theta,\dot{\theta},y,\dot{y}) \tag{2.42}$$

where $\Phi^{'}(\theta)$ and $\Phi^{''}(\theta)$ are defined in (2.29) and the matrix functions $L(\theta,y)$ and $N(\theta,\dot{\theta},y,\dot{y})$ are given by

$$L(\theta,y):=\begin{bmatrix}\text{grad } h\\ I_{n-1}\quad\mathbf{0}_{(n-1)\times 1}\end{bmatrix}+\begin{bmatrix}\mathbf{0}_{n\times(n-1)}\quad\Phi^{'}(\theta)\end{bmatrix}, \tag{2.43a}$$

$$N(\theta,\dot{\theta},y,\dot{y}):=\begin{bmatrix}1\\ \mathbf{0}_{(n-1)\times 1}\end{bmatrix}[\dot{y}^T,\dot{\theta}]\,\text{Hess } h\begin{bmatrix}\dot{y}\\ \dot{\theta}\end{bmatrix}+\Phi^{''}(\theta)\dot{\theta}^2, \tag{2.43b}$$

where $\text{grad } h=[(\partial h/\partial y_2),\dots,(\partial h/\partial y_n),(\partial h/\partial\theta)]$ is the gradient and Hess h is the Hessian of the function $h(y_2,\dots,y_n,\theta)$. Using the coordinate transformation $(q,\dot{q})\mapsto(y,\theta,\dot{y},\dot{\theta})$ with $q_2,\dots,q_n$ given in (2.39) and q_1 from (2.41), it follows from (2.12) and (2.42) that

$$M(\theta,y)^{-1}\Big[B(\theta,y)u-C(\theta,\dot{\theta},y,\dot{y})L(\theta,y)\begin{bmatrix}\dot{y}\\ \dot{\theta}\end{bmatrix}-G(\theta,y)\Big]=L(\theta,y)\begin{bmatrix}\ddot{y}\\ \ddot{\theta}\end{bmatrix}+N(\theta,\dot{\theta},y,\dot{y}), \tag{2.44}$$

Assuming that the function $L(\theta, y)$, defined in (2.43), is nonsingular in a vicinity of a solution

$$y = \begin{bmatrix} y_2; \ldots; y_n \end{bmatrix} \equiv \mathbf{0}_{(n-1)\times 1}, \quad \theta = \theta_*(t), \tag{2.45}$$

then (2.44) can be rewritten as

$$\begin{bmatrix} \ddot{y} \\ \ddot{\theta} \end{bmatrix} = L(\theta, y)^{-1} \Big\{ M(\theta, y)^{-1} \Big[B(\theta, y)u - C(\theta, \dot{\theta}, y, \dot{y}) L(\theta, y) \begin{bmatrix} \dot{y} \\ \dot{\theta} \end{bmatrix} - G(\theta, y) \Big] - N(\theta, \dot{\theta}, y, \dot{y}) \Big\}. \tag{2.46}$$

This can in turn be written in terms of $\ddot{y}$ as the $(n-1)$-dimensional subsystem

$$\ddot{y} = K(\theta, y)u + R(\theta, \dot{\theta}, y, \dot{y}) \tag{2.47}$$

where the matrix functions $K(\theta, y)$ and $R(\theta, \dot{\theta}, y, \dot{y})$ are defined as

$$\begin{aligned} K(\theta, y) &:= \begin{bmatrix} I_{n-1} & \mathbf{0}_{(n-1)\times 1} \end{bmatrix} L(\theta, y)^{-1} M(\theta, y)^{-1} B(\theta, y), \\ R(\theta, \dot{\theta}, y, \dot{y}) &:= \begin{bmatrix} I_{n-1} & \mathbf{0}_{(n-1)\times 1} \end{bmatrix} L(\theta, y)^{-1} \Bigg\{ M(\theta, y)^{-1} \Bigg[- G(\theta, y) \\ &\qquad - C(\theta, \dot{\theta}, y, \dot{y}) L(\theta, y) \begin{bmatrix} \dot{y} \\ \dot{\theta} \end{bmatrix} \Bigg] - N(\theta, \dot{\theta}, y, \dot{y}) \Bigg\}. \end{aligned} \tag{2.48}$$

One degree of underactuation

For the case when rank $B(\theta, y) = n - 1$, i.e., the system is underactuated by a degree of one, and assuming $K(\theta, y)$ is nonsingular in a vicinity of the solution (2.45), the partial feedback linearizing (PFL) controller

$$u = K(\theta, y)^{-1} \big(v - R(\theta, \dot{\theta}, y, \dot{y}) \big) \tag{2.49}$$

reduces (2.47) to

$$\ddot{y} = v, \tag{2.50}$$

where v is a control input to be defined later.

In order to find the remaining internal dynamics for the synchronization variable θ, one can multiply both sides of (2.44) by $B^{\perp}(\theta, y)M(\theta, y)$ and rearrange it to get

$$\begin{aligned} B^{\perp}(\theta, y) \Bigg\{ M(\theta, y) \Big[L(\theta, y) \begin{bmatrix} \ddot{y} \\ \ddot{\theta} \end{bmatrix} + N(\theta, \dot{\theta}, y, \dot{y}) \Big] + C(\theta, \dot{\theta}, y, \dot{y}) L(\theta, y) \begin{bmatrix} \dot{y} \\ \dot{\theta} \end{bmatrix} + G(\theta, y) \Bigg\} \\ = B^{\perp}(\theta, y) B(\theta, y) u \\ = 0, \end{aligned} \tag{2.51}$$

where the vector function $B^{\perp}(\theta, y)$ is the $1 \times n$ annihilator of $B(\theta, y)$. Then, from (2.31) and the definitions of the functions $L(\theta, y)$ and $N(\theta, \dot{\theta}, y, \dot{y})$ given in (2.43), one can see that when $y = \dot{y} = \ddot{y} = \mathbf{0}_{(n-1)\times 1}$, this becomes the reduced dynamics (2.34). Thus, using

Hadamard's Lemma (see Appendix A.3) about $y = \dot{y} = \ddot{y} = \mathbf{0}_{(n-1)\times 1}$ and using (2.50) to eliminate $\ddot{y}$, one can write (2.51) as

$$\begin{aligned}\alpha(\theta)\ddot{\theta} + \beta(\theta)\dot{\theta}^2 + \gamma(\theta) =& g_I(\theta,\dot{\theta},\ddot{\theta},y,\dot{y})I + g_y(\theta,\dot{\theta},\ddot{\theta},y,\dot{y})y \\ &+ g_{\dot{y}}(\theta,\dot{\theta},\ddot{\theta},y,\dot{y})\dot{y} + g_v(\theta,\dot{\theta},y,\dot{y})v\end{aligned} \tag{2.52}$$

where the left-hand side matches the structure of the reduced dynamics (2.34), the scalar function I is given in (2.36) and $g_I(\cdot)$, $g_y(\cdot)$, $g_{\dot{y}}(\cdot)$, $g_v(\cdot)$ are smooth functions of appropriate dimensions [46, 42]. The above arguments are summarized in the next theorem.

Theorem 2.4.3 ([46])**.** *Let the mechanical system* (2.9) *be underactuated by a degree* $n-m=1$*. Introducing a new set of coordinates* $[y_1,\dots,y_{n-1},\theta]$ *given by* (2.39) *and the* $\mathcal{C}^2$*-smooth geometric relations* (2.28)*, such that the matrix function* $L(\theta,y)$ *and* $K(\theta,y)$*, given by* (2.43) *and* (2.48)*, respectively, are nonsingular in a vicinity of a solution* (2.45)*. Then, the partial feedback linearizing input* (2.49) *brings the system* (2.9) *into the partly linear form*

$$\begin{aligned}\alpha(\theta)\ddot{\theta} + \beta(\theta)\dot{\theta}^2 + \gamma(\theta) =& g_I(\theta,\dot{\theta},\ddot{\theta},y,\dot{y})I + g_y(\theta,\dot{\theta},\ddot{\theta},y,\dot{y})y \\ &+ g_{\dot{y}}(\theta,\dot{\theta},\ddot{\theta},y,\dot{y})\dot{y} + g_v(\theta,\dot{\theta},y,\dot{y})v\end{aligned} \tag{2.53}$$

$$\ddot{y} = v \tag{2.54}$$

where the left-hand side of (2.53) *matches the structure of the* reduced dynamics (2.34) *and* $g_y(\cdot)$*,* $g_{\dot{y}}(\cdot)$*,* $g_v(\cdot)$ *are smooth functions of appropriate dimensions.*

Degree of underactuation greater than one or non-invertible matrix function $K(\theta,y)$

For the cases when rank$B(q) < n-1$ or the matrix function $K(\theta,y)$ has singular points either on or close to the solution (2.45), one can not use a partial feedback linearizing controller as (2.49). Instead, one introduces a control input

$$u = v + U(\theta,\dot{\theta},y,\dot{y}) \tag{2.55}$$

where $U(\cdot)$is a smooth function which on the orbit defined by (2.25) coincides with the nominal input $u_*(t)$, i.e.,

$$U\big(\theta_*(t),\dot{\theta}_*(t),\mathbf{0}_{(n-1)\times 1},\mathbf{0}_{(n-1)\times 1}\big) = u_*(t), \quad \forall t\in[0,T]. \tag{2.56}$$

This is always possible due the affinity of the Euler-Lagrange system (2.9) in u and the assumptions that $q = q_*(t)$ is a solution [42]. Inserting the controller (2.55) in (2.47), the y-dynamics can be written on the form

$$\ddot{y} = F(\theta,\dot{\theta},y,\dot{y}) + K(\theta,y)v \tag{2.57}$$

with $K(\theta,y)$ given in (2.48) and $F(\theta,\dot{\theta},y,\dot{y})$ is given by

$$F(\theta,\dot{\theta},y,\dot{y}) = R(\theta,\dot{\theta},y,\dot{y}) + K(\theta,y)U(\theta,\dot{\theta},y,\dot{y}) \tag{2.58}$$

which by construction vanishes on the solution $q_*(t)$, i.e.,

$$F\big(\theta_*(t),\dot{\theta}_*(t),\mathbf{0}_{(n-1)\times 1},\mathbf{0}_{(n-1)\times 1}\big) \equiv \mathbf{0}_{(n-1)\times 1}, \quad \forall t\in[0,T].$$

The remaining θ-dynamics are found in the same manner as the case of underactuation $n-1$, with the exception that there are $n-m$ equations for the reduced dynamics given by (2.32). Thus, the remaining θ-dynamics are of the form

$$\begin{aligned}\alpha_i(\theta)\ddot{\theta}+\beta_i(\theta)\dot{\theta}^2+\gamma_i(\theta) =& g_I^{(i)}(\theta,\dot{\theta},\ddot{\theta},y,\dot{y})I^{(i)}+g_y^{(i)}(\theta,\dot{\theta},\ddot{\theta},y,\dot{y})y\\ &+g_{\dot{y}}^{(i)}(\theta,\dot{\theta},\ddot{\theta},y,\dot{y})\dot{y}+g_v^{(i)}(\theta,\dot{\theta},y,\dot{y})v\end{aligned} \tag{2.59}$$

for $i=1,\dots,n-m$, where the left-hand side matches the structure of the reduced dynamics (2.32) and $g_I^{(i)}(\cdot)$, $g_y^{(i)}(\cdot)$, $g_{\dot{y}}^{(i)}(\cdot)$, $g_v^{(i)}(\cdot)$ are smooth functions of appropriate dimensions. This is summarized in the next theorem.

Theorem 2.4.4 (Transverse dynamics, [42])**.** *Let the mechanical system* (2.9) *be underactuated by a degree $n-m$. Introducing a new set of coordinates $[y_1,\dots,y_{n-1},\theta]$ given by* (2.39) *and the $\mathcal{C}^2$-smooth geometric relations* (2.28)*, such that the matrix function $L(\theta,y)$, given by* (2.43)*, is nonsingular in a vicinity of a solution* (2.45)*. Then, the controller* (2.55) *brings the system* (2.9) *onto the form*

$$\begin{aligned}\alpha_i(\theta)\ddot{\theta}+\beta_i(\theta)\dot{\theta}^2+\gamma_i(\theta) =& g_I^{(i)}(\theta,\dot{\theta},\ddot{\theta},y,\dot{y})I^{(i)}+g_y^{(i)}(\theta,\dot{\theta},\ddot{\theta},y,\dot{y})y \\ &+g_{\dot{y}}^{(i)}(\theta,\dot{\theta},\ddot{\theta},y,\dot{y})\dot{y}+g_v^{(i)}(\theta,\dot{\theta},y,\dot{y})v \end{aligned} \tag{2.60}$$

$$\ddot{y} = F(\theta,\dot{\theta},y,\dot{y})+K(\theta,y)v \tag{2.61}$$

for $i=1,\dots,n-m$, with $K(\theta,y)$ and $F(\theta,\dot{\theta},y,\dot{y})$ given in (2.48) *and* (2.58)*, repsectively, and where the left-hand side of* (2.60) *matches the structure of the reduced dynamics* (2.32)*, for which $\alpha_i(\cdot)$ is separated from zero. Furthermore, the scalar function $I^{(i)}(\cdot)$ is given in* (2.36) *and $g_I^{(i)}(\cdot)$, $g_y^{(i)}(\cdot)$, $g_{\dot{y}}^{(i)}(\cdot)$, $g_v^{(i)}(\cdot)$ are smooth functions of appropriate dimensions.*

One can, of course, use Theorem 2.4.4 for the case of underactuation $n-m=1$ when the assumption that the matrix function $K(\theta,y)$, defined in (2.48), is nonsingular might not hold such that the the exact partial feedback linearizing controller (2.49) is not possible, yielding just one equation from (2.60).

2.4.3 Transverse Coordinates and Transverse Linearization

It turns out that if the conditions of Theorem 2.4.4 holds such that one can write the mechanical system (2.9) on the form (2.60)-(2.61), then the system has possesses a natural choice of $(2n-1)$-transverse coordinates [42]

$$x_\perp^{(i)} = \big[I^{(i)}\big(\theta,\dot{\theta},\theta_*(0),\dot{\theta}_*(0)\big);\ y\ ;\ \dot{y}\big] \tag{2.62}$$

which can be introduced in a vicinity of the solution

$$y=\big[y_2;\dots;y_n\big]\equiv \mathbf{0}_{(n-1)\times 1},\quad \theta=\theta_*(t).$$

Here, the scalar function $I^{(i)}(\cdot)$ is defined in (2.36) for the left-hand side of (2.60) for all $i=1,\dots,n-m$, with the assumption that $\alpha_i(\theta_*(t))\neq 0,\forall t$. The transverse dynamics of these coordinates are given by

$$\dot{x}_\perp^{(i)} = \big[\dot{I}^{(i)};\ \dot{y}\ ;\ \ddot{y}\big] \tag{2.63}$$

where $\ddot{y}$ is given in (2.61) and $\dot{I}^{(i)}$ is given in (2.38) with u equal to the right-hand side of (2.60). It is important to note that (2.63) is of different order and is not equivalent to the system (2.60)-(2.61).

These transverse coordinates (2.62) automatically generates a moving Poincarè section $\{S(t)\}_{t\in[0,T]}$ associated with the motion of the periodic orbit (2.25) on which $x_{\perp}^{(i)}(t)$ defines the location on the surface $S^{(i)}(t)$. By introducing the scalar $\psi^{(i)} := \psi^{(i)}(\theta,\dot{\theta})$ such that the target trajectory is

$$\left\{\psi^{(i)} = \psi_*^{(i)}(t), I^{(i)} = 0, y = \mathbf{0}_{(n-1)\times 1}, \dot{y} = \mathbf{0}_{(n-1)\times 1}\right\}$$

and, by the assumption that $\left(|\dot{q}_*|^2(t)+|\ddot{q}_*|^2(t)\right) > 0$ for $t \in [0,T]$, the function $\psi_*^{(i)}(t) := \psi_*^{(i)}(\theta_*(t),\dot{\theta}_*(t))$ monotonically changes with time. Then, the moving Poincarè surface is defined by [42]

$$S^{(i)}(t) := \left\{[\theta;\dot{\theta};y;\dot{y}] \in \mathbb{R}^{2n} :\ \psi^{(i)}(\theta,\dot{\theta}) = \psi_*^{(i)}(t)\right\} \tag{2.64}$$

for $t \in [0,T]$ and can be expressed in $[q;\dot{q}]$-coordinates. Therefore, if one can find an explicit expression for $\psi^{(i)}(\theta,\dot{\theta})$, one can determine which section $S^{(i)}(\tau)$ the system states belongs to, which in turn can be used to determine the time instant τ corresponding to the orbit (2.25). Thus, for the case of underactuation one, the system can be made time-invariant by introducing a projection operator

$$\mathcal{T} : [\theta,\dot{\theta}] \mapsto t, \quad t \in [0,T], \tag{2.65}$$

using the scalar function $\psi(\theta,\dot{\theta})$ [43, 11].

Transverse Linearization

For the transverse coordinates (2.62), it is possible to compute a transverse linearization of the transverse dynamics (2.63) analytically. The first step is to linearize $\dot{I}^{(i)}$ from (2.38) with u equal to the right-hand side of (2.60), around the solution

$$\theta = \theta_*(t), \quad \dot{\theta} = \dot{\theta}_*(t), \quad \ddot{\theta} = \ddot{\theta}_*(t), \quad y = \dot{y} = \mathbf{0}_{(n-1)\times 1}, \quad v = 0, \tag{2.66}$$

of (2.60)-(2.61). Since $\dot{I}^{(i)}$ can be written on the form

$$\dot{I}^{(i)} = f^{(i)}(\theta,\dot{\theta},y,\dot{y},I^{(i)},v), \tag{2.67}$$

where $f^{(i)}(\cdot)$ is given by

$$f^{(i)}(\theta,\dot{\theta},y,\dot{y},I^{(i)},v) = \frac{2\dot{\theta}}{\alpha_i(\theta)}\left[g_I^{(i)}I^{(i)} + g_y^{(i)}y + g_{\dot{y}}^{(i)}\dot{y} + g_v^{(i)}v - \beta_i(\theta)I^{(i)}\right], \tag{2.68}$$

the linearization around (2.66) yields

$$\frac{d}{dt}\bar{I}^{(i)} = a_{11}^{(i)}(t)\bar{I}^{(i)} + a_{12}^{(i)}(t)\bar{Y}_1 + a_{13}^{(i)}(t)\bar{Y}_2 + b_1^{(i)}(t)\bar{V}, \tag{2.69}$$

where the variables $\bar{I}^{(i)}$, $\bar{Y}_1$, $\bar{Y}_2$, $\bar{V}$ give the first order approximations for the deviations of $I^{(i)}(\cdot)$, $y(\cdot)$, $\dot{y}(\cdot)$ and v of (2.60)-(2.61) from the nominal solution (2.66). Herein, $a_{11}^{(i)}$, $a_{12}^{(i)}$, $a_{13}^{(i)}$, $b_1^{(i)}$ are given by

$$a^{(i)}(t) = 2\dot{\theta}_*(t)/\alpha_i(\theta_*(t)), \tag{2.70a}$$

$$a_{11}^{(i)}(t) = a^{(i)}(t)\Big(g_I^{(i)}\big(\theta_*(t), \dot{\theta}_*(t), \ddot{\theta}_*(t), \mathbf{0}_{(n-1)\times 1}, \mathbf{0}_{(n-1)\times 1}\big) - \beta_i(\theta_*(t))\Big), \tag{2.70b}$$

$$a_{12}^{(i)}(t) = a^{(i)}(t) g_y^{(i)}\big(\theta_*(t), \dot{\theta}_*(t), \ddot{\theta}_*(t), \mathbf{0}_{(n-1)\times 1}, \mathbf{0}_{(n-1)\times 1}\big), \tag{2.70c}$$

$$a_{13}^{(i)}(t) = a^{(i)}(t) g_{\dot{y}}^{(i)}\big(\theta_*(t), \dot{\theta}_*(t), \ddot{\theta}_*(t), \mathbf{0}_{(n-1)\times 1}, \mathbf{0}_{(n-1)\times 1}\big), \tag{2.70d}$$

$$b_1^{(i)}(t) = a^{(i)}(t) g_v^{(i)}\big(\theta_*(t), \dot{\theta}_*(t), \mathbf{0}_{(n-1)\times 1}, \mathbf{0}_{(n-1)\times 1}\big). \tag{2.70e}$$

The transverse linearization of y and $\dot{y}$, with y given in (2.61), yields the dynamics for $\bar{Y}_1$ and $\bar{Y}_2$ on the form

$$\begin{aligned} \dot{\bar{Y}}_1 &= \bar{Y}_2 \\ \dot{\bar{Y}}_2 &= A_{31}(t)\bar{I}^{(i)} + A_{32}(t)\bar{Y}_1 + A_{33}(t)\bar{Y}_2 + B_3(t)\bar{V} \end{aligned} \tag{2.71}$$

where the matrix functions $A_{31}(t)$, $A_{32}(t)$, $A_{33}(t)$ and $B_3(t)$ are found by linearizing (2.61) around the solution (2.66), i.e., [42, 12]

$$\big[A_{31}(t); A_{32}(t); A_{33}(t); B_3(t)\big] = \left[\frac{\dot{\theta}\frac{\partial F}{\partial \dot{\theta}} - \ddot{\theta}\frac{\partial F}{\partial \theta}}{2(\dot{\theta}^2 + \ddot{\theta}^2)}; \frac{\partial F}{\partial y}; \frac{\partial F}{\partial \dot{y}}; N\right]_{\substack{\theta=\theta_*(t),\\ \dot{\theta}=\dot{\theta}_*(t)\\ \ddot{\theta}=\ddot{\theta}_*(t)\\ y=\dot{y}=0}} \tag{2.72}$$

where the linearization resulting in the term $A_{31}(t)$ stems from the fact that the integral function $I(\cdot)$ measures the deviation from the nominal solution of the motion generator, and is orthogonal to its vector field; see [43].

The equations (2.69) and (2.71) are combined to form the linearized transverse dynamics on state space form next.

Theorem 2.4.5 (Linearized transverse dynamics)**.** *By introducing* $X^{(i)} = \big[\bar{I}^{(i)}; \bar{Y}_1; \bar{Y}_2\big]$ *and* $\bar{V}$*, where* $\bar{I}^{(i)}$*,* $\bar{Y}_1$*,* $\bar{Y}_2$*,* $\bar{V}$ *give the first order approximations for the deviations of* $I^{(i)}(\cdot)$*,* $y(\cdot)$*,* $\dot{y}(\cdot)$ *and* v *of* (2.60)-(2.61) *from the nominal solution* (2.66)*, the transverse linearization of the system* (2.60)-(2.61) *can be written as the linear time-varying system*

$$\dot{X}^{(i)} = \mathcal{A}^{(i)}(t)X^{(i)} + \mathcal{B}^{(i)}(t)\bar{V} \tag{2.73}$$

where the time-varying, T*-periodic matrices* $\mathcal{A}^{(i)}(t)$ *and* $\mathcal{B}^{(i)}(t)$ *are given by*

$$\mathcal{A}^{(i)}(t) = \begin{bmatrix} a_{11}^{(i)}(t) & a_{12}^{(i)}(t) & a_{13}^{(i)}(t) \\ \mathbf{0}_{(n-1)\times 1} & \mathbf{0}_{(n-1)\times(n-1)} & I_{(n-1)} \\ A_{31}(t) & A_{32}(t) & A_{33}(t) \end{bmatrix}, \quad \mathcal{B}^{(i)}(t) = \begin{bmatrix} b_1^{(i)}(t) \\ \mathbf{0}_{m\times m} \\ B_3(t) \end{bmatrix}, \tag{2.74}$$

with $a_{11}^{(i)}$*,* $a_{12}^{(i)}$*,* $a_{13}^{(i)}$*,* $b_1^{(i)}$ *defined in* (2.70) *and* A_{31}*,* A_{32}*,* A_{33} *and* B_3 *defined in* (2.72)*.*

2.4.4 Extension to Impulsive Mechanical Systems with One Jump

Given the hybrid system (2.19) with one jump, it has been shown in [41] that the method of virtual holonomic constraints can be used to both find and control periodic trajectories for such systems using the methods of the previous sections. This is clearly seen by the next theorem.

Theorem 2.4.6 ([44, 41])**.** *Given an n degree of freedom impulsive mechanical system of the form* (2.19) *which is underactuated by a degree of one, and suppose that for some control input $u = u_*(t) = u_*(t + T_h)$, the impulsive mechanical system has a periodic solution*

$$q = q_*(t) = q_*(t + T_h), \quad \forall t, \quad T_h > 0 \tag{2.75}$$

with only one jump, i.e., $[q_*(0+); \dot{q}_*(0+)] \in \Gamma_+, \quad [q_*(T_h-); \dot{q}_*(T_h-)] \in \Gamma_-$, $F([q_*(T_h-); \dot{q}_*(T_h-)]) = [q_*(0+); \dot{q}_*(0+)]$. Now suppose the continuous-in-time arc of (2.75) *admits a re-parametrization $q_*(t) = \phi(\theta_*(t))$ defined by* (2.28) *with $\mathcal{C}^2$-functions $\phi_1(\cdot), \dots, \phi_n(\cdot)$. Compute the continuous dynamics of* (2.19) *when these relations are kept invariant, i.e., compute the coefficients of the virtual limit system* (2.34)*. Then, by necessity, the algebraic equations*

$$\begin{aligned} &I\big(\theta_*(0), \dot{\theta}_*(0), \theta_*(T_h), \dot{\theta}_*(T_h)\big) = 0, \\ &F\Big([q_-; \dot{q}_-]\Big)\Big|_{\substack{q_- = \phi(\theta_*(T_h)) \\ \dot{q}_- = \phi'(\theta_*(T_h))\dot{\theta}_*(T_h)}} = [q_+; \dot{q}_+]\Big|_{\substack{q_+ = \phi(\theta_*(0)) \\ \dot{q}_+ = \phi'(\theta_*(0))\dot{\theta}_*(0)}} \end{aligned} \tag{2.76}$$

hold, where $I(\cdot)$ is defined in (2.36)*.*

Finding Cycles

The relations (2.76) of Theorem 2.4.6 can be used for planning cycles of the system (2.19), as stated in [44] and [41], as follows:

1) Let $P = (p_1, \dots, p_k)$ be a vector of parameters and θ be a scalar variable; choose a set of $\mathcal{C}^2$-smooth functions: $\Phi(\theta, P) = [\phi_1(\theta, P); \dots; \phi_n(\theta, P)]$;

2) Simplify the continuous dynamics of (2.19) under the assumption that the relations $q_1 = \phi_1(\theta, P), \dots, q_n = \phi(\theta, P)$ are kept invariant. This results in the family of 2-dimensional manifolds $Z(P) \subset TQ$ defined by

$$Z(P) = \big\{[q; \dot{q}] \in TQ : \ q = \Phi(\theta, P), \ \dot{q} = \Phi'(\theta, P)\dot{\theta}\big\} \tag{2.77}$$

where $\theta = \theta(t)$ is a solution of the reduced dynamics (2.34), and where the phase curves of the system

$$\alpha(\theta, P)\ddot{\theta} + \beta(\theta, P)\dot{\theta}^2 + \gamma(\theta, P) = 0 \tag{2.78}$$

fill out $Z(P)$. Following Theorem 2.4.6, for each choice of P, the virtual limit system (2.78) is integrable; i.e., for any solution $\theta = \theta(t, P)$ of (2.78) well-defined for $t = \tau_1 > 0$, the function $I(\cdot)$ defined in (2.36), satisfies

$$I\big(\theta(\tau_1, P), \dot{\theta}(\tau_1, P), \theta(0, P), \dot{\theta}(0, P)\big) = 0; \tag{2.79}$$

3) Define the the curves γ_-, γ_+ and the mapping $\mathcal{F}:\ \gamma_- \mapsto \Gamma_-$ such that

$$\gamma_+ = \Gamma_+ \cap Z(P),\ \ \gamma_- = \Gamma_- \cap Z(P),\ \ \mathcal{F}\Big([\theta;\dot\theta]\Big)\Big|_{[\theta;\dot\theta]\in\gamma_-} = F\Big([q;\dot q]\Big)\Big|_{\substack{q=\phi(\theta,P)\\ \dot q=\phi'(\theta,P)\dot\theta}}; \tag{2.80}$$

4) Search for parameters $P = P_*$ such that the following algebraic equations have a non-trivial solution:

$$I(a,b,x,y)=0,\ \ \mathcal{F}\Big([x;y]\Big)=[a;b],\ \ [a;b]\in\gamma_+,\ \ [x;y]\in\gamma_-,\ \ a,b,x,y\in\mathbb{R}. \tag{2.81}$$

If the search is successful and $\theta = \theta_*(t,P_*)$ is the solution of (2.78) with $P=P_*$ initiated at $\theta_*(0,P_*)=a$, $\dot\theta_*(0,P_*)=b$ such that $\theta_*(Th,P_*)=x$, $\dot\theta_*(Th,P_*)=y$ for some $T_h>0$, then the hybrid mechanical system, (2.19), has the hybrid cycle defined by

$$q_1=\phi_1\big(\theta_*(t,P_*),P_*\big),\ldots,q_n=\phi_n\big(\theta_*(t,P_*),P_*\big). \tag{2.82}$$

Transverse Linearization

If a T_h-periodic cycle on the form (2.82) is found, then the transverse linearization on the interval $[t_0,t_0+T_h]$ is found by the method of Theorem 2.4.5. However, since the system is hybrid with the update law $F:\ \Gamma_-\to\Gamma_+$, see Figure 2.6(a), the mapping $[d^{TS}F]:\ TS(0)\to TS(T_h)$ between the tangent planes of the moving Poincarè sections must be developed such that $X(t_0)=[d^{TS}F]X(t_0+T_h)$. The mapping $[d^{TS}F]$ is, however, not trivial as the switching surfaces Γ_- and Γ_+ might not correspond with the Poincarè sections $S(T_h)$ and $S(0)$, such that the respective tangent planes $T\Gamma_-$, $T\Gamma_+$ and $TS(T_h)$, $TS(0)$ are likely to not correspond as well [41].

In order to find the mapping $[d^{TS}F]$, the procedure proposed in [41] and [12] will here be used. The procedure is outlined as follows:

1. Define the linearized update map $dF:\ T\Gamma_-\to T\Gamma_+$ by evaluating the Jacobian of the the update map $F(q,\dot q)$ at $[q_*(T_h);\dot q_*(T_h)]\in\Gamma_-$:

$$dF:=\frac{\partial F}{\partial[q;\dot q]}\Big|_{\substack{q=q_*(T_h)\\ \dot q=\dot q_*(T_h)}}:\ T\Gamma_-\Big|_{\substack{q=q_*(T_h)\\ \dot q=\dot q_*(T_h)}}\to T\Gamma_+\Big|_{\substack{q=q_*(0)\\ \dot q=\dot q_*(0)}}. \tag{2.83}$$

2. Find the projection operators $P^+_{\vec n(0)}:\ T\Gamma_+\to TS(0)$ along the normal vector $\vec n(0)$ of $TS(0)$ and $P^-_{\vec n(T_h)}:\ TS(T_h)\to T\Gamma_-$ along the normal vector $\vec n(T_h)$ of $TS(T_h)$; see Figure 2.6(b).

3. Then the linearization of $F(q,\dot q)$ associated with the moving Poincarè section $\{S(t)\}_{t\in[0,T_h]}$ is given by the linear mapping operator

$$[d^{TS}F]:\ TS(T_h)\ni X(T_h)\mapsto X(0):=\Big[P^+_{\vec n(0)}\Big]dF\Big[P^-_{\vec n(T_h)}\Big]X(T_h)\in TS(0). \tag{2.84}$$

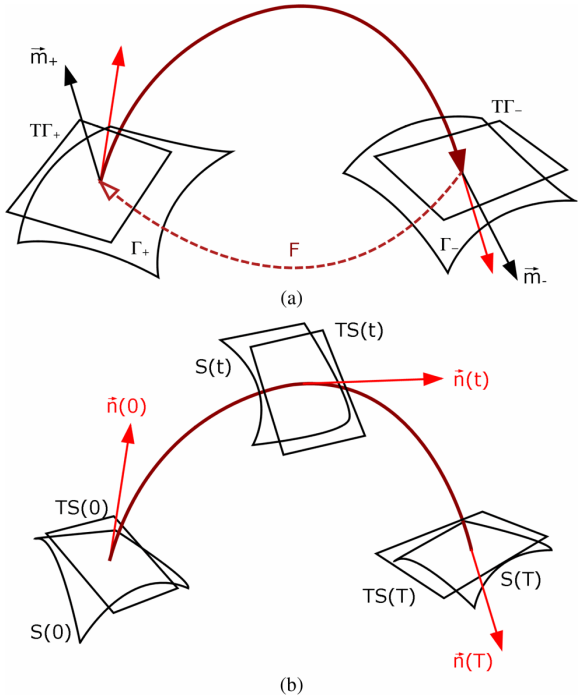


Figure 2.6: (a) Tangent planes $T\Gamma_-$ and $T\Gamma_+$ to the switching surfaces Γ_- and Γ_+ at two points, where the quasi-periodic trajectory hits Γ_- and is mapped to Γ_+ by the impact map $F(\cdot)$. The linearization of $F(\cdot)$ in a vicinity of the hybrid cycle is the linear mapping $dF : T\Gamma_- \to T\Gamma_+$. The vectors $\vec{m}_-$ and $\vec{m}_+$ denote the normals to $T\Gamma_-$ and $T\Gamma_+$, respectively; (b) Illustration of the moving Poincàre sections $S(t)$ and its tangent $TS(t)$, as well as its normal vector $\vec{n}(t)$. Reprinted from IEEE Transactions on Automatic Control, Vol. 54, No. 12, A.S. Shiriaev, L.B. Freidovich, *Transverse Linearization for Impulsive Mechanical Systems With one Passive Link,* pages 2882-2888, Copyright © 2009 IEEE.

In order to find this mapping, assume that there exists a linear mapping $\Lambda(t)$ between small increments of the transverse coordinates in terms of the state variables:

$$\Delta X(t) = \Lambda(t)\Delta x(t), \tag{2.85}$$

where $\Delta X(t) = [\Delta I; \Delta y; \Delta \dot{y}] \in TS(t) \cong \mathbb{R}^{2n-1}$ and $\Delta x(t) = [\Delta q(t); \Delta \dot{q}(t)] \in TS(t) \subset \mathbb{R}^{2n}$. However, since $\Lambda(t) \in \mathbb{R}^{(2n-1)\times 2n}$, it is not invertible, such that one can not find a linear mapping from small increments of the state variables in terms of the transverse coordinates by simply inverting $\Lambda(t)$. In order to achieve this, one can observe that the vector field of the nominal trajectory is orthogonal to $S(t)$ at time t [12], thus the normal vector is given by

$$\vec{n}(t) = [\dot{q}_*(t); \ddot{q}_*(t)] \tag{2.86}$$

such that $\vec{n}(\tau)^T \Delta x(\tau) = 0$. Hence, one can write

$$\Delta x(T_h) = \begin{bmatrix} \Lambda(T_h) \\ \vec{n}(T_h)^T \end{bmatrix}^{-1} \begin{bmatrix} \Delta X(T_h) \\ 0 \end{bmatrix}. \tag{2.87}$$

Now, since $\Delta x(T_h) \in TS(T_h)$, one must find its image δx_- on $T\Gamma_-$ by projecting it onto $T\Gamma_-$ using the projection operator $P^-_{\vec{n}(T_h)}$ by the following relation:

$$T\Gamma_- \ni \delta x_- := P^-_{\vec{n}(T_h)} \Delta X(T_h). \tag{2.88}$$

In order to find $P^-_{\vec{n}(T_h)}$, one can observe that δx_- can be written as

$$\delta x_- = \Delta x(T_h) + \vec{n}(T_h)\tau_{\Delta x(T_h)}, \tag{2.89}$$

where the scalar $\tau_{\Delta x(T_h)}$ denotes the length along the direction of $\vec{n}(T_h)$ from $\Delta x(T_h)$ which intersect with $T\Gamma_-$; see Figure 2.7. By noting that $\vec{m}_- \cdot \delta x_- = 0$, it follows that

$$\vec{m}_-^T \Delta x(T_h) + \vec{m}_-^T \vec{n}(T_h)\tau_{\Delta x(T_h)} = 0. \tag{2.90}$$

Hence,

$$\tau_{\Delta x(T_h)} = -\frac{\vec{m}_-^T \Delta x(T_h)}{\vec{m}_-^T \vec{n}(T_h)}, \tag{2.91}$$

which, as expected, clearly shows that $\tau_{\Delta x(T_h)} \equiv 0$ if $\vec{m}_- \parallel \vec{n}(T_h)$, i.e., if the hypersurfaces $T\Gamma_-$ and $TS(T_h)$ align. This yields the projection operator

$$P^-_{\vec{n}(T_h)} := \left(I_{2n} - \frac{\vec{n}(T_h)\vec{m}_-^T}{\vec{n}(T_h)^T \vec{m}_-} \right) \begin{bmatrix} \Lambda(T_h) \\ \vec{n}(T_h)^T \end{bmatrix}^{-1} \begin{bmatrix} I_{2n-1} \\ \mathbf{0}_{1\times 2n-1} \end{bmatrix}. \tag{2.92}$$

The next step is then to find $\delta x_+ \in T\Gamma_+$ in terms of δx_-, which is simply done by the Jacobian of the update law:

$$\delta x_+ := dF \delta x_- \tag{2.93}$$

with dF found from (2.83).

The last step is to find the operator $P^+_{\vec{n}(0)}$ which projects $\delta x_+ \in T\Gamma_+$ on to $TS(0)$ yielding $\Delta X(0)$. This is done by noting that $TS(0) \ni \Delta X(0) = \Lambda(0)\Delta x(0)$, where

$$\Delta x(0) := \delta x_+ - \text{Proj}_{\vec{n}(0)} \delta x_+,$$

and where $\text{Proj}_{\vec{n}(0)} \delta x_+$ is the projection of δx_+ along the normal $\vec{n}(0)$ of $TS(0)$ (see Figure 2.6(a)), which is defined as

$$\text{Proj}_{\vec{n}(0)} \delta x_+ := \frac{\vec{n}(0)^T \delta x_+}{\vec{n}(0)^T \vec{n}(0)} \vec{n}(0) = \frac{\vec{n}(0)\vec{n}(0)^T}{\vec{n}(0)^T \vec{n}(0)} \delta x_+. \tag{2.94}$$

Thus, one obtains

$$P^+_{\vec{n}(0)} := \Lambda(0) \left(I_{2n} - \frac{\vec{n}(0)\vec{n}(0)^T}{\vec{n}(0)^T \vec{n}(0)} \right), \tag{2.95}$$

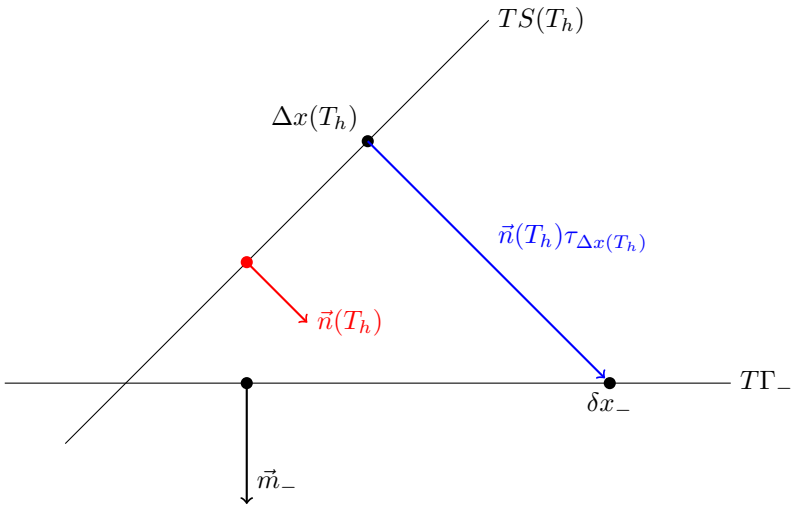


Figure 2.7: Projection from a point $\Delta x(T_h) \in TS(T_h)$ to a point $\delta x_- \in T\Gamma_-$ along the direction of $\vec{n}(T_h)$.

such that the complete map is of the form (2.84).

Thus, to summarize, the linearized transverse dynamics with one jump has the form

$$\begin{aligned} \dot{X} &= \mathcal{A}(\tau)X + \mathcal{B}(\tau)\bar{V}, \quad 0 \leq \tau < T_h, \\ X(0) &= dF^{TS}X(T_h), \end{aligned}$$

with τ computed $\tau = t$ moduloT_h.

2.5 Stabilizing Linear Time-Varying Systems

In this section, the topic will be stabilization of linear time-varying systems of the form

$$\dot{x}(t) = A(t)x(t) + B(t)u(t), \tag{2.96}$$

where $x(t) \in \mathbb{R}^n$ are the state variables, $u(t) \in \mathbb{R}^m$ are the actuator inputs, and $A(t)$ and $B(t)$ are $\mathcal{C}^0$-smooth $n \times n$ and $n \times m$ matrices on $t \in [t_0, T]$, respectively. In order to determine the stability of such a system, the notion of a *Hurwitz* matrix is given next.

Definition 4. *[56] The matrix function $A(t)$ is said to be Hurwitz[5] if there exists numbers $C > 0$, $\epsilon > 0$ such that for any solution $x = x(t)$ of the equation $\dot{x} = A(t)x$ one has*

$$|x(t)| \leq Ce^{-\epsilon(t-\tau)}|x(\tau)|, \quad t \geq \tau.$$

Moreover, if $A(t)$ is Hurwitz, then the system $\dot{x} = A(t)x$ is exponentially stable.

[5]It can be shown that any constant matrix A is Hurwitz if its eigenvalues lie in the left-half plane [23]. However, for the time-varying case, the matrix $A(t)$ is not necessarily Hurwitz on $[t_0, T]$ even though its eigenvalues lie in the left-half plane for all $t \in [t_0, T]$; see, e.g., [27, Theorem 7.4].

From this, it directly follows that if the matrix $A(t)$ of (2.96) is Hurwitz, then the origin is stable even when the system is unactuated. However, this is unlikely to be the case for most system. Thus, one needs to determine, by some manner, a control input $u(t)$ such that the closed-loop system is stable. In order to determine if such a control input is possible, one can use the notion of *controllability* stated next.

Theorem 2.5.1. *(Controllability, [39]) The system* (2.96) *is controllable on* $[t_0, T]$ *if and only if the* $n \times n$ *controllability Gramian*

$$W(t_0, T) = \int_{t_0}^{T} \Phi(t_0, t) B(t) B^T(t) \Phi^T(t_0, t) dt \tag{2.97}$$

is invertible[6]*. Herein, the* $n \times n$ *matrix* $\Phi(t, t_0)$ *is called the* transition matrix*, and is defined as* $\Phi(t, t_0) = X(t)$*, where* $X(t)$ *found from the* $n \times n$ *matrix differential equation*

$$\dot{X}(t) = A(t)X(t), \quad X(t_0) = I_n. \tag{2.98}$$

The implications of Theorem 2.5.1 follows next.

Theorem 2.5.2. *(Existence of stabilizing controller, [56, 39]) If the system* (2.96) *is controllable on* $[t_0, T]$*, then there exists a continuous* $m \times n$ *matrix function* $K(t)$ *such that the controller*

$$u(t) = K(t)x(t) \tag{2.99}$$

makes $\big(A(t) + B(t)K(t)\big)$ *Hurwitz for all* $t \in [t_0, T]$*, which in turn makes the origin of closed-loop system exponentially stable.*

Although one can use Theorem 2.5.2 to determine if there exists a controller which makes the system (2.96) exponentially stable[7], the question of how to find the matrix $K(t)$ still remains. Indeed, even though a stabilizing matrix $K(t)$ is found, it might lead to unrealistic requirements of the actuators. For this reason, the method of minimizing a quadratic cost function, which penalize both deviations of the states from the origin and the actuator outputs, leading to the *linear-quadratic regulator* (LQR), will be presented next.

The Linear-Quadratic Regulator Method

The linear-quadratic regulator (LQR) is a well known control strategy in the field of optimal control. The method is based on minimizing the quadratic cost functional

$$J\big(x(t_0), u(t), t_0, t_e\big) = \int_{t_0}^{t_e} \Big[x^T(t)Q(t)x(t) + u^T(t)\Gamma(t)u(t)\Big]dt + x^T(t_e)Sx(t_e), \tag{2.100}$$

[6]Note that the invertibility of the controllability Gramian of (2.97) is equivalent to the positive definiteness condition [56]

$$\int_{t_0}^{t_e} \big[X(t)^{-1}\big] B(t) B^T(t) \big[X(t)^{-1}\big]^T dt \succ 0.$$

[7]The controllability condition is only a sufficient condition for there to exists an exponentially stabilizing controller, and not a necessary condition. The necessary condition is $\mathcal{L}_2$-stabilizability of (2.96), which controllability implies; see, e.g., [56].

where the term $x^T(t)Q(t)x(t)$ penalize the deviation of the states from the origin, $u^T(t)\Gamma(t)u(t)$ penalize the actuator outputs and $x^T(t_e)Sx(t_e)$ penalize the final deviation of the states. Herein, $Q(t) \in \mathbb{R}^{n\times n}$ and $\Gamma(t) \in \mathbb{R}^{m\times m}$ are contiuous-in-time, symmetric positive-semidefinite (SPSD) and symmetric positive-definite (SPD) matrix functions, respectively, and $S \in \mathbb{R}^{n\times n}$ is SPSD. The reader interested in a more in-depth derivation of the method, and optimal control in general, is referred to the seminal paper [22] by Kalman.

It can be shown that the optimal[8] controller $u^0(t)$ which minimizes the cost function (2.100) and exponentially stabilizes the system (2.96) is given by [1, 22]

$$u^o(t) = -\Gamma(t)^{-1}B^T(t)R(t)x(t), \tag{2.101}$$

where $R(t)$ is a twice continuously differentiable, $n \times n$ symmetric matrix function which is the solution of the *Riccati differential equation* (RDE)

$$\dot{R}(t) + A^T(t)R(t) + R(t)A(t) + Q(t) = R(t)B(t)\Gamma(t)^{-1}B^T(t)R(t), \tag{2.102}$$

with the boundary condition $R(t_e) = S$. Moreover, the optimal cost functional $J^o\big(x(t), t, t_e\big) := J\big(x(t), u^o(t), t, t_e\big)$ which is the minimum of (2.100) subject to $u^o(t)$ for $t \in [t_0, t_e]$, has the quadratic form

$$J^o\big(x(t), t, t_e\big) = x^T(t)R(t)x(t), \tag{2.103}$$

referred to as the *cost-to-go*, i.e., the remaining cost from time t until t_e when subject to the optimal control input (2.101). Hence, the minimal of the cost functional (2.100) has the form

$$J^o\big(x(t_0), t_0, t_e\big) = x^T(0)R(0)x(0). \tag{2.104}$$

This is not difficult to see by defining

$$V(x(t), t) := x^T(t)R(t)x(t), \tag{2.105}$$

whose derivative w.r.t. time is

$$\begin{aligned}
\dot{V}(x(t), t) &= \dot{x}^T R(t)x(t) + x^T R(t)\dot{x}^T(t) + x^T\dot{R}(t)x(t \\
&= x^T(t)\Big[A^T(t)R(t) + R(t)A(t) + \dot{R}(t) - 2R(t)B(t)\Gamma(t)^{-1}B^T(t)R(t)\Big]x(t) \\
&= x^T(t)\Big[- Q(t) - R(t)B(t)\Gamma(t)^{-1}B^T(t)R(t)\Big]x(t).
\end{aligned}$$

Inserting the controller (2.101) into (2.100) yields

$$\begin{aligned}
J^o\big(x(t_0), t_0, t_e\big) &= \int_{t_0}^{t_e}\Big[x^T(t)Q(t)x(t) + R(t)B(t)\Gamma(t)^{-1}B^T(t)R(t)\Big]dt + x^T(t_e)Sx(t_e) \\
&= \int_{t_0}^{t_e} -\dot{V}(x, t)dt + x^T(t_e)Sx(t_e) \\
&= V(x(t_0), t_0) - V(x(t_e), t_e) + x^T(t_e)Sx(t_e).
\end{aligned}$$

[8] The superscript "o" identifies optimality.

Therefore, from the boundary condition $R(t_e) = S$, it is clear that $x^T(0)R(0)x(0)$ is the minimum of the cost functional. From this, it further follows that $R(t)$ must be at least a positive-semidefinite matrix function for all $t \in [t_0, T]$ as $J\big(x(t), u(t), t, t_e\big) \geq 0$ by definition.

Solving the Periodic Riccati differential equation

Solving the Riccati differential equation (2.102) is no easy task. Indeed, finding numerically reliable and efficient algorithms is still an active research subject. However, most of the proposed solutions assume time-invariant systems; see, e.g, [35, 8]. For this reason, a method proposed in [17, 18] for solving the *periodic Riccati differential equation* (PRDE) will be presented in this text.

Suppose that the matrices of (2.96) are T-periodic, i.e., $A(t) = A(t+T)$ and $B(t) = B(t+T)$. The stabilizing solution is then of form (2.101) with $R(t) = R(t+T)$, which is the solution of the periodic Riccati differential equation $\mathcal{R}\big(R(t), t\big) = 0$, where

$$\mathcal{R}\big(R(t), t\big) = \dot{R}(t) + A^T(t)R(t) + R(t)A(t) + Q(t) - R(t)B(t)\Gamma(t)^{-1}B^T(t)R(t). \tag{2.106}$$

It was found in [4] that solution of $\mathcal{R}\big(R(t), t\big) = 0$ is unique, as well as the *maximal solution*. The exact meaning of this is clarified in the next Proposition.

Proposition 2.5.3 (Maximal solution of the PRDE, [18])**.** *Let $A(t)$, $B(t)$, $Q(t)$ and $\Gamma(t)$ be continuous, T-periodc matrix functions; the pair (A, B) be stabilizable; and $Q(t) \succ 0$, $\Gamma(t) \succ 0$, $\forall t \in [0, T]$. Then, there exists a unique, T-periodic stabilizing solution $R_+(t)$ such that $\mathcal{R}\big(R_+(t), t\big) = 0$. Moreover, any T-periodic solution $R(t)$ of the Riccati inequality*

$$\mathcal{R}\big(R(t), t\big) \succeq 0, \quad \forall t \in [0, T], \tag{2.107}$$

satisfies the inequality

$$R_+(t) - R(t) \succeq 0, \quad \forall t \in [0, T]. \tag{2.108}$$

The inequality (2.107) can be transformed into a *linear matrix inequality* (LMI) by taking the Schur complement[9] of $\Gamma(t)$:

$$\bar{\mathcal{R}}\big(R(t), t\big) = \begin{bmatrix} \dot{R}(t) + A^T(t)R(t) + R(t)A(t) + Q(t) & R(t)B(t) \\ B^T(t)R(t) & \Gamma(t) \end{bmatrix} \succeq 0, \quad \forall t \in [0, T]. \tag{2.109}$$

Thus, the stabilizing solution satisfies $\bar{\mathcal{R}}\big(R(t), t\big) \succeq 0$ and maximizes the functional

$$\mathcal{J}\big(R(t)\big) = \int_0^T \mathrm{tr}\big(R(t)\big)dt, \tag{2.110}$$

where $\mathrm{tr}(\cdot)$ denotes the trace of the matrix functions, i.e., the sum of the elements on the main diagonal.

[9] For a matrix $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$, the Schur complement of A_{22} is given by $A_c = A_{11} - A_{12}A_{22}^{-1}A_{21}$. Moreover, if $A_{22} \succ 0$ and A_c is positive (semi-)definite, then A is positive (semi-)definite [58].

In order to transform this from a infinite problem to a finite, one can approximate the matrix function $R(t)$ with a symmetric, T-periodic trigonometric polynomial:

$$\hat{R}(t) = R_{a,0} + \sum_{k=1}^{N} \Big(\cos(k\omega t) R_{a,k} + \sin(k\omega t) R_{b,k} \Big), \tag{2.111}$$

with $\omega = 2\pi/T$ and the $2N+1$ coefficients $R_{a,k}$ and $R_{b,k}$. Thus, by taking a number of L evenly spaced time-samples, $t_1, t_2, \ldots, t_L$, one can find an approximations $\bar{R}_+(t)$ of the stabilizing solution by solving the semi-definite programming (SDP) problem:

$$\begin{aligned} \underset{\bar{R}(t)}{\text{minimize}} \quad & -\bar{\mathcal{J}}(\bar{R}(t)) \\ \text{subject to} \quad & S_j \succeq 0, \;\; j = 1, ..., L, \end{aligned} \tag{2.112}$$

where

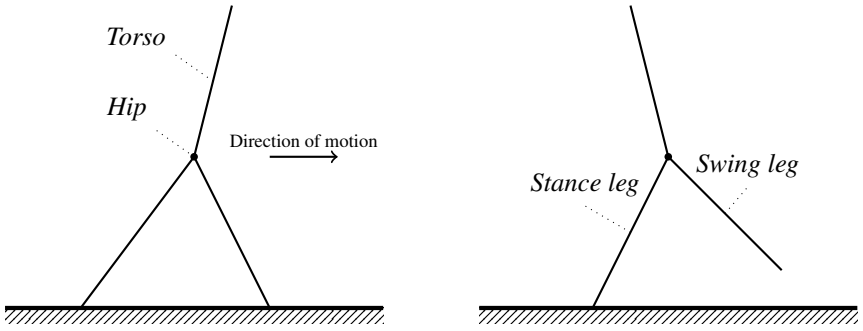
$$\bar{\mathcal{J}}(\bar{R}(t)) = \sum_{j=1}^{L} tr \left(R_{a,0} + \sum_{n=1}^{N} \Big(\cos(n\omega t_j) R_{a,n} + \sin(n\omega t_j) R_{b,n} \Big) \right)$$

and $S_j = S(\bar{R}(t_j), t_j)$, for some sufficiently large integers N and L.

Chapter 3

Equations of Motion and Impact Models for the Three-Link Biped

In this chapter, the mathematical model describing the planar compass biped will be derived. The compass biped considered in this text, is an open kinematic chain consisting of two sub-chains, called the *legs*, connected to another sub-chain called the *torso* through a revolute joint referred to as the *hip*. When walking, the biped switches between two phases: *double support*, when both legs are in contact with the ground; and *single support*, or the *swing phase*, where only one of the legs, referred to as the *stance leg*, is in contact with the ground and where the other leg is referred to as the *swing leg*. These phases are depicted **(a)** and **(b)** in Figure 3.1, respectively. The biped is thus a hybrid system where the dynamics in the swing phase can be modelled using the Euler-Lagrange equation, which will be done in Section 3.1, while the dynamics during double support, which will be called the impact phase, is discussed in Section 3.2.



(a) Double support. **(b)** Swing phase.

Figure 3.1: Phases of the three-link biped with torso shown in the saggital plane.

For walking, a few standard assumptions will be made [55, 49] for the biped during

the gait:

- During the swing phase, the stance leg acts like a pivot, i.e., its tip can be regarded as being fixed to the ground.
- The collision of the swing leg with the ground at impact is perfectly inelastic, i.e., no rebound; no slipping occurs and the swing leg becomes the new stance leg.
- At impact, the stance leg detaches from the ground and becomes the new swing leg.
- The gait is symmetric, i.e., the configuration is identical at the beginning of each consecutive step with the exception that the swing- and stance leg are switched.
- The biped can only move in the *sagittal plane*, i.e., from left to right; see Figure 3.1a.

Another common assumption is that the impact can be modelled as an instantaneous event [55, 49, 26]. This will be discussed further in Section 3.2.

The biped consists of three connected links, each of which can be considered to distributed mass with the *center of mass* (COM) given for each link. Thus, a minimum of three generalized coordinates are needed to describe the configuration of the robot, while two more are needed in order to determine its position in the inertial (World) frame. Furthermore, the biped is powered by two actuators (motors) connected at the hip, each acting between the torso and one of the legs. These are denoted by the vector $u = [u_1, u_2]^T$ where u_1 is connected between the torso and stance leg, while u_2 is connected between the torso and the swing leg.

The biped has a total of five degrees of freedom: the rotation of each link and the x-y coordinates of some point on the biped, e.g., its center of mass or the hip, measured in the inertial frame. Since it only has two actuators, it is thus underactuated by a degree of three. However, since the assumption that the stance leg does not move during the swing phase and acts like a pivot, one can describe the configuration of the biped in each phase by the rotations of the link and move the inertial frame from, e.g., the stance leg to the new stance leg after each transition phases. Therefore, one can reduced the degrees of freedom to three, i.e., the configuration, such that it is underactuated by a degree of one.

3.1 Equations of Motion in the Swing Phase

In this section, the equations of motion in the swing phase will be derived using the Euler-Lagrange equation. Each link of the biped is considered to have distributed mass, i.e., it is not modelled as a point mass. For each link, $i = 1, 2, 3$, the constant scalar m_i denotes the mass of the link i located at its centre of mass, J_i denotes its moment of inertia, L_i denotes the link's length and l_i denotes the length to its center of mass from the previous frame. A schematic of the biped is shown in Figure 3.2 where the masses and the inertias of the torso and the hip are combined into m_2 and J_2, respectively.

The configuration of the biped can be described using several conventions for the angles. In this text, two conventions will be presented: one using absolute angles and one

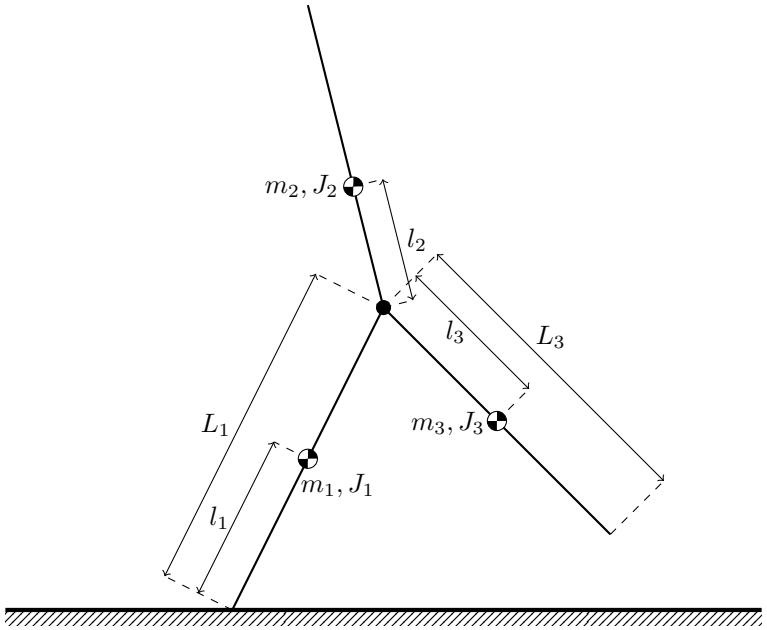


Figure 3.2: Masses, inertias and link lengths of the three-link biped.

using relative angles. These are shown in Figure 3.3 . Herein, the absolute angles are given in terms of the relative angles by

$$\begin{bmatrix} q_1^a \\ q_2^a \\ q_3^a \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} q_1^r \\ q_2^r \\ q_3^r \end{bmatrix} + \begin{bmatrix} -\pi/2 \\ -\pi/2 \\ -3\pi/2 \end{bmatrix}, \tag{3.1}$$

hence the relative angles can be found in terms of the absolute by

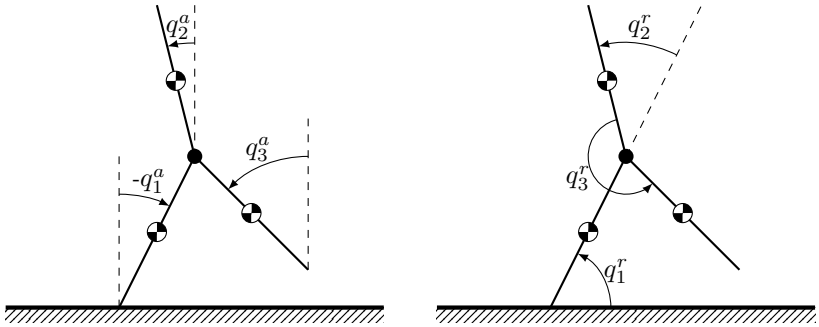
$$\begin{bmatrix} q_1^r \\ q_2^r \\ q_3^r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} q_1^a \\ q_2^a \\ q_3^a \end{bmatrix} + \begin{bmatrix} \pi/2 \\ 0 \\ \pi \end{bmatrix}. \tag{3.2}$$

In this text, the relative coordinates will be used in order to derive the equations of motion. However, one can simply find these in absolute coordinates by the coordinate transformation (3.1).

Position and velocity vectors using relative coordinates

Using relative angles and the coordinate convention in Figure 3.3 with frame assignment as in Figure 3.4, the position vectors in the inertial frame o_0 to the center mass of each of the links can be found using the homogeneous transformation

$$T_i^{i-1} = \begin{bmatrix} R_i^{i-1} & o_i^{i-1} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{3.3}$$



(a) Absolute coordinates. **(b)** Relative coordinates.

Figure 3.3: Coordinate conventions.

between frame o_{i-1} and o_i, where R_i^{i-1} denotes the rotation between the frames around the z_{i-1}-axis and o_i^{i-1} denotes the translation along the x_i-axis. Using the Denavit-Hartenberg convention [51], the transformation matrix between frame $i-1$ and i is given by

$$T_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i & 0 & a_i\sin\theta_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

where θ_i is the angle between the x_{i-1}- and x_i-axis measured around the z_{i-1}-axis, and a_i is the distance between the z_{i-1}- and z_i-axis measured along the x_i-axis. Using this, the distance from the frame o_0 to the frame o_i is given by the vector $p_i = 0_i^0$, where 0_i^0 is extracted from

$$T_i^0 = T_1^0 \cdots T_i^{i-1} \tag{3.5}$$

Using Figure 3.4, and by setting $q := [q_1; q_2; q_3] = [q_1^r; q_2^r; q_3^r]$, the homogeneous transformation matrices between the frames are found to be

$$T_1^0 = \begin{bmatrix} \cos q_1 & -\sin q_1 & 0 & l_1\cos q_1 \\ \sin q_1 & \cos q_1 & 0 & l_1\sin q_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_H^0 = \begin{bmatrix} \cos q_1 & -\sin q_1 & 0 & L_1\cos q_1 \\ \sin q_1 & \cos q_1 & 0 & L_1\sin q_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_2^H = \begin{bmatrix} \cos q_2 & -\sin q_2 & 0 & l_2\cos q_2 \\ \sin q_2 & \cos q_2 & 0 & l_2\sin q_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_3^H = \begin{bmatrix} \cos(q_2+q_3) & -\sin(q_2+q_3) & 0 & l_3\cos(q_2+q_3) \\ \sin(q_2+q_3) & \cos(q_2+q_3) & 0 & l_3\sin(q_2+q_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.6}$$

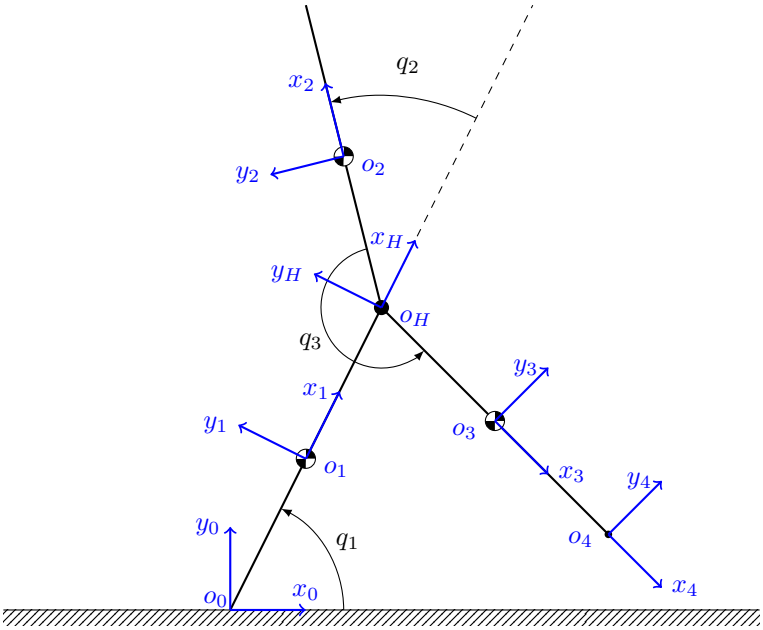


Figure 3.4: Coordinate and frame convention using relative angles.

Thus, $T_2^0 = T_H^0 T_2^H$ and $T_3^0 = T_H^0 T_3^H$, giving, after some simplification, the positions vectors

$$\vec{p}_1 = o_1^0 = \begin{bmatrix} l_1 \cos q_1 \\ l_1 \sin q_1 \end{bmatrix}, \tag{3.7a}$$

$$\vec{p}_2 = o_2^0 = \begin{bmatrix} L_1 \cos q_1 + l_2 \cos(q_1 + q_2) \\ L_1 \sin q_1 + l_2 \sin(q_1 + q_2) \end{bmatrix}, \tag{3.7b}$$

$$\vec{p}_3 = o_3^0 = \begin{bmatrix} L_1 \cos q_1 + l_3 \cos(q_1 + q_2 + q_3) \\ L_1 \sin q_1 + l_3 \sin(q_1 + q_2 + q_3) \end{bmatrix}. \tag{3.7c}$$

From this, the velocities are simply found from the time-derivative of the positions using the Jacobian $\mathbf{J}_i(q)$ as follows:

$$\vec{v}_i = \frac{d\vec{p}_i}{dt} = \frac{\partial \vec{p}_i}{\partial q}\dot{q} = \mathbf{J}_i(q)\dot{q}, \quad i = 1, 2, 3.$$

Hence, the velocities are given by

$$\vec{v}_1 = \mathbf{J}_1\dot{q} = \begin{bmatrix} -l_1\sin q_1 & 0 & 0 \\ l_1\cos q_1 & 0 & 0 \end{bmatrix}\dot{q}, \tag{3.8a}$$

$$\vec{v}_2 = \mathbf{J}_2\dot{q} = \begin{bmatrix} -L_1\sin q_1 - l_2\sin(q_1+q_2) & -l_2\sin(q_1+q_2) & 0 \\ L_1\cos q_1 + l_2\cos(q_1+q_2) & l_2\cos(q_1+q_2) & 0 \end{bmatrix}\dot{q}, \tag{3.8b}$$

$$\vec{v}_3 = \mathbf{J}_3\dot{q} = \begin{bmatrix} -L_1\sin q_1 - l_3\sin_{123} & -l_3\sin_{123} & -l_3\sin_{123} \\ L_1\cos q_1 + l_3\cos_{123} & l_3\cos_{123} & l_3\cos_{123} \end{bmatrix}\dot{q}, \tag{3.8c}$$

where $\sin_{123} = \sin(q_1+q_2+q_3)$ and $\cos_{123} = \cos(q_1+q_2+q_3)$.

Kinetic and Potential Energy

The kinetic energy of link i is found using

$$\mathcal{K}_i = \frac{1}{2}m_i\vec{v}_i\cdot\vec{v}_i + \frac{1}{2}J_i\omega_i^2 = \frac{1}{2}m_i\dot{q}^T\mathbf{J}_i(q)^T\mathbf{J}_i(q)\dot{q} + \frac{1}{2}J_i\omega_i^2 \tag{3.9}$$

where $\omega_i = \sum_{k=1}^{i}\dot{q}_i$. Thus, the kinetic energy of the biped is given by

$$\mathcal{K}(q,\dot{q}) = \mathcal{K}_1 + \mathcal{K}_2 + \mathcal{K}_3, \tag{3.10}$$

where

$$\mathcal{K}_1 = \frac{1}{2}\dot{q}_1^2(l_1^2m_1 + J_1), \tag{3.11a}$$

$$\begin{aligned}\mathcal{K}_2 =& \frac{1}{2}\dot{q}_1^2\big(2L_1l_2m_2\cos q_2 + L_1^2m_2 + l_2m_2 + J_2\big) + \frac{1}{2}\dot{q}_2^2\big(l_2^2m_2 + J_2\big) \\ &+ \dot{q}_1\dot{q}_2\big(L_1l_2m_2\cos q_2 + l_2^2m_2 + J_2\big),\end{aligned} \tag{3.11b}$$

$$\begin{aligned}\mathcal{K}_3 =& \frac{1}{2}\dot{q}_1^2\big(2L_1l_3m_3\cos(q_2+q_3) + L_1^2m_3 + l_3m_3 + J_3\big) \\ &+ \frac{1}{2}\big(\dot{q}_3^2 + \dot{q}_2^2\big)\big(l_3^2m_3 + J_3\big) + \big(l_3^2m_3 + J_3\big)\dot{q}_2\dot{q}_3 \\ &+ \big(L_1l_3m_3\cos(q_2+q_3) + l_3^2m_3 + J_3\big)\big(\dot{q}_2 + \dot{q}_3\big)\dot{q}_1.\end{aligned} \tag{3.11c}$$

The potential energy of the biped is the sum of the potential energy of the links, $V(q) = \mathcal{P}_1 + \mathcal{P}_2 + \mathcal{P}_3$, where the potential energy of each link can be computed as

$$\mathcal{P}_i = m_igh_i \tag{3.12}$$

where g is the gravitational acceleration and h_i denotes the height of the centre of mass of joint i computed along the y-axis of the inertial frame, which corresponds to the second element of the position vectors in (3.7). Thus, the potential energy of the biped is given by

$$V(q) = (L_1gm_2 + L_1gm_3 + gl_1m_1)\sin q_1 + gl_2m_2\sin(q_2+q_1) + gl_2m_3\sin(q_1+q_2+q_3). \tag{3.13}$$

The equation of motion can now be found by defining the Lagrangian

$$\mathcal{L}(q, \dot{q}) := \mathcal{K} - V(q)$$

and using the Euler-Lagrange equation (2.8). In order to state the equation on the form (2.9), i.e.,

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = B(q)u, \tag{3.14}$$

one must find the matrix functions $M(q)$, $C(q, \dot{q})$, $G(q)$ and $B(q)$. This is done next.

The Inertia Matrix

The elements of the inertia matrix $M(q)$ are found using (2.7), such that the i-jth elements are given as

$$\begin{aligned}
m_{1,1}(q) =& m_1 l_1^2 + 2L_1 l_2 m_2 \cos(q_2) + L_1^2 m_2 + l_2^2 m_2 + 2L_1 l_3 m_3 \cos(q_2 + q_3) && \text{(3.15a)}\\
&+ L_1^2 m_3 + l_3^2 m_3 + J_1 + J_2 + J_3,\\
m_{1,2}(q) =& m_{2,1}(q) = L_1 l_2 m_2 \cos(q_2) + L_1 l_3 m_3 \cos(q_2 + q_3) + l_2^2 m_2 + l_3^2 m_3 && \text{(3.15b)}\\
&+ J_2 + J_3,\\
m_{1,3}(q) =& m_{3,1}(q) = L_1 l_3 m_3 \cos(q_2 + q_3) + l_3^2 m_3 + J_3, && \text{(3.15c)}\\
m_{2,2}(q) =& l_2^2 m_2 + l_3^2 m_3 + J_2 + J_3 && \text{(3.15d)}\\
m_{2,3}(q) =& m_{3,2}(q) = l_3^2 m_3 + J_3, && \text{(3.15e)}\\
m_{3,3}(q) =& l_3^2 m_3 + J_3. && \text{(3.15f)}
\end{aligned}$$

The Coriolis and Centrifugal Matrix

Using (2.10), the non-zero elements of the matrix function $C(q, \dot{q})$ are found to be

$$\begin{aligned}
c_{1,1}(q, \dot{q}) =& - L_1 l_2 m_2 \sin(q_2)\dot{q}_2 - L_1 l_3 m_3 \sin(q_2 + q_3)\big(\dot{q}_2 + \dot{q}_3\big), && \text{(3.16a)}\\
c_{1,2}(q, \dot{q}) =& - L_1 l_2 m_2 \sin(q_2)\big(\dot{q}_1 + \dot{q}_2\big) - L_1 l_3 m_3 \sin(q_2 + q_3)\big(\dot{q}_1 + \dot{q}_2 + \dot{q}_3\big) && \text{(3.16b)}\\
c_{1,3}(q, \dot{q}) =& - L_1 l_3 m_3 \sin(q_2 + q_3)\big(\dot{q}_1 + \dot{q}_2 + \dot{q}_3\big), && \text{(3.16c)}\\
c_{2,1}(q, \dot{q}) =& \big(L_1 l_2 m_2 \sin(q_2) + L_1 l_3 m_3 \sin(q_2 + q_3)\big)\dot{q}_1, && \text{(3.16d)}\\
c_{3,1}(q, \dot{q}) =& L_1 l_3 m_3 \sin(q_2 + q_3)\dot{q}_1. && \text{(3.16e)}
\end{aligned}$$

The Gravity Vector

The gravitational vector is computed as $G(q) = [\partial V(q)/\partial q_1, \ldots, \partial V(q)/\partial q_n]^T$, yielding

$$G(q) = \begin{bmatrix} g_1(q) \\ g_2(q) \\ g_3(q) \end{bmatrix} \tag{3.17}$$

where

$$\begin{aligned} g_1(q) =& g\big(L_1\cos(q_1)m_2 + L_1\cos(q_1)m_3 + \cos(q_1+q_2)l_2m_2 \\ &+ \cos(q_1+q_2+q_3)l_3m_3 + \cos(q_1)l_1m_1\big), \end{aligned} \tag{3.18a}$$

$$g_2(q) = g\big(l_2m_2\cos(q_1+q_2) + l_3m_3\cos(q_1+q_2+q_3)\big), \tag{3.18b}$$

$$g_3(q) = gl_3m_3\cos(q_1+q_2+q_3). \tag{3.18c}$$

The Coupling Matrix

There are two actuators: one situated between the stance leg and the torso, and another between the torso and the swing leg. The torques generated are defined to be positive in the counter-clockwise direction. Thus, in order to find the coupling matrix $B(q)$, one can start by noting that for the case of relative angles, a virtual displacement δq_1 generates no virtual work about either of the actuators, while the displacements δq_2 and δq_3 generates work about u_1 and u_2, respectively. Hence by D'Alembert's principle, the coupling matrix for the relative case is

$$B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{3.19}$$

Note that if one wants to find the coupling matrix when using absolute angles, one can use (3.19) and the relations (3.2) to find that it is given by [54]

$$B_a = \left(\frac{\partial}{\partial q^a}\begin{bmatrix} q_2^r \\ q_3^r \end{bmatrix}\right)^T = \begin{bmatrix} -1 & 0 \\ 1 & -1 \\ 0 & 1 \end{bmatrix}. \tag{3.20}$$

3.2 Impact models

Whenever the end of the swing leg hits the ground, an impact will occur. This is equivalent to $q \in \Gamma_-$, where Γ_- is the impact surface which given by

$$\Gamma_- = \big\{q \in Q :\ L_1\sin q_1 + L_3\sin(q_1+q_2+q_3) = 0\big\}. \tag{3.21}$$

When an impact occurs, the previous swing leg should become the stance leg and vice versa. For the case of absolute angles, this gives the straight forward mapping

$$\begin{bmatrix} q_{1+}^a \\ q_{2+}^a \\ q_{3+}^a \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} q_{1-}^a \\ q_{2-}^a \\ q_{3-}^a \end{bmatrix}. \tag{3.22}$$

To finds this mapping for the case of relative angles, one can use this relation together with (3.1) to find

$$\begin{bmatrix} q_{1+}^a \\ q_{2+}^a \\ q_{3+}^a \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} q_{1-} \\ q_{2-} \\ q_{3-} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} -\pi/2 \\ -\pi/2 \\ -3\pi/2 \end{bmatrix}. \tag{3.23}$$

Hence, by plugging this into (3.2), the relative angles before and after impact are given by the mapping

$$q_+ = \begin{bmatrix} q_{1+} \\ q_{2+} \\ q_{3+} \end{bmatrix} = P_q q_- + c = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} q_{1-} \\ q_{2-} \\ q_{3-} \end{bmatrix} + \begin{bmatrix} -\pi \\ \pi \\ \pi \end{bmatrix}. \tag{3.24}$$

For the velocities, one needs to take into account that the impact gives rise sudden changes in the velocities, thus one needs to develop an impact model which gives a good approximation of this change. Such impact models can either be discrete models or continuous models. For the discrete models, it is assumed that the impact takes place over an infinitesimally small period of time such that the velocity mapping takes the form

$$\dot{q}_+ = P_{\dot{q}}(q_-)\dot{q}_-, \tag{3.25}$$

where $P_{\dot{q}}(q_-) = P_q \hat{P}_{\dot{q}}(q_-)$ with the relabelling matrix P_q from (3.24) and where $\hat{P}_{\dot{q}}(q_-)$ denotes the change of velocities due to impact before the relabelling of the coordinates. Such discrete models normally only give a good approximation for rigid bodies with stiff impacts [13].

In continuous models, the contact normal forces due to the impact are added to (3.14) during the duration of the contact for the bodies which impacts. Such models have some advantages over the discrete models, mainly that they can take into account both contact which takes place over a very short duration of time, i.e., impacts, as well as contact over longer duration and the use of several friction models [13]. However, such models often require several parameters dependent on the bodies in contact which can be difficult to obtain and might need to be tuned in order to get a realistic approximation of the impact [13]. For this reason, only discrete methods will be used in this thesis.

In the next sections, two models for $P_{\dot{q}}(q_-)$ will be derived, both of which are linear and modelled as discrete mappings. The first is an algebraic method based on the kinematic definition of restitution, i.e., Newton's method of the coefficient of restitution, while the other is based on the conservation of angular momentum. Note that both models result in exactly the same velocity mapping $P_{\dot{q}}(q_-)$.

3.2.1 Algebraic Method with Newton's Model of Restitution

In [19], an algebraic formulation of an impact model for kinematic chains was presented using Newton's model of restitution, which has been used for a three-link biped in [16]. In order to use this model, one must use the excessive coordinates $q_e = [q_1; q_2; q_3; x; y]$ where x and y are the Cartesian coordinates of the end of the stance leg as seen in Figure 3.4. The equations of motions then take the form

$$M_e(q_e)\ddot{q}_e + C(q_e, \dot{q}_e)\dot{q}_e + G(q_e) = B_e(q_e)u + \delta F_{ext} \tag{3.26}$$

where δF_{ext} represents the external forces acting on the biped at the points of contact. Furthermore, the model uses the following assumptions [19, 16]:

1. The impact takes place over an infinitesimally small period of time.

2. The external forces during the impact can be represented as instantaneous impulses.

3. Impulsive forces may result in an instantaneous change in velocities of the generalized coordinates, but the positions remain continuous.

4. The torques supplied by the actuators are not impulsional.

By these assumptions, the generalized coordinates can be regarded as constants in the duration of the impact and all the non-impulsive terms of (3.26) can be neglected, such that by integrating it, one obtains

$$M_e(q_e)(\dot{q}_{e+} - \dot{q}_{e-}) = \int_{t_-}^{t_+} \delta F_{ext}(\tau) d\tau = F_{ext} \tag{3.27}$$

where $\dot{q}_{e+}$ and $\dot{q}_{e-}$ are the velocities immediately after and prior to the impact, respectively. Here, the inertia matrix $M_e(q_e)$ is found by the same method as in section 3.1 by adding the vector $\vec{p}_0 = [x; y]$ to each of the position vectors in (3.7). Its non-zero elements are given in (B.1) in Appendix B.

As the stance leg detaches after impact, the only external forces acting on the biped due to the impact are at the end tip of the swing leg, which position vector $\vec{p}_{sw}$ is given by

$$\vec{p}_{sw} = \begin{bmatrix} x + L_1 \cos q_1 + L_3 \cos(q_1 + q_2 + q_3) \\ y + L_1 \sin q_1 + L_3 \sin(q_1 + q_2 + q_3) \end{bmatrix}. \tag{3.28}$$

Thus, the external forces can be found as

$$\begin{aligned} F_{ext} &= \left(\frac{\partial \vec{p}_{sw}}{\partial q_e} \right)^T \begin{bmatrix} F^t \\ F^n \end{bmatrix} = E \begin{bmatrix} F^t \\ F^n \end{bmatrix} \\ &= \begin{bmatrix} -L_1 \sin q_1 + f(q) & f(q) & f(q) & 1 & 0 \\ -L_1 \cos q_1 + g(q) & g(q) & g(q) & 0 & 1 \end{bmatrix}^T \begin{bmatrix} F^t \\ F^n \end{bmatrix}, \end{aligned} \tag{3.29}$$

where F^t and F^n are the tangential and normal forces, respectively, acting on the tip of the swing leg. Here, $f(q) = -L_3 \sin(q_1 + q_2 + q_3)$ and $g(q) = L_3 \cos(q_1 + q_2 + q_3)$.

The unknown variables to be found are F^t, F^n and $\dot{q}_{e+}$ as the velocities, $\dot{q}_{e-}$, prior to the impact are known, with $\dot{x}_-$ and $\dot{y}_-$ equal to zero as the stance leg acts as a pivot before the impact. Furthermore, by the assumption that the swing does neither slip nor rebound, the relation

$$\left(\frac{\partial \vec{p}_{sw}}{\partial q_e} \right) \dot{q}_{e+} = 0 \tag{3.30}$$

must hold. Thus, the problem is reduced to seven unknowns, $\dot{q}_{e+}$, F^t and F^n, with seven linear equations given by (3.27) and (3.30), which combined give

$$\begin{bmatrix} M_e(q_e) & -E \\ E^T & \mathbf{0}_{2\times 2} \end{bmatrix} \begin{bmatrix} \dot{q}_{e+} \\ F^t \\ F^n \end{bmatrix} = \begin{bmatrix} M_e(q_e) \\ \mathbf{0}_{2\times 5} \end{bmatrix} \dot{q}_{e-}. \tag{3.31}$$

Since $M_e(q_e)$ is invertible and E has full rank, the left-hand matrix can be inverted such that the unknowns can be found by

$$\begin{bmatrix} \dot{q}_{e+} \\ F^t \\ F^n \end{bmatrix} = \begin{bmatrix} M_e(q_e) & -E \\ E^T & \mathbf{0}_2 \end{bmatrix}^{-1} \begin{bmatrix} M_e(q_e) \\ \mathbf{0}_{2\times 5} \end{bmatrix} \dot{q}_{e-}. \tag{3.32}$$

Thus, the impact velocity mapping in relative coordinates is given by

$$\dot{q}_+ = P_{\dot{q}}(q_-)\dot{q}_- = P_q \begin{bmatrix} I_3 & \mathbf{0}_{3\times 4} \end{bmatrix} \begin{bmatrix} M_e(q_e) & -E \\ E^T & \mathbf{0}_2 \end{bmatrix}^{-1} \begin{bmatrix} M_e(q_e) \\ \mathbf{0}_{2\times 5} \end{bmatrix} \begin{bmatrix} I_3 & \mathbf{0}_2 \\ \mathbf{0}_{2\times 3} & \mathbf{0}_2 \end{bmatrix} \dot{q}_-, \tag{3.33}$$

with P_q from (3.24). Note that the tangential and normal force can be found from the expression

$$\begin{bmatrix} F^t \\ F^n \end{bmatrix} = P_F(q_-)\dot{q}_- = \begin{bmatrix} \mathbf{0}_{2\times 5} & I_2 \end{bmatrix} \begin{bmatrix} M_e(q_e) & -E \\ E^T & \mathbf{0}_2 \end{bmatrix}^{-1} \begin{bmatrix} M_e(q_e) \\ \mathbf{0}_{2\times 5} \end{bmatrix} \begin{bmatrix} I_3 & \mathbf{0}_2 \\ \mathbf{0}_{2\times 3} & \mathbf{0}_2 \end{bmatrix} \dot{q}_-. \tag{3.34}$$

3.2.2 Conservation of Angular Momentum

It is possible to derive a simpler model than the one from the previous section using the same assumptions, namely that the angular momentum about the point of impact should be conserved through the impact, as well as the angular momentum of the torso and stance leg about the hip. This stems from the assumption that the only forces acting on the biped through the duration of the impact, apart from the constraint forces, are the normal and tangent impact force at the point of impact. Thus, the angular momentum should be conserved about the point of impact as the impact forces generate no torque about this point. The same argument also holds true for the angular momentum of the torso and the swing leg about the hip as the torque from the actuator is assumed to be non-impulsive.

The angluar momentum L_i of a rigid body i about a point j in the inertial frame frame is given by [9]

$$L_i = \vec{r}_{ji} \times m_i \vec{v}_i + J_i \vec{\omega}_i, \tag{3.35}$$

where $\vec{r}_{ji}$ is the distance from the point j to i and $\vec{v}_i$ is the velocity of the body i, both measured in the inertial frame; J_i is the moment of inertia of the body and $\vec{\omega}_i$ is its angular velocity.

To find the angular momentum of the biped about the point o_4 given in Figure 3.3 (i.e., the point of impact), the vectors from this point to the COMs of the links of the biped must be found. Herein, absolute angles of the links of the robot will be used to clarify the procedure which are found from relative angles using the coordinate transformation from (3.1). The vectors from o_4 to the COM of each link measured in the inertial frame are thus

given as follows:

$$\begin{aligned}\vec{r}_{41} &= \begin{bmatrix} -L_3\sin(q_3^a) + (L_1 - l_1)\sin(q_1^a) \\ L_3\cos(q_3^a) - (L_1 - l_1)\cos(q_1^a) \\ 0 \end{bmatrix}, \\ \vec{r}_{42} &= \begin{bmatrix} -L_3\sin(q_3^a) - l_2\sin(q_2^a) \\ L_3\cos(q_3^a) + l_2\cos(q_2^a) \\ 0 \end{bmatrix}, \quad \vec{r}_{43} = \begin{bmatrix} -(L_3 - l_3)\sin(q_3^a) \\ (L_3 - l_3)\cos(q_3^a) \\ 0 \end{bmatrix}.\end{aligned} \tag{3.36}$$

The velocities of the links prior to and after the impact, i.e., $\vec{v}_{i-}$ and $\vec{v}_{i+}$, respectively, depend on the point of pivot but not on the point of reference. As the pivot point before the impact is given by the tip of the stance leg, o_o, the velocities $\vec{v}_{i-}$ are equal to those of (3.8) by adding the third dimension. Thus, using absolute angles, they are

$$\begin{aligned}\vec{v}_{1-} &= \begin{bmatrix} -l_1\dot{q}_{1-}^a\cos q_1^a \\ -l_1\dot{q}_{1-}^a\sin q_1^a \\ 0 \end{bmatrix}, \quad \vec{v}_{2-} = \begin{bmatrix} -L_1\dot{q}_{1-}^a\cos q_1^a - l_2\dot{q}_{2-}^a\cos q_2^a \\ -L_1\dot{q}_{1-}^a\sin q_1^a - l_2\dot{q}_{2-}^a\sin q_2^a \\ 0 \end{bmatrix}, \\ \vec{v}_{3-} &= \begin{bmatrix} -L_1\dot{q}_{1-}^a\cos q_1^a + l_3\dot{q}_{3-}^a\cos q_3^a \\ -L_1\dot{q}_{1-}^a\sin q_1^a + l_3\dot{q}_{3-}^a\sin q_3^a \\ 0 \end{bmatrix}.\end{aligned} \tag{3.37}$$

By the assumption that the swing leg should not rebound, but become the new stance leg, it will act as the pivot after the impact, meaning that the velocities $\vec{v}_{i+}$ must be found from the point o_4. This yields

$$\begin{aligned}\vec{v}_{1-} &= \begin{bmatrix} -L_3\dot{q}_{3+}^a\cos q_3^a + (L1 - l_1)\dot{q}_{1+}^a\cos q_1^a \\ -L_3\dot{q}_{3+}^a\sin q_3^a + (L1 - l_1)\dot{q}_{1+}^a\sin q_1^a \\ 0 \end{bmatrix}, \\ \vec{v}_{2+} &= \begin{bmatrix} -L_3\dot{q}_{3+}^a\cos q_3^a - l_2\dot{q}_{2+}^a\cos q_2^a \\ -L_3\dot{q}_{3+}^a\sin q_3^a - l_2\dot{q}_{2+}^a\sin q_2^a \\ 0 \end{bmatrix}, \quad \vec{v}_{3+} = \begin{bmatrix} -(L_3 - l_3)\dot{q}_{3+}^a\cos q_3^a \\ -(L_3 - l_3)\dot{q}_{3+}^a\sin q_3^a \\ 0 \end{bmatrix}.\end{aligned} \tag{3.38}$$

Since absolute angles are used, the angular velocity of a link i is simply the the vector $\vec{\omega}_{i\pm} = [0; 0; \dot{q}_{i\pm}^a]$. Therefore, the total angular momentum, L_b, of the biped is given by

$$L_b = \sum_{i=1}^{3} L_i = \sum_{i=1}^{3} \left(\vec{r}_{4i} \times m_i\vec{v}_{i-} + J_i\vec{\omega}_{i-}\right) = \sum_{i=1}^{3} \left(\vec{r}_{4i} \times m_i\vec{v}_{i+} + J_i\vec{\omega}_{i+}\right). \tag{3.39}$$

In order to find the angular momentum of the stance leg, L_{sl}, and the torso, L_t, about the hip, the vectors $\vec{r}_{H1}$ and $\vec{r}_{H2}$ must be defined. These are given by

$$\vec{r}_{H1} = \begin{bmatrix} (L_1 - l_1)\sin q_1^a \\ -(L_1 - l_1)\cos q_1^a \\ 0 \end{bmatrix}, \quad \vec{r}_{H2} = \begin{bmatrix} -l_2\sin q_2^a \\ l_2\cos q_2^a \\ 0 \end{bmatrix}. \tag{3.40}$$

The angular momenta are thus given by

$$L_{sl} = \vec{r}_{H1} \times m_1 \vec{v}_{1-} + J_1 \vec{\omega}_{1-} = \vec{r}_{H1} \times m_1 \vec{v}_{1+} + J_1 \vec{\omega}_{1+} \tag{3.41}$$

and

$$L_t = \vec{r}_{H2} \times m_2 \vec{v}_{2-} + J_2 \vec{\omega}_{2-} = \vec{r}_{H2} \times m_2 \vec{v}_{2+} + J_2 \vec{\omega}_{2+}. \tag{3.42}$$

Since the angular velocities, $\dot{q}_-$, prior to the impact are known, the three unknown angular velocities, $\dot{q}_{i+}$ for $i = 1, 2, 3$, can be found from the three equations (3.39), (3.41) and (3.42) by differentiating them w.r.t. the angular velocities as they are linear in these. Let L_{k-} denote the angular momentum prior to the impact about frame j, i.e.,

$$L_{k-} = \vec{r}_{jk} \times m_k \vec{v}_{k-} + J_k \vec{\omega}_{k-},$$

and let L_{k+} be the angular momentum after impact, where $k \in \{b, sl, t\}$. Thus, the relation between $\dot{q}^a_-$ and $\dot{q}^a_+$ can be found by combining the three equation and differentiating $\hat{L}_{k-} = L_{k-} \cdot \hat{k}$ w.r.t. $\dot{q}^a_{i-}$ and $\hat{L}_{k+} = L_{k+} \cdot \hat{k}$ w.r.t. $\dot{q}^a_{i+}$, where $\hat{k}$ is the unit vector of the z-axis in the inertial frame, i.e., $\hat{k} = [0; 0; 1]$. This yields

$$Q_+ \dot{q}^a_+ = Q_- \dot{q}^a_-, \tag{3.43}$$

where

$$Q_+ = \begin{bmatrix} \frac{\partial \hat{L}_{sl+}}{\partial \dot{q}^a_{1+}} & \frac{\partial \hat{L}_{sl+}}{\partial \dot{q}^a_{2+}} & \frac{\partial \hat{L}_{sl+}}{\partial \dot{q}^a_{3+}} \\ \frac{\partial \hat{L}_{t+}}{\partial \dot{q}^a_{1+}} & \frac{\partial \hat{L}_{t+}}{\partial \dot{q}^a_{2+}} & \frac{\partial \hat{L}_{t+}}{\partial \dot{q}^a_{3+}} \\ \frac{\partial \hat{L}_{b+}}{\partial \dot{q}^a_{1+}} & \frac{\partial \hat{L}_{b+}}{\partial \dot{q}^a_{2+}} & \frac{\partial \hat{L}_{b+}}{\partial \dot{q}^a_{3+}} \end{bmatrix} \quad \text{and} \quad Q_- = \begin{bmatrix} \frac{\partial \hat{L}_{sl-}}{\partial \dot{q}^a_{1-}} & \frac{\partial \hat{L}_{sl-}}{\partial \dot{q}^a_{2-}} & \frac{\partial \hat{L}_{sl-}}{\partial \dot{q}^a_{3-}} \\ \frac{\partial \hat{L}_{t-}}{\partial \dot{q}^a_{1-}} & \frac{\partial \hat{L}_{t-}}{\partial \dot{q}^a_{2-}} & \frac{\partial \hat{L}_{t-}}{\partial \dot{q}^a_{3-}} \\ \frac{\partial \hat{L}_{b-}}{\partial \dot{q}^a_{1-}} & \frac{\partial \hat{L}_{b-}}{\partial \dot{q}^a_{2-}} & \frac{\partial \hat{L}_{b-}}{\partial \dot{q}^a_{3-}} \end{bmatrix}. \tag{3.44}$$

By using the coordinate transformation K between absolute angular velocities and relative angular velocities given by differentiating (3.2) w.r.t. time, which is defined by

$$K = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}, \tag{3.45}$$

and the relabelling matrix from (3.24), one finds the angular velocities prior to the impact by

$$\dot{q}_+ = P_{\dot{q}}(q_-)\dot{q}_- = P_q K Q_+^{-1} Q_- K^{-1} \dot{q}_-. \tag{3.46}$$

Chapter 4

Procedure for finding Periodic Gaits

In this chapter, a procedure for finding gaits based numerical optimization will be presented, where gaits are found by inducing limit cycles inherent in the natural dynamics for a given open-loop control action. The main focus will be on finding gaits using the reduced dynamics found from the virtual holonomic constraint apporach presented in Section 2.4, but a basic search procedure on the full dynamics will also be given in order to compare with that on the reduced dynamics.

The search space will be restricted to *symmetric gaits*, meaning that the configuration of the robot will equal at the beginning and the end of step step, with the exception that the swing leg becomes the stance leg, and vice versa. Mathematically, this can be written as the following relations using absolute angles:

$$q_1^a(t_0) = -q_1^a(t_0 + T), \tag{4.1a}$$
$$q_1^a(t_0) = q_3^a(t_0 + T), \tag{4.1b}$$
$$q_3^a(t_0) = q_1^a(t_0 + T), \tag{4.1c}$$
$$q_2^a(t_0) = q_2^a(t_0 + T) = \varphi, \tag{4.1d}$$

where t_0 is denotes the time at the beginning of the step and T denotes the step duration, or step period, which, from now on, will be denoted $t_+ = t_0$ and $t_- = t_0 + T$, i.e., the time immediately after and prior to impact of the swing leg. Herein, the angle φ will be referred to as the *lean angle* of the torso, such that the torso is leaning forward if $\varphi < 0$. The problem will be further restricted to a constant step length, denoted L_{step}. From this, the initial and final angle of the stance leg, denoted $q_1(t_+)$ and $q_1(t_-)$, respectively, can be computed in relative angles using the following relations

$$q_1(t_-) = \arccos\Big(\frac{L_{step}}{2L_1}\Big), \tag{4.2a}$$
$$q_1(t_+) = \pi - q_1(t_-). \tag{4.2b}$$

Thus, for a symmetric gait, the initial and final configuration of the biped over a step can be directly computed knowing the step length, L_{step}, and the lean angle of the torso, φ, using the coordinate transformation of (3.2).

The chapter is outlined as follows. In Section 4.1 presents a basic search procedure for finding gaits on the full dynamical model. Section 4.2 introduces a search procedure for finding gaits from the reduced dynamics arising from enforcing virtual holonomic constraints on the system. The results from the search procedure are given in Section 4.3, with a brief discussion on the method and found results given in Section 4.4.

4.1 Finding Feasible Gaits on the Full Model

In this section, a basic search procedure for finding gaits on the full dynamical model will be presented for comparison with the method on the reduced dynamics presented later. Moreover, the procedure can give some indication as to see if one can expect to find gaits for a certain set of initial conditions, given by the step length and lean angle of the torso, due to constraints on the value and change of torque supplied by the actuators. For this, one must have the dynamics of the system, described by (2.9), as a constraint in the optimization problem, meaning it will be a highly nonlinear and non-convex optimization problem.

In order to constrain the optimization problem to follow the dynamics, one must use some integration method, e.g., Euler's method [9] for t from 0 to T with N discrete steps, giving the constraints

$$q_k - q_{k+1} + dt\dot{q}_{k+1} = 0, \tag{4.3a}$$

$$M(q_k)\Big(\dot{q}_{k+1} - \dot{q}_k\Big) + dt\Big(C(q_k, \dot{q}_k)\dot{q}_k + G(q_k) - B(q_k)u_k\Big) = 0, \tag{4.3b}$$

for $k = 0, \ldots, N-1$ and where $dt = T/N$ defines the time-step. Moreover, one must add the impact interpolation and velocity constraints:

$$q_{(k=1)} = P_q q_{(k=N)} + c \tag{4.4a}$$

$$\dot{q}_{(k=1)} = P_{\dot{q}}(q_-)\dot{q}_{(k=N)} \tag{4.4b}$$

where P_q and c are given in (3.24) and $P_{\dot{q}}$ is found from (3.46) or (3.33). It also possible to constrain the maximum actuator outputs and their maximum allowed rate:

$$u_{i,min} \leq u_k \leq u_{i,max}, \quad i \in \{1, 2\}, \tag{4.5a}$$

$$-dt \cdot \dot{u}_{i,max} \leq u_{k+1} - u_{i,k} \leq dt \cdot \dot{u}_{i,max}. \tag{4.5b}$$

Due to the relabelling of the legs at impact, the actuators u_1 and u_2 must in turn be relabelled, and since discontinuous jumps in the required torques must be avoided, the following constraints are needed in order to ensure a smooth torque profile over the impact[1]:

[1] Such constraints are dependent on the mounting of the actuators on the mechanical system. Indeed, if the actuators are mounted between the legs and the torso, then the constraints should be $u_{i,1} = -u_{j,N}$. However, (4.6) is here used as its enforcement can be easily seen in the found results.

$$u_{1,1} = u_{2,N}, \tag{4.6a}$$
$$u_{2,1} = u_{1,N}. \tag{4.6b}$$

Furthermore, using the normal and tangential forces, respectively denoted F^n and F^t, found from (3.34), one can add the assumption from the no rebound condition

$$F^N \geq 0, \tag{4.7}$$

as well as the no slip condition

$$F^t/F^n \leq \mu \tag{4.8}$$

as constraints, where μ denotes the friction coefficient between the swing leg and the walking surface. However, this is omitted in this thesis as it is assumed that μ is sufficiently large as to prevent slipping. The variables of the optimization problem are therefore q_k, $\dot{q}_k$ and u_k, for $k = 1, \ldots, N$, and the period T, giving a total of $8N + 1$ optimization variables.

The choice of objective function can be many, but a good choice is to minimize the energetic cost of transport (COT). This can be written as

$$W_{COT} = \frac{1}{g m_T L_{step}} \int_0^T \Big(|u_1(t)\dot{q}_2(t)| + |u_2(t)\dot{q}_3(t)| \Big) dt \tag{4.9}$$

where m_T denotes the total mass of the robot. Another performance index that can be used is the minimization of the quadratic cost over the duration of the step:

$$W_{CC} = \int_0^T \Big(q^T Q_q q + \dot{q}^T Q_{\dot{q}} \dot{q} + u^T R u \Big) dt, \tag{4.10}$$

where Q_q and $Q_{\dot{q}}$ are symmetric, positive semi-definite 3×3 matrices weighting the states and $R = R^T$ is a 2×2 positive definite matrix weighting the actuator inputs.

If the main concern when optimizing is to find a trajectory with the lowest maximum torque required, one can simply minimize the maximum value needed by any of the actuators over the duration of the step, which is equivalent to minimizing the cost function

$$W_{LT} = \max U \tag{4.11}$$

where $U = \{|u_1(t)|\} \cup \{|u_2(t)|\}\ \forall t \in [0, T]\}$.

4.2 Finding Feasible Gaits from the Reduced Dynamics

In this section, a method of searching for gaits using the reduced dynamics found from the virtual holonomic constraints approach will be presented. The method presented can be used for any choice of motion generator θ as long as it is strictly monotonic over the step, which is required by [55, Proposition 1] due to uniqueness of the solution. However, the main emphasis will be on $q_1 = \theta$, i.e., the angle of the stance leg, as it is likely to be monotonically decreasing over any "normal" gait, and since it is the most common choice in the literature [12, 26, 16].

4.2.1 Finding the Reduced Dynamics

Given the vector of the synchronization functions from (2.29) as

$$\begin{aligned}\Phi(\theta) &= [\phi_1(\theta); \phi_2(\theta); \phi_2(\theta)],\\ \Phi^{'}(\theta) &= [\phi_1^{'}(\theta); \phi_2^{'}(\theta); \phi_3^{'}(\theta)],\\ \Phi^{''}(\theta) &= [\phi_1^{''}(\theta); \phi_2^{''}(\theta); \phi_3^{''}(\theta)],\end{aligned} \tag{4.12}$$

such that $q = \Phi$, $\dot{q} = \Phi^{'}(\theta)\dot{\theta}$ and $\ddot{q} = \Phi^{'}(\theta)\ddot{\theta} + \Phi^{''}(\theta)\dot{\theta}^2$, and noting that

$$B^{\perp} = [1, 0, 0] \tag{4.13}$$

is an annihilator of the coupling matrix B, given in (3.19), the procedure of (2.31) reduces the dynamical system to the reduced dynamics equation

$$\alpha(\theta)\ddot{\theta} + \beta(\theta)\dot{\theta}^2 + \gamma(\theta) = 0. \tag{4.14}$$

Herein, the functions $\alpha(\theta)$, $\beta(\theta)$ and $\gamma(\theta)$ are found using (2.33), and are given as

$$\begin{aligned}\alpha(\theta) =&(L_1\cos(\phi_2(\theta))l_2m_2 + L_1\cos(\phi_2(\theta)+\phi_3(\theta))l_3m_3 + l_2^2m_2 + l_3^2m_3 \\ &+ J2 + J3)\phi_2^{'} + (L_1\cos(\phi_2(\theta)+\phi_3(\theta))l_3m_3 + l_3^2m_3 + J_3)\phi_3^{'}\\ &+ 2L_1\cos(\phi_2(\theta))\phi_1^{'}l_2m_2 + 2L_1\cos(\phi_2(\theta)+\phi_3(\theta))\phi_1^{'}l_3m_3 + L_1^2\phi_1^{'}m_2\\ &+ L_1^2\phi_1^{'}m_3 + \phi_1^{'}l_1^2m_1 + \phi_1^{'}l_2^2m_2 + \phi_1^{'}l_3^2m_3 + J_1\phi_1^{'} + J_2\phi_1^{'} + J_3\phi_1^{'},\end{aligned} \tag{4.15a}$$

$$\begin{aligned}\beta(\theta) =&(L_1\cos(\phi_2(\theta))l_2m_2 + L_1\cos(\phi_2(\theta)+\phi_3(\theta))l_3m_3 + l_2^2m_2 + l_3^2m_3 \\ &+ J_2 + J_3)\phi_2^{''} + (L_1\cos(\phi_2(\theta)+\phi_3(\theta))l_3m_3 + l_3^2m_3 + J_3)\phi_3^{''}\\ &- 2L_1\sin(\phi_2(\theta))\phi_1^{'}\phi_2^{'}l_2m_2 - 2L_1\sin(\phi_2(\theta)+\phi_3(\theta))\phi_1^{'}\phi_2^{'}l_3m_3\\ &- 2L_1\sin(\phi_2(\theta)+\phi_3(\theta))\phi_1^{'}\phi_3^{'}l_3m_3 - 2L_1\sin(\phi_2(\theta)+\phi_3(\theta))\phi_2^{'}\phi_3^{'}l_3m_3\\ &- L_1\sin(\phi_2(\theta))(\phi_2^{'})^2l_2m_2 - L_1\sin(\phi_2(\theta)+\phi_3(\theta))(\phi_2^{'})^2l_3m_3\\ &- L_1\sin(\phi_2(\theta)+\phi_3(\theta))(\phi_3^{'})^2l_3m_3 + 2L_1\phi_1^{''}\cos(\phi_2(\theta))l_2m_2\\ &+ 2L_1\phi_1^{'}\cos(\phi_2(\theta)+\phi_3(\theta))l_3m_3 + L_1^2\phi_1^{''}m_2 + L_1^2\phi_1^{''}m_3 + \phi_1^{''}l1^2m_1\\ &+ \phi_1^{''}l_2^2m_2 + \phi_1^{''}l_3^2m_3 + J_1\phi_1^{''} + J_2\phi_1^{''} + J_3\phi_1^{''},\end{aligned} \tag{4.15b}$$

$$\begin{aligned}\gamma(\theta) =&\cos(\phi_2(\theta)+\phi_1(\theta))gl_2m_2 + \cos(\phi_2(\theta)+\phi_3(\theta)+\phi_1(\theta))gl_3m_3\\ &+ (L_1gm_2 + L_1gm_3 + gl_1m_1)\cos(\phi_1(\theta)).\end{aligned} \tag{4.15c}$$

Moreover, if one uses the the matrices $B_2 = [0; 1; 0]$ and $B_3 = [0; 0; 1]$ instead of the annihilator $B^{\perp}$, one gets two new equations which are also dependent on the control inputs:

$$\alpha_2(\theta)\ddot{\theta} + \beta_2(\theta)\dot{\theta}^2 + \gamma_2(\theta) = u_1, \tag{4.16a}$$

$$\alpha_3(\theta)\ddot{\theta} + \beta_3(\theta)\dot{\theta}^2 + \gamma_3(\theta) = u_2. \tag{4.16b}$$

The full expressions for the functions $\alpha_i(\theta)$, $\beta_i(\theta)$ and $\gamma_i(\theta)$, for $i = 1, 2$, are given in (B.2) and (B.3) in Appendix B.

If one now observes that the variables $\ddot{\theta}$ and $\dot{\theta}^2$ are not independent, but are given by the relation [47] (see also Appendix A.1)

$$\ddot{\theta} = \frac{1}{2}\frac{\partial}{\partial\theta}\big(\dot{\theta}^2\big), \tag{4.17}$$

one can define $Y := \dot{\theta}^2$, and thus (4.14) and the two equations of (4.16) can be written as three first order equations of the form:

$$\alpha(\theta)Y^{'}(\theta) + 2\beta(\theta)Y(\theta) + 2\gamma(\theta) = 0, \tag{4.18a}$$

$$\alpha_2(\theta)Y^{'}(\theta) + 2\beta_2(\theta)Y(\theta) + 2\gamma_2(\theta) = 2u_1, \tag{4.18b}$$

$$\alpha_3(\theta)Y^{'}(\theta) + 2\beta_3(\theta)Y(\theta) + 2\gamma_3(\theta) = 2u_2 \tag{4.18c}$$

which only depend on θ. This will be important for integrating the system internally in the optimization problem as the systems of (4.18) are time-independent and only of first order compared to (4.14).

Another interesting property of these equations can be found by looking closely at (4.15a) and (4.15b), where one can see that

$$\beta(\theta) = \frac{\partial}{\partial\theta}\big(\alpha(\theta)\big). \tag{4.19}$$

This property will be used later in order to simplify the integral function. Note, however, that his relation does not hold for the other two equations, i.e., $\beta_i(\theta) \neq \frac{\partial}{\partial\theta}\big(\alpha_i(\theta)\big)$ for $i = 2, 3$. A general case for when this property holds is stated in the next Theorem.

Theorem 4.2.1. *Given an Euler-Lagrange system of the form* (2.8) *with one degree of underactuation and n generalized coordinates $q \in Q \subseteq \mathbb{R}^n$. Let $B^{\perp} = [b_1^{\perp}, \ldots, b_n^{\perp}]$ be the annihilator of the coupling matrix $B(q)$ with constant, scalar elements $b_i^{\perp}$. Then, the relation*

$$\beta(\theta) = \frac{\partial}{\partial\theta}\big(\alpha(\theta)\big) \tag{4.20}$$

holds for the reduced dynamics (2.34) *if and only if*

$$\frac{1}{2}\Phi^{'}(\theta)^T\left[\sum_{i=1}^{n} b_i^{\perp}\frac{\partial M(q)}{\partial q_i}\Big|_{q=\Phi(\theta)}\right]\Phi^{'}(\theta) = 0 \quad \forall\theta \in \mathbb{R}. \tag{4.21}$$

Proof. The proof is given in Appendix A.4. □

Corollary 4.2.1.1. *For systems with more than one degrees of underactuation, Theorem 4.2.1 holds for each of the resulting equations* (2.32) *by using the rows of the $m \times n$ annihilator matrix.*

Proof. The proof of this is trivial and follows directly from the proof of Theorem 4.2.1. □

It is not difficult see that the requirements of Theorem 4.2.1 is fulfilled for the three link biped since

$$\Phi^{'}(\theta)^{T}\left[\frac{\partial M(q)}{\partial q_1}\Big|_{q=\Phi(\theta)}\right]\Phi^{'}(\theta)=0$$

as the inertia matrix is independent of the passive coordinate q_1, such that $\frac{\partial M(q)}{\partial q_1}=\mathbf{0}_{3\times 3}$. This in itself is not surprising as the kinetic energy of the system is not dependent on q_1 as it is an absolute angle measured w.r.t. the inertial frame, and thus, by [54, Proposition B.8], the inertia matrix is independent of it.

It is important to note that even though the requirements of Theorem 4.2.1 are not fulfilled, one can acquire similar forms to (4.20) by clever choice of virtual holonomic constraints. Indeed, in [46] a clever choice of constraints by the authors gave the even more powerful relation[2]

$$\frac{\partial\alpha(\theta)}{\partial\theta}=\frac{1}{2}\beta(\theta)$$

for the cart pendulum, where the requirements of Theorem 4.2.1 were impossible to fulfil. Moreover, for an n-degree system, one can define $\mathcal{A}(\theta):=\phi_1^{'}(\theta)\alpha_1(\theta)+\cdots+\phi_n^{'}(\theta)\alpha_n(\theta)$ and $\mathcal{B}(\theta):=\phi_1^{'}(\theta)\beta_1(\theta)+\cdots+\phi_n^{'}(\theta)\beta_n(\theta)$, for which the relations

$$\frac{\partial\mathcal{A}(\theta)}{\partial\theta}=\frac{1}{2}\mathcal{B}(\theta)$$

always holds. For a completely passive system, this property can be used to easily integrate and find another function which keeps its value throughout the motion. This property comes from the total energy of the system; see [2].

4.2.2 Choosing Synchronization Functions

The choice of virtual holonomic constraints are many, but a common choice when searching are different types of polynomials as they can take many shapes with relatively few coefficients. In this text, three types of polynomials will be tried and compared; these are: regular polynomials, Bezièr curves and trigonometric polynomials. Note that there are other good choices of virtual constraints, e.g., different types of splines, which are a set of polynomials themselves.

Let the motion generator be chosen as $\theta=q_1$. It then follows that the synchronization function for q_1 and its partial derivatives are given by

$$\phi_1(\theta)=\theta, \tag{4.22a}$$

$$\phi_1^{'}(\theta)=1, \tag{4.22b}$$

$$\phi_1^{''}(\theta)=0. \tag{4.22c}$$

$$\tag{4.22d}$$

For the remaining synchronization functions, the following will be used as the polynomial variable

[2]This relation means that one can write the reduced dynamics on the from $\frac{\partial}{\partial\theta}\left(\alpha(\theta)\dot{\theta}\right)+2\gamma(\theta)=0$ using the relation (4.17), which then easily can be integrated.

$$s := s(\theta) = \frac{\theta - \theta_+}{\theta_- - \theta_+}, \tag{4.23}$$

such that $s(\theta_+) = 0$, $s(\theta_-) = 1$ and $s \in [0, 1]$. Thus, the partial derivatives of the remaining synchronization functions can be computed as

$$\phi_i^{'}(\theta) = \frac{\partial \phi_i}{\partial \theta} = \frac{\partial \phi_i}{\partial s}\frac{\partial s}{\partial \theta},$$
$$\phi_i^{''}(\theta) = \frac{\partial^2 \phi_i}{\partial \theta^2} = \frac{\partial^2 \phi_i}{\partial s^2}\Big(\frac{\partial s}{\partial \theta}\Big)^2,$$

where $\partial s/\partial \theta = 1/(\theta_- - \theta_+)$.

Regular polynomials

Using Nth order regular polynomials, the synchronization functions are by

$$\phi_i(\theta) = \sum_{k=0}^{N} a_{i,k} s^k, \quad i \in \{2, 3\}, \tag{4.24}$$

where $a_{2,k}$ and $a_{3,k}$ are the *polynomial coefficients* of the VHCs. The partial derivatives are thus given by

$$\phi_i^{'}(\theta) = \frac{1}{\theta_- - \theta_+} \sum_{k=1}^{N} k a_{i,k} s^{k-1}, \tag{4.25a}$$

$$\phi_i^{''}(\theta) = \frac{1}{(\theta_- - \theta_+)^2} \sum_{k=2}^{N} k(k-1) a_{i,k} s^{k-2}. \tag{4.25b}$$

Using the property that $s(\theta_+) = 0$, one finds that $\phi_i(\theta_+) = a_{i,0}$. Thus, one can find $a_{2,0}$ and $a_{3,0}$ prior to the search by the formula

$$a_{2,0} = \frac{\pi}{2} + \varphi - \theta_+, \tag{4.26a}$$

$$a_{3,0} = \frac{3\pi}{2} - \varphi - \theta_+, \tag{4.26b}$$

such that the remaining $2N$ coefficients are left as optimization variables.

Bezièr curves

Another choice of virtual holonomic constraints are Bezièr curves, often called Bezièr polynomials, giving the following synchronization functions [10]:

$$\phi_i(\theta) = \sum_{k=0}^{N} a_{i,k} B_{N,k}(s) = \sum_{k=0}^{N} a_{i,k} \frac{N!}{k!(N-k)!} s^k (1-s)^{N-k}, \quad i \in \{2, 3\}, \tag{4.27}$$

where $B_{N,k}$ are Nth order Bernstein polynomials. The first and second derivatives of the Bezièr curves are given by the following [10]

$$\phi_i^{'}(\theta) = \frac{1}{\theta_- - \theta_+} \sum_{k=0}^{N-1} B_{N-1,k}(s) N\big(a_{i,k+1} - a_{i,k}\big), \tag{4.28a}$$

$$\phi_i^{''}(\theta) = \frac{1}{(\theta_- - \theta_+)^2} \sum_{k=0}^{N-2} B_{N-2,k}(s) N(N-1)\big(a_{i,k+2} - 2a_{i,k+1} + a_{i,k}\big), \tag{4.28b}$$

from which it follows that

$$\phi_i^{'}(\theta_+) = \frac{N}{\theta_- - \theta_+}\big(a_{i,1} - a_{i,0}\big), \tag{4.29a}$$

$$\phi_i^{'}(\theta_-) = \frac{N}{\theta_- - \theta_+}\big(a_{i,N} - a_{i,N-1}\big), \tag{4.29b}$$

$$\phi_i^{''}(\theta_+) = \frac{N(N-1)}{\big(\theta_- - \theta_+\big)^2}\big(a_{i,2} - 2a_{i,1} + a_{i,0}\big), \tag{4.29c}$$

$$\phi_i^{''}(\theta_-) = \frac{N(N-1)}{\big(\theta_- - \theta_+\big)^2}\big(a_{i,N} - 2a_{i,N-1} + a_{i,N-2}\big). \tag{4.29d}$$

One of the advantages of using Bezièr curves is that both the first and last of the polynomial coefficients, $a_{i,0}$ and $a_{i,N}$, of the synchronization can be precomputed if one knows the initial and final configuration of the biped, compared to regular polynomials where only the first could be pre-computed. The first coefficients are found by the same formula as the regular polynomials, i.e., from (4.26), while the last are found from

$$a_{2,N} = \frac{\pi}{2} + \varphi - \theta_-, \tag{4.30a}$$

$$a_{3,N} = \frac{3\pi}{2} - \varphi - \theta_-, \tag{4.30b}$$

meaning that the remaining $2N-2$ coefficients are left as optimization variables.

Another useful property is that the resulting curve from the Bezièr polynomial lies within the convex hull of the control points [10], which in this case are the coefficient a_i, k for $k = 0, \ldots, N$. This makes Bezièr curves ideal for numerical optimization as small changes the in the coefficients does not result in large changes in the curve. Moreover, it enables intuitively choosing values of the coefficients as the initial condition for the search.

Trigonometric polynomials

Another choice for the synchronization functions are trigonometric polynomials, which are truncated Fourier series which are well suited to approximate periodic functions [24]. When using trigonometric polynomials, the synchronization functions are given by

$$\phi_i(\theta) = a_{i,0} + \sum_{k=1}^{N} \Big(a_{i,k}\cos(k\omega_i s) + b_{i,k}\sin(k\omega_i s)\Big), \quad i \in \{1,2\}, \tag{4.31}$$

such that the partial derivatives are

$$\phi_i^{'}(\theta) = \frac{\omega_i}{\theta_- - \theta_+} \sum_{k=1}^{N} k\Big(- a_{i,k}\sin(k\omega_i s) + b_{i,k}\cos(\omega_i s)\Big), \tag{4.32a}$$

$$\phi_i^{''}(\theta) = \frac{\omega_i^2}{(\theta_- - \theta_+)^2} \sum_{k=1}^{N} k^2\Big(- a_{i,k}\cos(k\omega_i s) - b_{i,k}\sin(k\omega_i s)\Big). \tag{4.32b}$$

The parameters ω_2 and ω_3 are referred to as the *harmonics* of the trigonometric polynomials. However, since the synchronization function are not truly periodic such that the harmonics can be directly computed, they must either be set to a constant value of choice or added as optimization variables. However, having these as optimization variables will make the resulting problem even more non-linear such that it is likely to be advantageous to tune these by hand to get a satisfactory performance rather than having them as variables, even if they are restricted to a small range.

As none of the parameters can be chosen directly prior to the search, the number of optimization variables are $2(2N + 1)$ with the addition of two more if the harmonics are left as optimization variables.

4.2.3 Constraints

Interpolation Constraints

In order to simplify the problem, as well as give some degree choice, one can fix certain variables of the problem. As previously mentioned, in this text, the step length and the lean angle of the torso will be be fixed immediately prior to and after the impact, as it was when optimizing over the full model. Thus, since one wants q_2^a to be equal to the angle φ at impact, one must add the constraint

$$\phi_1(\theta) + \phi_2(\theta) - \pi/2 = \varphi, \quad \theta \in \{\theta_+, \theta_-\}. \tag{4.33}$$

For the step length, the relations of (4.2) gives the constraints on the angles of the swing leg prior to and after impact as

$$\phi_1(\theta_-) = \arccos\Big(\frac{L_{step}}{2L_1}\Big), \tag{4.34a}$$

$$\phi_1(\theta_+) = \pi - \phi_1(\theta_-), \tag{4.34b}$$

where $\phi_1(\theta_+) = \pi - \phi_1(\theta_-)$ comes from the assumption of a symmetric gait, giving $\phi_1(\theta_+) > \phi_1(\theta_-)$. Furthermore, due to symmetry of the stance and swing leg before and after impact, one must add the constraint

$$2\phi_1(\theta) + \phi_2(\theta) + \phi_3(\theta) - 2\pi = 0, \quad \theta \in \{\theta_+, \theta_-\}. \tag{4.35}$$

Lastly, the impact mapping interpolation due to the symmetric configuration gives the constraints

$$\phi_1(\theta_+) = \phi_1(\theta_-) + \phi_2(\theta_-) + \phi_3(\theta_-) - \pi, \tag{4.36}$$

$$\phi_2(\theta_+) = -\phi_3(\theta_-) + \pi, \tag{4.37}$$

$$\phi_3(\theta_+) = -\phi_2(\theta_-) + \pi. \tag{4.38}$$

If the choice of motion generator is $\theta = q_1$, then θ_+ and θ_- are found directly from (4.34), and $\phi_1(\theta)$ is thus known, which simplifies the problem.

Impact Velocity constraints

In order to satisfy the the change of velocities at impact, given by the relation

$$\dot{q}_+ = P_{\dot{q}}(q_-)\dot{q}_-, \tag{4.39}$$

one now must take into account that the velocities are now governed by the motion generator by the relation $\dot{q} = \Phi^{'}(\theta)\dot{\theta}$. Plugging this into (4.39) yields

$$\Phi^{'}(\theta_+)\dot{\theta}_+ = P_{\dot{q}}(\Phi(\theta_-))\Phi^{'}(\theta_-)\dot{\theta}_-. \tag{4.40}$$

In order to simplify this constraint, one can define the column vector

$$\Xi(\theta_-) := P_{\dot{q}}(\Phi(\theta_-))\Phi^{'}(\theta_-) = [\xi_1(\theta_-); \xi_2(\theta_-); \xi_3(\theta_-)],$$

such that the following should hold

$$\Phi^{'}(\theta_+)\dot{\theta}_+ = \Xi(\theta_-)\dot{\theta}_-.$$

This means that for all $i \in \{1, 2, 3\}$, the relation

$$\phi_i^{'}(\theta_+)/\xi_i(\theta_-) = \dot{\theta}_-/\dot{\theta}_+ = \kappa \tag{4.41}$$

should hold for some nonzero κ, which further implies

$$\phi_i^{'}(\theta_+)/\xi_i(\theta_-) = \phi_j^{'}(\theta_+)/\xi_j(\theta_-) = \kappa, \quad \forall i, j \in \{1, 2, 3\},$$

assuming $\xi_i(\theta_-) \neq 0$. Thus, the following three bilinear constraints must be added to the problem:

$$\phi_1^{'}(\theta_+) = \kappa\xi_1(\theta_-), \tag{4.42a}$$

$$\phi_2^{'}(\theta_+) = \kappa\xi_2(\theta_-), \tag{4.42b}$$

$$\phi_3^{'}(\theta_+) = \kappa\xi_3(\theta_-), \tag{4.42c}$$

where κ is an optimization variable. These relations means that the angles of the velocities prior to and after impact are related by the impact map. Furthermore, the velocity amplitudes can be any as long as their proportion equals κ, i.e., $\dot{\theta}_- = \kappa\dot{\theta}_+$.

One can now further deduce that if the motion generator is given by $\theta = q_1$, it is clear that θ should be monotonically decreasing since $\theta_+ > \theta_-$. Thus, the following must be true:

$$\dot{\theta}_+ < 0 \quad \text{and} \quad \dot{\theta}_- < 0,$$

giving the additional contraint

$$\kappa > 0. \tag{4.43}$$

Note that the constraint $\kappa > 0$ holds irrespectively of the choice of θ as it must be either monotonically increasing or decreasing, such that $\kappa = \dot{\theta}_-/\dot{\theta}_+ > 0$ for any choice of θ.

Impact direction constraint

At impact, the swing leg should be have a negative vertical velocity, thus

$$L_1\dot{\theta}_-\phi_1^{'}(\theta_-)\cos(\phi_1(\theta_-)) \\ + L_3\dot{\theta}_-\big(\phi_1^{'}(\theta_-)+\phi_2^{'}(\theta_-)+\phi_3^{'}(\theta_-)\big)\cos(\phi_1(\theta_-)+\phi_2(\theta_-)+\phi_3(\theta_-)) < 0.$$

If one assumes that $L_1 = L_3$, i.e., both legs have the same length, which is reasonable, and that the gait is symmetric such that

$$\cos(\phi_1(\theta_-)) = \cos(\phi_1(\theta_-)+\phi_2(\theta_-)+\phi_3(\theta_-)),$$

then one can write the constraint as

$$\dot{\theta}_-\cos(\phi_1(\theta_-))\big(2\phi_1^{'}(\theta_-)+\phi_2^{'}(\theta_-)+\phi_3^{'}(\theta_-)\big) < 0.$$

If the motion generator is chosen as $\theta = q_1$, then it is clear that $\dot{\theta} < 0$ on the interval and that $\cos(\phi_1(\theta_-)) > 0$, so the constraint becomes

$$2\phi_1^{'}(\theta_-)+\phi_2^{'}(\theta_-)+\phi_3^{'}(\theta_-) > 0. \tag{4.44}$$

Constraints on $\dot{\theta}_+$ and $\dot{\theta}_-$

Suppose the synchronization functions of (4.12) are given. In order to check if a solution exists for these functions, it is known from Theorem 2.4.1 that

$$I(\theta_+,\dot{\theta}_+,\theta_-,\dot{\theta}_-) = \dot{\theta}_-^2 - \exp\left\{-\int_{\theta_+}^{\theta_-}\frac{2\beta(\tau)}{\alpha(\tau)}d\tau\right\} \\ \times\left[\dot{\theta}_+^2 - \int_{\theta_+}^{\theta_-}\exp\left\{\int_{\theta_+}^{s}\frac{2\beta(\tau)}{\alpha(\tau)}d\tau\right\}\frac{2\gamma(s)}{\alpha(s)}ds\right] \tag{4.45}$$

must be equal to zero for some pair $\dot{\theta}_+, \dot{\theta}_-$ for there to be a solution. Thus, one must find $\dot{\theta}_+$ and $\dot{\theta}_-$ such that

$$I(\theta_+,\dot{\theta}_+,\theta_-,\dot{\theta}_-) = 0, \tag{4.46}$$

with the known constraints on $\dot{\theta}_-$ and $\dot{\theta}_+$ from the previous section upheld, namely $\kappa > 0$, as well as $\dot{\theta}_+ < 0$ and $\dot{\theta}_- < 0$ if q_1 is chosen as the motion generator.

By defining

$$\Psi_1 := \Psi_1(\theta_+,\theta_-) = \exp\left\{-\int_{\theta_+}^{\theta_-}\frac{2\beta(\tau)}{\alpha(\tau)}d\tau\right\}$$

and

$$\Psi_2 := \Psi_2(\theta_+,\theta_-) = \int_{\theta_+}^{\theta_-}\exp\left\{\int_{\theta_+}^{s}\frac{2\beta(\tau)}{\alpha(\tau)}d\tau\right\}\frac{2\gamma(s)}{\alpha(s)}ds,$$

which can be computed only knowing θ_+ and θ_-, the constraint (4.46) can be written as

$$\dot{\theta}_-^2 - \Psi_1\Big(\dot{\theta}_+^2 - \Psi_2\Big) = 0. \tag{4.47}$$

There are several integration methods which can be used for computing Ψ_1 and Ψ_2. A method based on the trapezoidal rule is given in Appendix C.1. However, using the trapezoidal rule can lead to a rather large error if the number of integration steps is not chosen high enough due to the nested integrals in Ψ_2. Luckily, the relation noted in (4.20), i.e.,

$$\beta(\theta) = \frac{\partial}{\partial\theta}\big(\alpha(\theta)\big),$$

means that the integral functions can be written on a form without the need to computing nested integrals. This new form is stated in the next proposition.

Proposition 4.2.2. *If the relation* (4.20) *holds for equation* (4.14)*, then the integral function* (4.45) *can be written on the form*

$$I(\theta_+, \dot\theta_+, \theta_-, \dot\theta_-) = \dot\theta_-^2 - \hat\Psi_1\Big(\dot\theta_+^2 - \hat\Psi_2\Big), \tag{4.48}$$

where the functions $\hat\Psi_1$ and $\hat\Psi_2$ are given as

$$\hat\Psi_1 := \hat\Psi_1(\theta_+, \theta_-) = \left(\frac{\alpha(\theta_+)}{\alpha(\theta_-)}\right)^2 \tag{4.49}$$

and

$$\hat\Psi_2 := \hat\Psi_2(\theta_+, \theta_-) = \frac{2}{\big(\alpha(\theta_+)\big)^2}\int_{\theta_+}^{\theta_-} \alpha(\tau)\gamma(\tau)d\tau. \tag{4.50}$$

Proof. The proof is given in Appendix A.5. □

Remark (First integral)**.** *It is interesting to remark that according to [32], by defining*

$$\tilde M(\theta) = \left(\frac{\alpha(\theta)}{\alpha(0)}\right)^2,$$

which they refer to as the virtual mass*, and*

$$\tilde V(\theta) = \frac{1}{\big(\alpha(0)\big)^2}\int_0^{\theta} \alpha(\tau)\gamma(\tau)d\tau,$$

which they refer to as the virtual potential*; the Euler-Lagrange equation* (2.8) *with the Lagrangian $\tilde{\mathcal{L}}(\theta, \dot\theta) = \frac{1}{2}\tilde M(\theta)\dot\theta^2 - \tilde V(\theta)$, leads to the reduced dynamics* (4.14)*. Furthermore, the total energy $E_0(\theta, \dot\theta) = \frac{1}{2}\tilde M(\theta)\dot\theta^2 + \tilde V(\theta)$ of* (4.14) *is a first integral of the system, meaning that is constant along all solutions of* (4.14)*. Indeed, it is not hard to see that by taking $E_0(\theta_-, \dot\theta_-) - E_0(\theta_+, \dot\theta_+)$, and assuming it to be equal to zero, one obtains* (4.48)*.*

There is, however, a disadvantage of using (4.48) for optimization as it becomes infinite when $\alpha(\theta_-) = 0$. This can be remedied by multiplying (4.48) by $\big(\alpha(\theta_-)\big)^2$, yielding a new integral function which is zero along any solution:

$$\hat I(\theta_-, \dot\theta_-, \theta_+, \dot\theta_+) = \big(\alpha(\theta_-)\dot\theta_-\big)^2 - \big(\alpha(\theta_+)\dot\theta_+\big)^2 + 2\int_{\theta_+}^{\theta_-} \alpha(\tau)\gamma(\tau)d\tau. \tag{4.51}$$

By noting that this can also be zero for certain functions $\alpha(\theta)$, e.g., $\alpha(\theta) = 0 \; \forall \theta \in [\theta_+, \theta_-]$, one must add the additional constraint:

$$\alpha(\theta) \neq 0, \quad \forall \theta \in [\theta_+, \theta_-]. \tag{4.52}$$

This constraint is equivalent to saying that $\alpha(\theta)$ must either always positive or always negative on the interval, i.e., if $\alpha(\theta_-) > 0$, then $\alpha(\theta) > 0$ for all $\theta \in (\theta_+, \theta_-]$, and vice versa. If the motion generator is chosen as $\theta = q_1$, one can deduce from (4.15a) and the constraint (4.47) that $\alpha(\theta_+) > 0$, thus the constraint

$$\alpha(\theta) > 0, \quad \forall \theta \in [\theta_+, \theta_-], \tag{4.53}$$

must be used. Note, however, that the constraint (4.52) must hold regardless of the choice of the integral function as to avoid asymptotes (infinite acceleration) in the phase-space of (2.34).

Since one now only needs to evaluate one integral, i.e., the one in $\hat{\Psi}_2$, in order to compute the value of the integral function (4.48), one can now easily implement Simpson's rule of integration [24] which will likely be far more accurate than when computing the integral function with nested integrals(c.f., equation (4.45)) using the trapezoidal rule as in Appendix C.1. A proposed implementation for computing (4.51) using Simpson's rule is given in Algorithm 1.

Algorithm 1 Approximation of the integral function using Simpson's rule of integration.

Set $d\theta = (\theta_- - \theta_+)/n$ for some sufficiently large, even integer n
$s_0 = (\alpha(\theta_+)\gamma(\theta_+))$
$s_1 = 0$
$s_2 = 0$
$\theta = \theta_+$
for i from 1 to $n - 3$ **do**
    $\theta = \theta + d\theta$
    $s_1 = s_1 + \Gamma + \alpha(\theta)\gamma(\theta)$
    $\theta = \theta + d\theta$
    $s_2 = s_2 + \alpha(\theta)\gamma(\theta)$
    $i = i + 2$
end for
$\theta = \theta + d\theta$
$s_1 = s_1 + \alpha(\theta)\gamma(\theta)$
$\Gamma = d\theta \frac{2}{3}(s_0 + 4s_1 + 2s_2)$
Return $I(\theta_+, \dot{\theta}_+, \theta_-, \dot{\theta}_-) = \left(\alpha(\theta_-)\dot{\theta}_-\right)^2 - \left(\alpha(\theta_+)\dot{\theta}_+\right)^2 + \Gamma$

Continuity condition for the actuators over the impact

From the condition (4.6), i.e., $u_1(0) = u_2(T)$ and $u_2(0) = u_1(T)$, one must find a way to write this as a constraint for the reduced dynamics. This is not difficult knowing the final and end configuration, as well as the initial velocity of the motion generator and the

κ parameter. Since the acceleration of the motion generator is given by $\ddot{\theta} = -\big(\beta(\theta)\dot{\theta}^2 + \gamma(\theta)\big)/\alpha(\theta)$, one can simply use the relations (4.16), giving the following constraints

$$\alpha_2(\theta_+)\ddot{\theta}_+ + \beta_2(\theta_+)\dot{\theta}_+^2 + \gamma_2(\theta_+) = \alpha_3(\theta_-)\ddot{\theta}_- + \beta_3(\theta_-)\dot{\theta}_-^2 + \gamma_3(\theta_-), \tag{4.54a}$$

$$\alpha_3(\theta_+)\ddot{\theta}_+ + \beta_3(\theta_+)\dot{\theta}_+^2 + \gamma_3(\theta_+) = \alpha_2(\theta_-)\ddot{\theta}_- + \beta_2(\theta_-)\dot{\theta}_-^2 + \gamma_2(\theta_-), \tag{4.54b}$$

where $\dot{\theta}_-$ can be found using the relation $\dot{\theta}_- = \kappa\theta_+$.

4.2.4 Summary of the optimization problem

In this section, a summary of the optimization problem for the reduced dynamics is presented. Compared to the optimization of the full model as presented in Section 4.1, where the number of optimization variables was the total number of states and actuator inputs times the number of integration steps; the number of optimization variables for the reduced dynamics are restricted to the number of parameters in the synchronization functions in addition to the initial velocity $\dot{\theta}_+$ and the proportion parameter $\kappa = \dot{\theta}_-/\dot{\theta}_+$. Thus, the number of optimization variables using Nth order polynomials with q_1 as the MG are $2N+2$ using regular polynomials, $2N$ using Bezièr curves and $4(N+1)$ for trigonometric polynomials (if the harmonics ω_2 and ω_3 are set to constant values).

It should be noted that the initial velocity $\dot{\theta}_+$ and the parameter κ can be found directly from (4.41) and (4.45) just knowing then parameters of the synchronization variables such that they can be omitted as optimization variables. However, having these as optimization variables, and restricting them using linear inequality constraints, lets one easily restrict the search space to an area of interest, such that they act as an additional heuristic.

The objective function of the problem can be chosen as either the minimization of (4.9), (4.10) or (4.11). Thus, for each iteration of the solver, the reduced dynamics (4.14) must be integrated from θ_+ to θ_- and then evaluating the torque requirements using (4.16). Since the period T of the step is unknown, this motivates integrating over (4.18a) instead, which, in addition to only being a first order equation compared to (4.14) which is of second order, is also time-independent and can be simply integrated over $[\theta_+, \theta_-]$ using an integration method of choice. Thus, one finds that its initial value, Y_+, is simply found from $Y_+ = \dot{\theta}_+^2$, and that the additional constraint

$$Y(\theta) \geq 0, \quad \forall\theta \in [\theta_+, \theta_-] \tag{4.55}$$

must be added as $Y(\theta) := \dot{\theta}^2$. If the motion generator is chosen as $\theta = q_1$ such that $\dot{\theta} < 0$ for all $\theta \in [\theta_+, \theta_-]$, one can determine $\dot{\theta}$ from $Y(\theta)$ by the relation $\dot{\theta} = -\sqrt{Y(\theta)}$.

Since one has eliminated the time-dependence, the objective function (4.9) can not be used directly. However, by using the relation $\dot{q}_i = \phi_i'(\theta)\dot{\theta}$, it can be rewritten on the form

$$W_{COT} = \frac{1}{g m_T L_{step}} \int_{\theta_+}^{\theta_-} \Big(|u_1(\tau)\phi_2'(\tau)| + |u_2(\tau)\phi_3'(\tau)|\Big) d\tau, \tag{4.56}$$

with $u_1(t)$ and $u_2(t)$ found from (4.18).

The constraints include all the aforementioned, i.e., the interpolation constraint, impact velocity and direction constraints, as well as those on θ_+ and κ. Some of these constraints

are dependent on the choice of the motion generator θ and must, therefore, be modified accordingly. In addition to these constraint, one can add the constraints (4.7) and (4.8), as well as heuristic constraints of choice, e.g., acceleration requirements at the beginning or end of the step, bounds on the maximum value $Y(\theta)$ and the allowed range of κ, both of which affects the step period T.

The problem is highly non-linear and non-convex such that any found solution is unlikely to be a global solution. Furthermore, since on can not simply pass the gradient and the Hessian of the objective function to the solver, this problem can be regarded as a *blind search*, where the solver must try to find estimates of these by itself. Thus, if the value of the chosen objective function is highly sensitive to changes in the parameters of the synchronization functions, the necessary computation time might be long in order to find any solution, which in turn is likely to be a poor local minima.

4.3 Results

In this section, the results from optimization on the full dynamical model and the reduced dynamics will be presented. The physical parameters used, rounded to three decimal places, are given in Table 4.1. In order to easily compare the different results, all the presented gaits have a step length of $15\,\mathrm{cm}$ and a $-0.01\,\mathrm{rad}$ lean angle of the torso measured in absolute angles. Note, however, that gaits have been found for a large array of different step lengths and lean angles. Furthermore, all the phase portraits of the angles corresponding to the gaits will be presented in absolute angles (see Figure 3.3a) in order to give a clear representation. All computations have been done on a PC with 16 GB of RAM and an Intel i7 four core processor running at $2.5\,\mathrm{GHz}$.

Table 4.1: Physcial parameters of the compass biped rounded to three decimal places.

Link 1		Link 2		Link 3	
l_1	$0.467\,\mathrm{m}$	l_2	$0.264\,\mathrm{m}$	l_3	$0.183\,\mathrm{m}$
L_1	$0.650\,\mathrm{m}$	L_2	$1.000\,\mathrm{m}$	L_3	$0.650\,\mathrm{m}$
m_1	$1.898\,\mathrm{kg}$	m_2	$3.799\,\mathrm{kg}$	m_3	$1.898\,\mathrm{kg}$
J_1	$0.092\,\mathrm{kg\,m^2}$	J_2	$0.445\,\mathrm{kg\,m^2}$	J_3	$0.092\,\mathrm{kg\,m^2}$

4.3.1 Full dynamical model

When searching for gaits on the full dynamical model, the optimization problems was formulated using the YALMIP optimization Toolbox[29] for MATLAB[3], together with the *fmincon* solver provided by MATLAB running the interior-point algorithm[4] due to the large number of parameters.

[3] MATLAB is a registered trademark of The MathWorks, Inc.

[4] For more information about the algorithms used in this text, namely: interior-point, active-set and sequential quadratic programming; see, e.g., [34].

The number of integration steps was set to 100, with the allowed maximum change of torque, $\dot{u}_{max}$, set to $0.5\,\mathrm{N\,m\,s^{-1}}$. The objective function was chosen as the quadratic cost functional of (4.10) with $Q = \mathbf{0}_6$ and $R = I_2$, thus only penalizing the actuator inputs. The computation time was approximately one hour, with the resulting gait shown in Figure 4.1. The gait has a step period of approximately $0.72\,\mathrm{s}$ and a maximum torque of $0.2\,\mathrm{N\,m}$. Interestingly, the actuators almost have a $\mathcal{C}^1$-smooth transition over the impact. However, Figure 4.1b clearly shows several abrupt changes in direction for both actuators over the step. This could possibly be remedied by lowering the allowed maximum change of torque and increasing the number of integration steps, which in turn could lead to a large increase in necessary computation time. Therefore, this was not attempted as the gaits found on the full dynamical model were found as to compare with those found on the reduced dynamics.

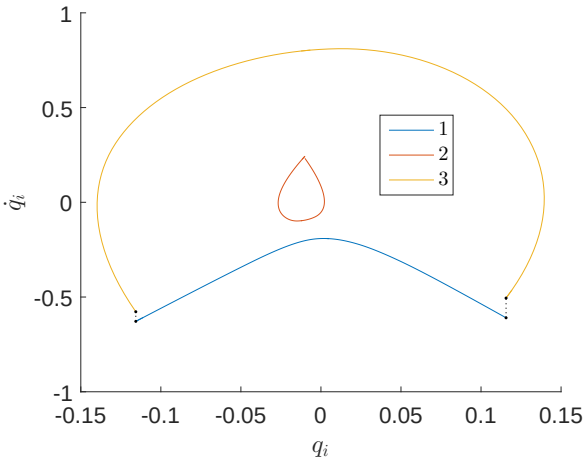
Another interesting property of the gait is that the energy of the biped is monotonically increasing over the step, as seen in Figure 4.2. This implies that the only energy added to the system by the actuators, is the amount needed in order to satisfy the energy loss at impact.

4.3.2 Reduced dynamics with stance leg as the motion generator

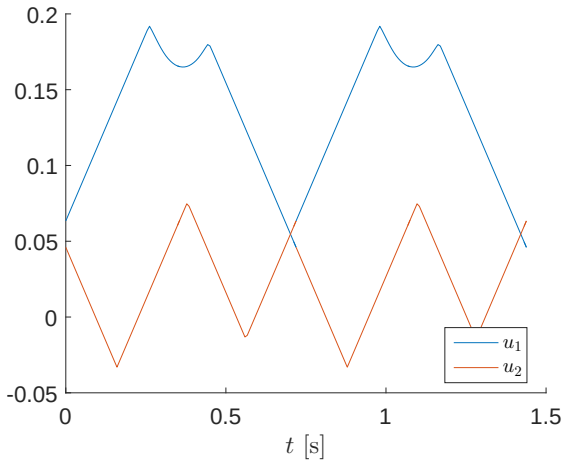
The gait searching procedure using the reduced dynamics with the motion generator chosen as $\theta = q_1$ made finding feasible gaits an easy task. The time needed to find a feasible gait was reduced drastically compared to the search on the full dynamics, rarely taken longer than a few minutes, with some gaits found in less than 5 seconds. The required time was found to depend heavily on factors like the choice of solver, the number of optimization variables, the chosen virtual holonomic constraints, the initial conditions of the search and the choice of objective function. The resulting gaits were also heavily dependent on these factors. However, no truly low-torque gaits were found compared to that found from the search on the full dynamical model.

The *fmincon* solver was used when optimizing on the reduced dynamics, with the following algorithms tested: interior-point, sequential quadratic programming (SQP) and active set. As previously mentioned from the search on the full dynamics, the interior-point algorithm is well suited for problems with many optimization variables and was found to be the solver which was by far the most likely to find a solution when the search was initiated poorly, such as badly chosen initial conditions and with many optimizations variables. This could often result in the SQP and active set algorithms terminating before any solution was found. The active set algorithm especially struggled if the order of the synchronization functions were higher than four, while the SQP algorithm could find solutions even for the order as high as eight in some cases if the initial conditions were chosen properly.

Although the interior-point algorithm was most likely to find a solution, it was often slow compared to the SQP algorithm and the resulting solution would often result in a higher final value of the objective functions compared to the SQP algorithm. Therefore, the SQP algorithm was mainly used in the search procedure. Should it fail to find a feasible solution, the interior-point algorithm would be used to find a feasible gait, and then the found values of this gait would be used as the initial conditions for a new search using the SQP algorithm to see if a better solution could be found. The active set algorithm



(a) Phase portrait in absolute angles.



(b) Nominal torque profile for the actuators over two steps.

Figure 4.1: Gait found on the full dynamical model.

was only used for problems with a small number of optimization variables and with initial conditions which were known to be good, in the sense that they were close to a feasible gait.

Since the choice of virtual holonomic constraints resulted in often quite different gaits, a comparison of these will be given next. In order to easily compare them, the order will be restricted to 6 and the initial conditions for the initial velocity of the motion generator and the parameter κ were set to $\dot{\theta}_+^2 = 0.3\,\mathrm{rad\,s^{-1}}$ and $\kappa = 1$, while the remaining constraint parameters were set to initial values depending on the type. All the aforementioned cost functionals were tested, but it was found that (4.11) resulted in the fastest convergence and

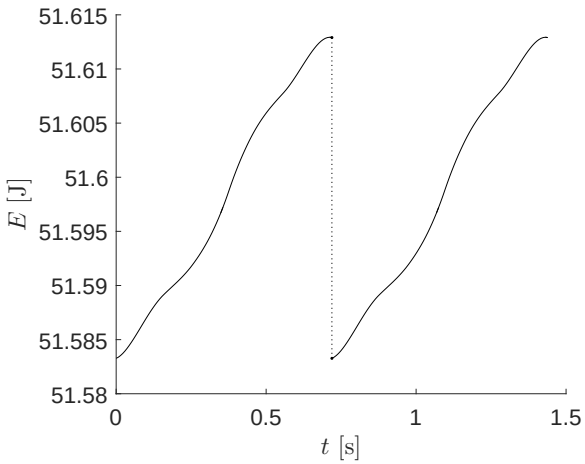


Figure 4.2: Energy of the biped for the gait found on the full dynamical model over two steps.

gave, not surprisingly, the best results as minimization of maximum required torque was deemed most important. All the presented results were thus found using this cost criteria.

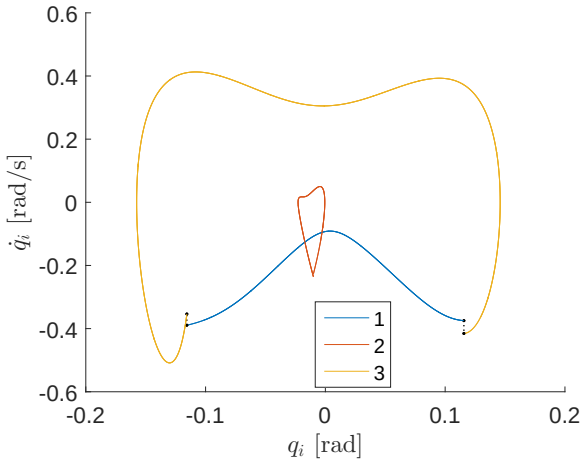
When a gait was found by the given solver, the gait was the simulated over a few steps using the ode45 solver provided by MATLAB with a relative tolerance of 10^{-6} and absolute tolerance 10^{-12}. The initial conditions $(\theta_0, \dot{\theta}_0)$ were used to compute the initial and final configurations of the biped by $x = [\Phi(\theta); \Phi'(\theta)\dot{\theta}]$. Impacts were detected using ode45's Event option using (3.21) and the assumption that vertical velocity of the tip of the swing leg should be negative. If the relation $\|x_0 - x_+\| < 10^{-6}$ was fulfilled, with x_+ computed using the end configuration and the impact maps (3.25), (3.24), the gait was deemed feasible. Otherwise, the number of integrations steps was increased and the search was starter over.

Regular polynomials

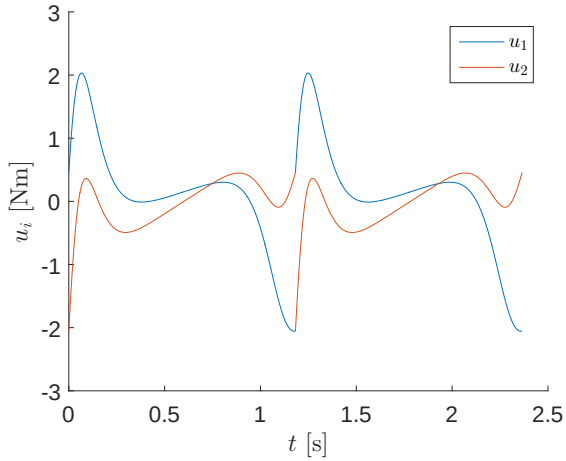
With the virtual holonomic constraints, $\phi_2(\theta)$ and $\phi_3(\theta)$, chosen as regular polynomials, the minimum order needed to find a feasible gait was 5 for the given step length and lean angle. With the order set to 6, the total number of optimization variables was 14. When searching, the initial values of the polynomial parameters were all set to zero.

The gait with the lowest torque requirement with a maximum of $2.06\,\mathrm{N\,m}$ and step period of 1.18 seconds was found using the interior-point algorithm and is shown in Figure 4.3. The cost of transport was found to be approximately 0.247. The search took approximately 153 seconds with 100 integration steps. With the same number of integration steps, the SQP algorithm converged after 14.5 seconds to one with a maximum of $2.62\,\mathrm{N\,m}$ and a step period 1.12 seconds. The active set algorithm converged to a gait with $19.7\,\mathrm{N\,m}$ and a step period of 0.37 seconds after 9.58 seconds.

The energy of the biped during the gait of Figure 4.3 is shown in Figure 4.4. Clearly, the energy is far from monotonically increasing, implying that the actuators add a lot of



(a) Phase portrait in absolute angles.



(b) Nominal torque profile for the actuators over two steps.

Figure 4.3: Gait found from the reduced dynamics with the motion generator chosen as $\theta = q_1$ and with the virtual holonomic constraints as 6th-order regular polynomials.

"unnecessary" energy to the system during the step in order to satisfy the synchronization of the VHCs rather than just that required in order to satisfy the impact conditions, as it was the case for the gait found on the full dynamical model.

Bezièr curves

When using Bezièr curves of order 6, the problem had a total of 12 optimization variables. Since the first and last parameter of each of the virtual holonomic constraints were

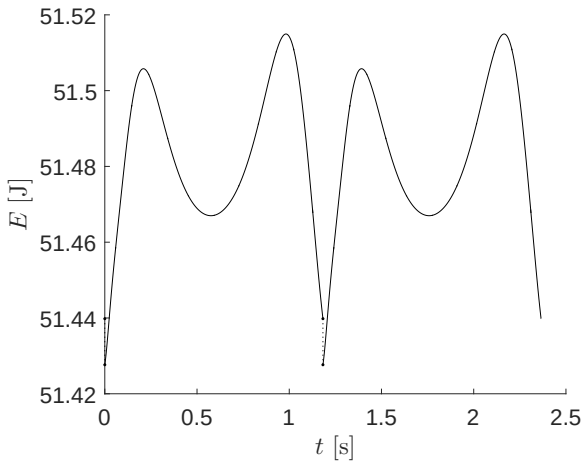


Figure 4.4: Energy of the biped for the gait found from the reduced dynamics over with 6th-order regular polynomials as virtual holonomic constraints.

computed before the search, the remaining parameters were chosen as follows for the ith constraint:

$$a_{i,j} = a_{i,0} + \delta a_i \cdot j, \quad j = 1, 2, .., m-1,$$

where m is the order of the curve and $\delta a_i = \big(a_{i,m} - a_{i,0}\big)/m + 1$ such that the initial curve lies within the end points of the first and last parameters.

The gait with the lowest torque found was the exact same gait found using regular polynomials (see Figure 4.3). However, the search now took 12.5 seconds using the SQP algorithm with 110 integration steps, compared to 153 second for the regular polynomials. The algorithm also converged to this solution with 100 integration steps but struggled to fully satisfy all the constraints into the required accuracy of 10^{-12}, and thus halted.

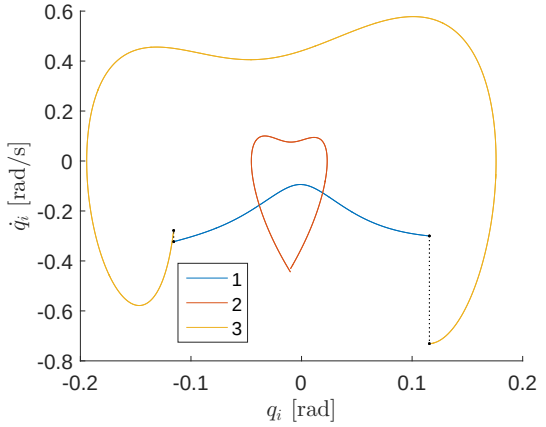
The active set algorithm converged to almost the same gait, which had a maximum of $2.08\,\mathrm{N\,m}$ with the same number of integration steps after 27.5 seconds, while the interior-point algorithm found the exact same gait as the SQP algorithm after 51 seconds of searching.

The lowest order for the given step length and lean angle using Bezièr curves was found to be 5. The gait was quite similar to that of Figure 4.3 with a maximum torque of $2.6\,\mathrm{N\,m}$ and a step period of approximately 1.2 seconds. The reason that the lowest order for feasible gaits was found to be 5 is not surprising considering that the first and last parameters are restricted to satisfy the configuration interpolation, while the second and the second last are constrained to satisfy the impact velocity mapping, as can be seen by (4.29). Thus the two remaining parameters are needed in order to satisfy the continuity of the actuators of the impact. It is therefore not surprising that a lower torque profile was found for a curve of order six as the remaining parameter could be chosen as to shape the torque profile throughout the step. The same reasoning can be used for the order of regular polynomials as well.

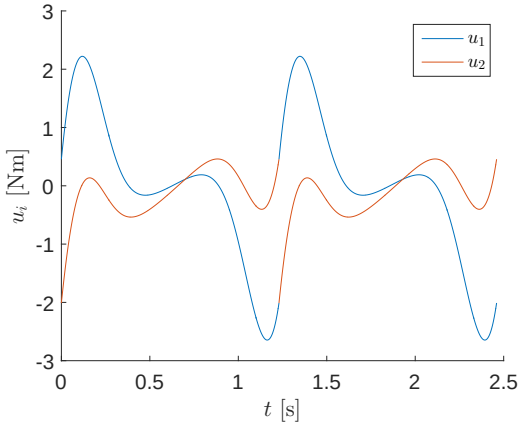
Trigonometric polynomials

With the virtual holonomic constraints chosen as trigonometric polynomials, the initial values of the parameters for the search were all set to zero except for the $a_{2,0}$ and $a_{3,0}$ which were set to the values from (4.26). The harmonics ω_2, ω_3, were either set to a constant value or left as optimization variables constrained to be within the interval $[\pi/12, 2\pi]$.

The lowest order in which a feasible gait was found was 3 using the interior-point algorithm after 162 seconds of searching with the harmonics set to $\omega_2 = \omega_3 = 1$. The gait shown in Figure 4.5; it had a maximum torque requirement of $-2.65\,\mathrm{N\,m}$, with a step period of 1.23 seconds and cost of transport approximately 0.266. The continuity of the actuators is also very close to being $\mathcal{C}^1$-smooth over the impact.



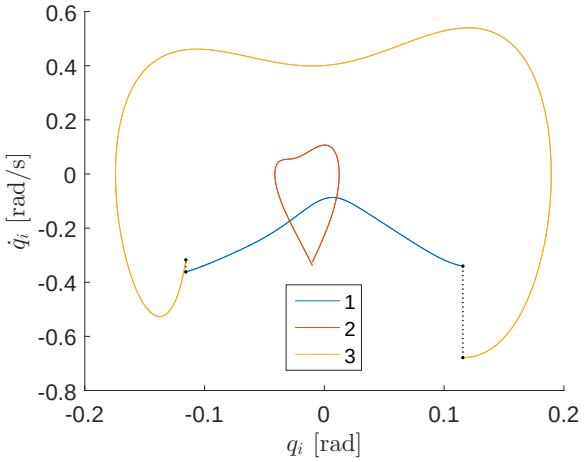
(a) Phase portrait in absolute angles.



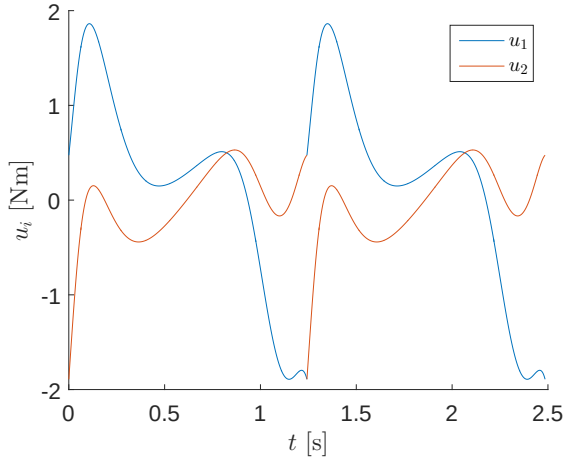
(b) Nominal torque profile for the actuators over two steps.

Figure 4.5: Gait found from the reduced dynamics with the motion generator chosen as $\theta = q_1$ and with the virtual holonomic constraints as 3rd-order trigonometric polynomials.

With an order of 6, the optimization problem had a total of 16 variables. Due to this, only the interior-point and SQP algorithms were able to find a feasible gait. The gait with the lowest torque requirement was found using the SQP algorithm after approximately 15 seconds. The gait is shown in Figure 4.6. The maximum required torque was $1.89\,\mathrm{N\,m}$, with a step period of 1.24 seconds and cost of transport approximately 0.244.



(a) Phase portrait in absolute angles.



(b) Nominal torque profile for the actuators over two steps.

Figure 4.6: Gait found from the reduced dynamics with the motion generator chosen as $\theta = q_1$ and with the virtual holonomic constraints as 6th-order trigonometric polynomials.

4.3.3 Reduced dynamics with an unknown motion generator

In order to see if the choice of $\theta = q_1$ as the motion generator is a reasonable choice, the optimization procedure for the reduced dynamics was also run with an "unknown" motion generator, which was simply chosen as $\theta = s \in [0, 1]$. By unknown, it is meant that it is not a known property found directly from the configuration, q, of the system. However, it must be noted that any found solution still represents a feasible gait for the mechanical system as the reduced dynamics correspond to the enforcement of the VHCs.

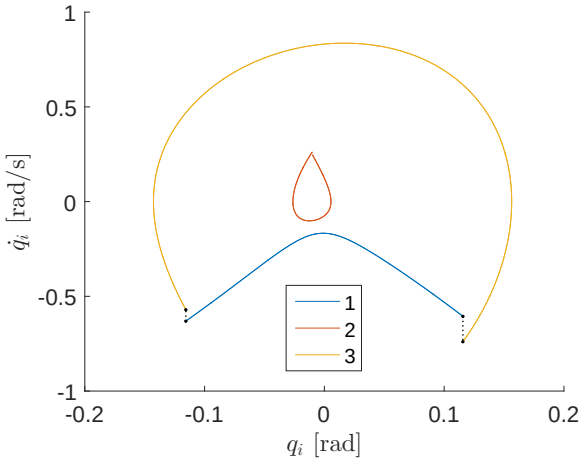
All the VHCs were chosen as 6th-order Bezièr curves, as described in Section 4.2.2, such that $\partial s/\partial\theta = 1$. The initial and end values were thus $\theta_+ = 0$ and $\theta_- = 1$, such that the constraints on the velocity and on $\alpha(\theta)$ had to be modified to $\dot{\theta} > 0$ and $\alpha(\theta) < 0$ for all $\theta \in [\theta_+, \theta_-]$. As the parameters of the synchronization function $\phi_1(\theta)$ had to be added to the problem, in addition to those of $\phi_2(\theta)$ and $\phi_3(\theta)$, the total number of optimization variables was 17. The SQP algorithm of *fmincon* was chosen as the solver.

The optimization problem was run with 80 integration steps[5] and took approximately 52 seconds, with the resulting in gait shown in Figure 4.7. The gait has a maximum torque requirement of approximately $-0.0848\,\mathrm{N\,m}$, with a step period of 0.76 seconds and cost of transport approximately 0.0031. It is also quite similar to that found on the full dynamical model (c.f., Figure 4.1) but with half the required torque and a smoother overall torque profile, as well as a $\mathcal{C}^1$-smooth torque continuity over the impact. Furthermore, as seen in Figure 4.8, the energy over the step is very close to being strictly monotonically increasing, except for a small decrease of energy at the beginning of the step.

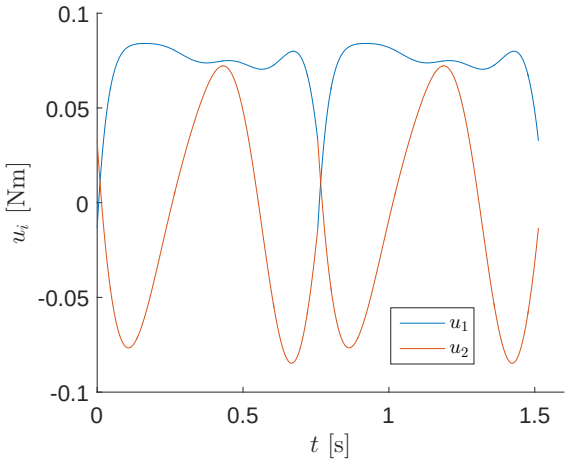
By minimizing the COT using the objective function (4.56), the gait shown in Figure 4.10 was found. It had a COT of 0.0025, with the energy of the system strictly increasing over the step, as seen in Figure 4.9. However, it had a maximum torque of $0.285\,\mathrm{N\,m}$ and not a $\mathcal{C}^1$-smooth torque continuity over the impact. Although, the torque requirements for both actuators is zero at both the beginning and end of the step, which is a useful property for implementation of a real system.

By changing the step length and lean angle, one could find gaits with even lower torque requirements and COT using this procedure. As an example, by minimizing the COT for a step length of $10\,\mathrm{cm}$ and lean angle of $-0.0015\,\mathrm{rad}$, the gait in Figure 4.11 was found. It had a COT of approximately $8 \cdot 10^{-4}$ and maximum torque requirement of $0.0425\,\mathrm{N\,m}$.

[5] Surprisingly, increasing the number of integration steps resulted in gaits with slightly worse performance. This is likely due to the problem having many local minima.



(a) Phase portrait in absolute angles.



(b) Nominal torque profile for the actuators over two steps.

Figure 4.7: Gait found from the reduced dynamics with the MG chosen as $\theta = s \in [0, 1]$ and with the VHCs as 6th-order Bezièr curves.

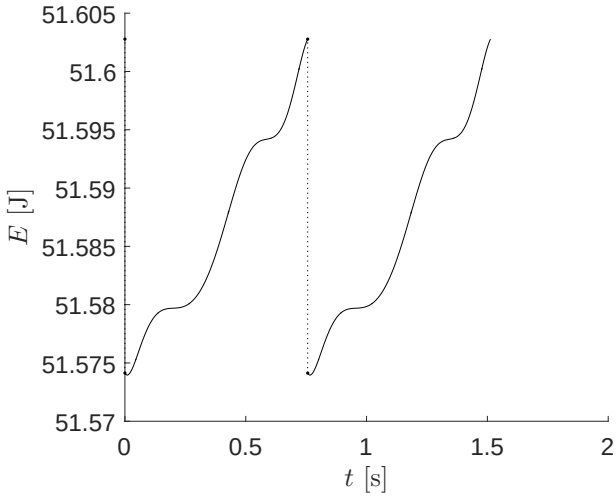


Figure 4.8: Energy of the biped for the gait found from the reduced dynamics with $\theta = s \in [0, 1]$ as the MG and 6th-order Bezièr curves as VHCs.

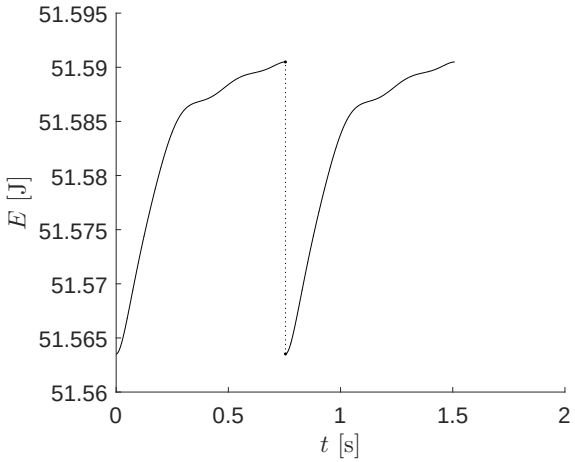
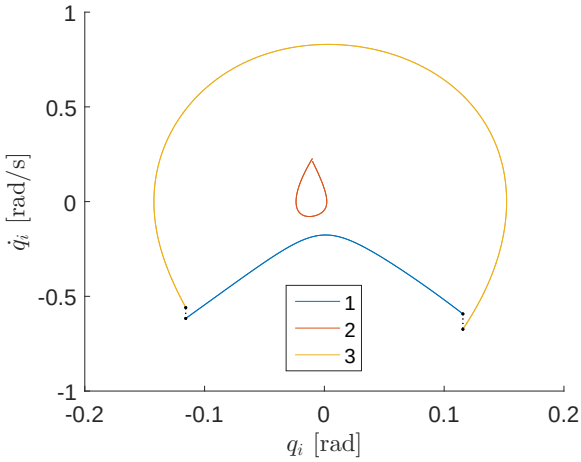
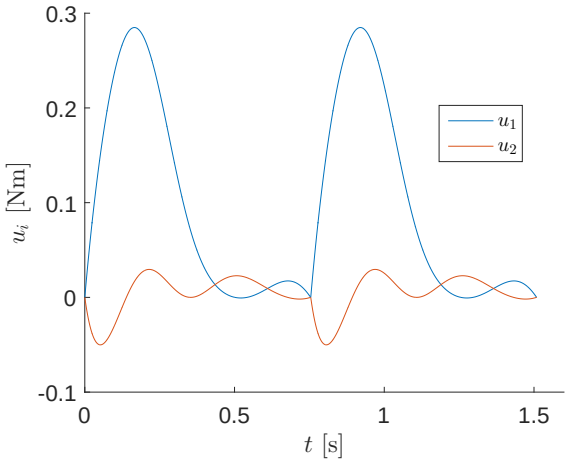


Figure 4.9: Energy of the biped for the gait found by minimizing (4.56) with $\theta = s \in [0, 1]$ as the MG and 6th-order Bezièr curves as VHCs.

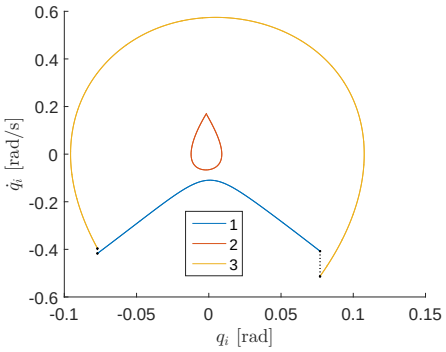


(a) Phase portrait in absolute angles.

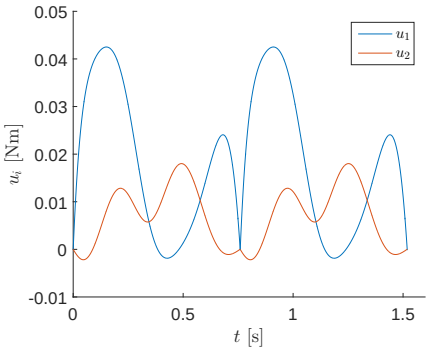


(b) Nominal torque profile for the actuators over two steps.

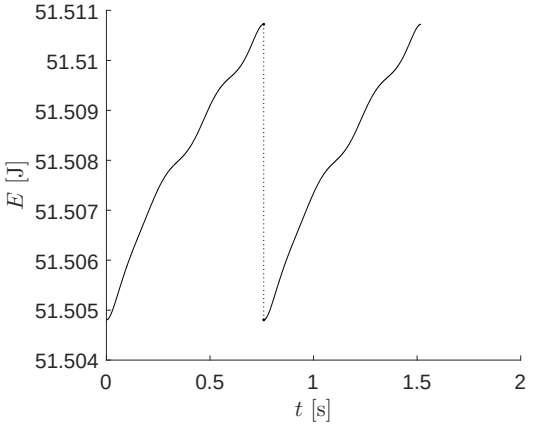
Figure 4.10: Gait found by minimizing (4.56) with the MG chosen as $\theta = s \in [0, 1]$ and with the VHCs as 6th-order Bezièr curves.



(a) Phase portrait in absolute angles.



(b) Nominal torque profile for the actuators.



(c) Energy of the system over two steps.

Figure 4.11: Gait found by minimizing (4.56) for a step length 0f $10\,\mathrm{cm}$ and lean angle of $-0.0015\,\mathrm{rad}$ with the MG chosen as $\theta = s \in [0, 1]$ and with the VHCs as 6th-order Bezièr curves.

4.4 Discussion

In this chapter, a method was presented for finding feasible gaits on the three-link compass biped by searching on the reduced dynamics found by enforcing geometric relations, called virtual holonomic constraints, between the configuration variables of the system and a scalar motion generator. Moreover, a basic method for finding feasible gaits on the full dynamical model was presented for comparison.

Searching for gaits on the reduced dynamics resulted in a dramatic reduction in the time needed for finding gaits compared to the search on the full dynamical model. Yet, the resulting gaits were found be heavily dependent on several factors. The main factor found was the choice of the motion generator. Although the angle of stance leg seems to be a natural choice, the results showed that it clearly restricts the search space such that the gaits found have much high COT and maximum required torque than what is possible. There may be many reasons for this. Firstly, with q_1 as the motion generator, the fact that $\phi_1^{'}(\theta) = 1$ and $\phi_1^{''}(\theta) = 0$ limits the possible values of $\Phi^{'}(\theta)$ and $\Phi^{''}(\theta)$ due the impact constraints (4.42) and (4.44), as well as the actuator interpolation constraint (4.54). Moreover, the reduced dynamics (4.14) becomes quite restricted, which again affects the required torques through (4.16). For these reasons, it can be advantageous to find what is feasible for the system by using the procedure with an unknown MG as in Section 4.3.3. Then, one should search for a MG which can reproduce similar results if desired.

It is also interesting to note the resulting solutions dependence on factors like the initial conditions, choice of virtual holonomic constraints, the solver and number of integration steps. These are clear indications that the highly nonlinear search-space has a lot of local minima for all choices of objective functions. Since no gradient nor Hessian is passed to the solver, this further complicates the search. Thus, a linear comparison system, as mentioned in [36], if it exists, could possibly result in faster convergence to more "global" solutions as on could pass approximations of the gradient and Hessian to the solvers based on an already found gait.

The observation from Figures 4.2, 4.8, 4.9 and 4.11c that the energy of the robot is strictly increasing, or close to it, over the step for gaits with low torque requirements and COT is very interesting. This is evidence that the gaits with the lowest torque requirements are those which are the closest to being passive, in the sense that the only added energy to the system is that required in order to satisfy the energy loss at impact, resulting in a very low energetic cost of transport. However, comparing the gaits from Figures 4.7 and 4.10 shows that the smallest torque requirements does not correspond to the lowest COT, and vice versa. Moreover, Figure 4.11 also shows that such factors are, not surprisingly, heavily dependent on the step length and lean angle of the torso. Thus, if one wants to only minimize, say, COT, one should have the step length and lean angle as optimization variables, which is likely to lead to a large increase in the necessary search time for finding feasible solutions.

It is important to note that the COT and torques for the gaits found do not represent what is feasible on a real system, as the factors like friction and uncertainty in system parameters will increase the necessary energy which must be added to the system. Thus, the COT of the gaits found can not be compared to other bipedal robots (c.f. Figure 1.1) before experiments on a real biped robot have been performed. Yet, they give clear indications that energy efficient gaits are feasible and exists for the compass biped.

Chapter 5

Development of Stabilizing Controllers

In the previous chapter, gaits for the three-link compass biped were found using the virtual holonomic constraints approach. The gaits corresponded to limit cycles induced on the system through open-loop control action which were found by enforcing the geometric relations of the VHCs upon the system. However, the found limit cycles are unlikely to be stable, even for small perturbation away from the nominal cycle. For this reason, the development of an orbitally stabilizing controller which increases the stability region of the walking motion is needed.

In this chapter, exponentially orbritally stabilizing controllers will be developed by stabilizing the origin of the linearized transverse dynamics corresponding to the limit cycle by solving the Riccati differential equation with one jump. Herein, controllers will be developed using two different methods. In Section 5.1, the method presented in [46] will be used to construct a stabilizing controller where the angle of the stance leg is used as the motion generator (MG). However, as shown in the previous chapter, the gaits which were found with this MG required far more torque and had a much higher minimum COT that what is feasible on the mechanical system. For this reason, a method of constructing a stabilizing controller where the MG is found through a projection from the configuration of the biped will be presented in Section 5.2. This method transfers the method presented in [37] to an underactuated hybrid system and uses excessive transverse coordinates in order to generate the stabilizing controller. Lastly, in Section 5.3, a brief discussion on the found controllers and presented results will be given.

5.1 Motion Generator as the Angle of the Stance Leg

In this section, a stabilizing controller will be developed using the angle of the stance leg as the MG. First, the method of partial feedback linearization (PFL), which was presented in Section 2.4.2, will be used to bring parts of the transverse dynamics on linear form. This

results in an open-loop controller which, on the nominal solution, makes the dynamical system exactly follow the limit cycle corresponding the the virtual holonomic constraints. Then, the remaining internal dynamics will be linearized in order to form the linearized transverse dynamics. A mapping for the transverse coordinates will then be developed to take into account the instantaneous change in the velocities and swapping of coordinates due to the impact, and how this affect the transverse states. Next, a stabilizing controller will be developed by solving the Riccati differential equation with one jump using a semi-definite programming approach. Lastly, some results from numerical simulations will be presented.

5.1.1 Partial Feedback Linearization

With the motion generator chosen as $\theta = q_1$, the synchronization vector is given by

$$\Phi(\theta) = [\theta; \phi_2(\theta); \phi_3(\theta)].$$

Thus, one can introduce the new set of coordinates

$$y_1 = q_1 - \theta, \quad y_2 = q_2 - \phi_2(\theta), \quad y_3 = q_3 - \phi_3(\theta), \tag{5.1}$$

corresponding to the deviations from the VHCs. Herein, $y_1 \equiv 0$ as $\theta = q_1$, such that one can rewrite the generalized coordinates as

$$\begin{aligned} q_1 &= \theta, \\ q_2 &= y_2 + \phi_2(\theta), \\ q_3 &= y_3 + \phi_3(\theta), \end{aligned} \tag{5.2}$$

with their velocity and acceleration vectors given by

$$\dot{q} = L(\theta, y) \begin{bmatrix} \dot{y} \\ \dot{\theta} \end{bmatrix}, \tag{5.3a}$$

$$\ddot{q} = L(\theta, y) \begin{bmatrix} \ddot{y} \\ \ddot{\theta} \end{bmatrix} + N(\theta, \dot{\theta}, y, \dot{y}), \tag{5.3b}$$

where $y = [y_2; y_3]$. Clearly, since q_1 is only dependent on θ, the function $h(y_2, y_3, \theta)$ from (2.41) is equal to zero, such that both its gradient and Hessian is zero as well. Thus,

$$L(\theta, y) = L(\theta) = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & \phi_2^{'}(\theta) \\ 0 & 1 & \phi_3^{'}(\theta) \end{bmatrix}, \tag{5.4a}$$

$$N(\theta, \dot{\theta}, y, \dot{y}) = N(\theta, \dot{\theta}) = \Phi^{''}(\theta)\dot{\theta}^2. \tag{5.4b}$$

Now, by using the procedure of (2.46), one can deduce the 2-dimensional subsystem

$$\ddot{y} = K(\theta, y)u + R(\theta, \dot{\theta}, y, \dot{y}), \tag{5.5}$$

where the functions $K(\theta, y)$ and $R(\theta, \dot{\theta}, y, \dot{y})$ are defined in (2.48). This can be done since the function $L(\theta)$ of (5.4a) is clearly non-singular; indeed

$$L(\theta)^{-1} = \begin{bmatrix} -\phi_2^{'}(\theta) & 1 & 0 \\ -\phi_3^{'}(\theta & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

The question now remains to check if the function $K(\theta, y)$ is non-singular such that an exact PFL controller as (2.49) can be used, which for convenience is repeated here:

$$u = K(\theta, y)^{-1}\big(v - R(\theta, \dot{\theta}, y, \dot{y})\big). \tag{5.6}$$

This controller leads to the simple form of (2.54), i.e., $\ddot{y} = v$. If $K(\theta, y)$ can be singular along, or close to, the solution of interest, then a controller of the form (2.55) must be used instead, such that one is left with the more complicated form of (2.61).

Using (2.48), one finds that the inverse of the matrix function $K(\theta, y)$ can be written as

$$K(\theta, y)^{-1} = \frac{1}{k_d(\theta, y)} \begin{bmatrix} k_{11}(\theta, y) & k_{12}(\theta, y) \\ k_{21}(\theta, y) & k_{22}(\theta, y) \end{bmatrix} \tag{5.7}$$

where

$$\begin{aligned} k_d(\theta, y) =& (L_1\phi_2^{'}(\theta)l_2m_2 + 2L_1l_2m_2)\cos(\phi_2(\theta) + y_2) + (L_1\phi_2^{'}(\theta)l_3m_3 \\ &+ L_1\phi_3^{'}(\theta)l_3m_3 + 2L_1l_3m_3)\cos(\phi_2(\theta) + y_2 + \phi_3(\theta) + y_3) \\ &+ ((l_2^2m_2 + l_3^2m_3 + J_2 + J_3)\phi_2^{'}(\theta)) + ((l_3^2m_3 + J_3)\phi_3^{'}(\theta)) \\ &+ L_1^2m_2 + L_1^2m_3 + m_1l_1^2 + l_2^2m_2 + l_3^2m_3 + J_1 + J_2 + J_3, \end{aligned} \tag{5.8}$$

such that the denominators of the matrix elements, k_{11}, k_{12}, k_{21}, k_{22}, are all equal to unity. Therefore, the inverse of the matrix function $K(\theta, y)$ is well defined as long as $k_d \neq 0$. It should be noted that $k_d(\theta, y)$ is not the determinant of $K(\theta, y)$, which it can be shown is given by

$$\det\big(K(\theta, y)\big) = \frac{k_d(\theta, y)}{\det\big(M(\theta, y)\big)}.$$

By close inspection, one can observe from (5.8) and (4.15a) that $k_d(\theta, 0) = \alpha(\theta)$ since $\phi_1^{'}(\theta) = 1$ when $\theta = q_1$. Noting from Theorem 2.4.1 that $\alpha(\theta)$ is bounded away from zero along all solutions of interest, it follows that there exists some $\epsilon > 0$ such that for $\forall y_2, y_3$ where $\|y\| < \epsilon$, then $|k_d(\theta_*(t), y)| > 0$ along some solution of interest $\theta_*(t)$. How large the perturbation parameter ϵ will be is hard say without knowing the functions $\phi_2^{'}(\theta)$ and $\phi_3^{'}(\theta)$, but as y_2 and y_3 only appear inside the sinusoidal functions in $k_d(\theta, y)$, which are bounded inside $[-1, 1]$, it is fair to assume that ϵ will be quite large, possibly even infinite. However, since q_1 might also get perturbed outside the boundaries of solution $\theta_*(t)$, one must also verify that the functions $\phi_2^{'}(\theta)$, $\phi_3^{'}(\theta)$ does not make $k_d(\theta, y) = 0$ for $\theta = q_1$ outside the boundaries of $\theta_*(t)$. Thus, one should always check that $k_d(\theta, y)$ is bounded away from zero for reasonable values of ϵ and for q_1 outside the boundaries of the solution of interest, either by algebraic or by numerical methods, before determining to use the controller (2.49). However, it will be assumed for the remainder of this chapter that the matrix function $K(\theta, y)$ is nonsingular in a sufficiently large region about the nominal solution such that the controller (5.6) can be used.

5.1.2 Remaining Internal Dynamics

When the PFL controller (5.6) is used, the dynamics of the transverse coordinates y are brought to the linear form $\ddot{y} = v$. Thus, one must find the remaining internal dynamics such that the one can linearize the dynamics corresponding to the integral function $I(\cdot)$.

Using (2.51), one finds that the internal dynamics can be written as

$$\hat{\alpha}(\theta, y)\ddot{\theta} + \hat{\beta}(\theta, y)\dot{\theta}^2 + \hat{\gamma}(\theta, y) = g_{\dot{y}}(\theta, \dot{\theta}, y, \dot{y})\dot{y} + g_v(\theta, y)v \tag{5.9}$$

which is equal to the reduced dynamics (4.14) when $y = \dot{y} = v = 0$, which, from now on, will be referred to as the *zero-dynamics*. Herein, the vector functions on the right-hand side are given by

$$g_v(\theta, y) = [g_{v_1}(\theta, y), g_{v_2}(\theta, y)] \quad \text{and} \quad g_{\dot{y}}(\theta, \dot{\theta}, y, \dot{y}) = [g_{\dot{y}_2}(\theta, \dot{\theta}, y, \dot{y}), g_{\dot{y}_3}(\theta, \dot{\theta}, y, \dot{y})].$$

The full expression of these functions can be found in (B.4) in Appendix B.

In order to get the left-hand side of (5.9) to equal the zero-dynamics, one can add and subtract the reduced dynamics (4.14) to the left-hand side such that one can write

$$\begin{aligned}\alpha(\theta)\ddot{\theta} + \beta(\theta)\dot{\theta}^2 + \gamma(\theta) =& \big(\alpha(\theta) - \hat{\alpha}(\theta, y)\big)\ddot{\theta} + \big(\beta(\theta) - \hat{\beta}(\theta, y)\big)\dot{\theta}^2 + \big(\gamma(\theta) - \hat{\gamma}(\theta, y)\big) \\ &+ g_{\dot{y}}(\theta, \dot{\theta}, y, \dot{y})\dot{y} + g_v(\theta, y)v.\end{aligned}$$

In order to get this on the form (2.52), one can take a first-order Taylor expansions of

$$g(\theta, \dot{\theta}, \ddot{\theta}, y) = \big(\alpha(\theta) - \hat{\alpha}(\theta, y)\big)\ddot{\theta} + \big(\beta(\theta) - \hat{\beta}(\theta, y)\big)\dot{\theta}^2 + \big(\gamma(\theta) - \hat{\gamma}(\theta, y)\big)$$

in terms of the integral function $I(\cdot)$ and y.[1] The terms of $I(\cdot)$ of (2.52) are found using the orthogonal property of the integral function, i.e.,

$$g_I(\theta, \dot{\theta}, \ddot{\theta}) = \frac{1}{2(\dot{\theta}^2 + \ddot{\theta}^2)}\left[\dot{\theta}\frac{\partial g}{\partial \dot{\theta}} - \ddot{\theta}\frac{\partial g}{\partial \theta}\right]_{y=0}.$$

However, by noting that $g(\theta, \dot{\theta}, \ddot{\theta}, 0) = \frac{\partial}{\partial \theta} g(\theta, \dot{\theta}, \ddot{\theta}, 0) = \frac{\partial}{\partial \dot{\theta}} g(\theta, \dot{\theta}, \ddot{\theta}, 0) = 0$, it is clear that $g_I(\theta, \dot{\theta}, \ddot{\theta}) = 0$, so one only needs to take the expansion in terms of y.

The first-order Taylor expansion of $g(\theta, \dot{\theta}, \ddot{\theta}, y)$ in terms of y is equivalent to the expansion of the functions $\hat{\alpha}(\theta, y)$, $\hat{\beta}(\theta, y)$ and $\hat{\gamma}(\theta, y)$ about $y = 0$, i.e.,

$$\begin{aligned}g_y &= \left[\ddot{\theta}\frac{\partial}{\partial y}\big(\alpha(\theta) - \hat{\alpha}(\theta, y)\big) + \dot{\theta}^2\frac{\partial}{\partial y}\big(\beta(\theta) - \hat{\beta}(\theta, y)\big) + \frac{\partial}{\partial y}\big(\gamma(\theta) - \hat{\gamma}(\theta, y)\big)\right]_{y=0} \\ &= -\left[\ddot{\theta}\frac{\partial}{\partial y}\hat{\alpha}(\theta, y) + \dot{\theta}^2\frac{\partial}{\partial y}\hat{\beta}(\theta, y) + \frac{\partial}{\partial y}\hat{\gamma}(\theta, y)\right]_{y=0},\end{aligned} \tag{5.10}$$

[1] If Hadamard's Lemma is used directly, it results in the form $\hat{\alpha}(\theta, y) = \big(\alpha(\theta) - \hat{\alpha}(\theta, y)\big)\frac{1}{y}y$ such that one can not simply set $y = 0$ for the transverse linearization. Thus, one would need to use L'Hospital's Rule, which in this case is equivalent to the first-order Taylor expansion.

with $g_y := g_y(\theta, \dot{\theta}, \ddot{\theta})$ and where the zero-order terms have been excluded as they disappear since $\alpha(\theta) = \hat{\alpha}(\theta, 0)$, $\beta(\theta) = \hat{\beta}(\theta, 0)$ and $\gamma(\theta) = \hat{\gamma}(\theta, 0)$. The full expression of $g_y(\theta, \dot{\theta}, \ddot{\theta}) = [g_{y_2}(\theta, \dot{\theta}, \ddot{\theta}), g_{y_3}(\theta, \dot{\theta}, \ddot{\theta})]$ is given in (B.5). The remaining internal dynamics thus has the form

$$\alpha(\theta)\ddot{\theta} + \beta(\theta)\dot{\theta}^2 + \gamma(\theta) = g_y(\theta, \dot{\theta}, \ddot{\theta})y + g_{\dot{y}}(\theta, \dot{\theta}, y, \dot{y})\dot{y} + g_v(\theta, y)v. \tag{5.11}$$

It now remains to use (5.11) to find the transverse linearization of the transverse coordinate $I(\cdot)$ using the relation of Theorem 2.4.2. However, it turns out that the transverse coordinates proposed in Section 2.4.2 can be simplified by making them independent of the integral function (2.36). This is done by using Proposition 4.2.2 and the relation (4.20) by introducing the new integral function (4.51) as the new transverse coordinate, which was found by multiplying (4.48) by $\big(\alpha(\theta)\big)^2$, yielding

$$\hat{I}(\theta, \dot{\theta}, \theta_0, \dot{\theta}_0) = \big(\alpha(\theta)\dot{\theta}\big)^2 - \big(\alpha(\theta_0)\dot{\theta}_0\big)^2 + 2\int_{\theta_0}^{\theta} \alpha(\tau)\gamma(\tau)d\tau. \tag{5.12}$$

Now, if the linearized transverse dynamics shall be independent of the new integral function, then its time-derivative must not be dependent on itself when the right-hand side of the zero dynmaics is not zero, which was the case for (2.38) in Theorem 2.4.2.

In order to show that the time-derivative of (4.51) is independent of itself when the zero dynamics has the form (2.37), one can start by noting that

$$\frac{\partial}{\partial\theta}\hat{I}(\theta, \dot{\theta}, \theta_0, \dot{\theta}_0) = 2\dot{\theta}^2\alpha(\theta)\frac{\partial}{\partial\theta}\alpha(\theta) + 2\alpha(\theta)\gamma(\theta) = 2\dot{\theta}^2\alpha(\theta)\beta(\theta) + 2\alpha(\theta)\gamma(\theta)$$

and

$$\frac{\partial}{\partial\dot{\theta}}\hat{I}(\theta, \dot{\theta}, \theta_0, \dot{\theta}_0) = 2\dot{\theta}\big(\alpha(\theta)\big)^2.$$

Now, since

$$\frac{d}{dt}\hat{I}(\theta, \dot{\theta}, \theta_0, \dot{\theta}_0) = \frac{\partial}{\partial\theta}\hat{I}(\theta, \dot{\theta}, \theta_0, \dot{\theta}_0)\dot{\theta} + \frac{\partial}{\partial\dot{\theta}}\hat{I}(\theta, \dot{\theta}, \theta_0, \dot{\theta}_0)\ddot{\theta}$$

and

$$\ddot{\theta} = \frac{1}{\alpha(\theta)}\Big(W - \beta(\theta)\dot{\theta}^2 - \gamma(\theta)\Big),$$

one obtains, after some simplification,

$$\frac{d}{dt}\hat{I}(\theta, \dot{\theta}, \theta_0, \dot{\theta}_0) = 2\dot{\theta}\alpha(\theta)W, \tag{5.13}$$

which is independent of $\hat{I}(\theta, \dot{\theta}, \theta_0, \dot{\theta}_0)$ as long as W is also independent of it.

5.1.3 The Hybrid Linearized Transverse Dynamics

Using the remaining internal dynamics (5.11) together with (5.13), the transverse linearization around the solution $(\theta_*, \dot{\theta}_*) = (\theta_*(t), \dot{\theta}_*(t))$ yields the matrices

$$\mathcal{A}(t) = \begin{bmatrix} 0 & a_{12}(t) & a_{13}(t) \\ \mathbf{0}_{2\times1} & \mathbf{0}_2 & I_2 \\ \mathbf{0}_{2\times1} & \mathbf{0}_2 & \mathbf{0}_2 \end{bmatrix} \quad \text{and} \quad \mathcal{B}(t) = \begin{bmatrix} b_1(t) \\ \mathbf{0}_2 \\ I_2 \end{bmatrix}, \tag{5.14}$$

where

$$a(t) = 2\dot{\theta}_* \alpha(\theta_*), \tag{5.15}$$

$$a_{12}(t) = a(t) g_y(\theta_*, \dot{\theta}_*, \ddot{\theta}_*, \mathbf{0}), \tag{5.16}$$

$$a_{13}(t) = a(t) g_{\dot{y}}(\theta_*, \dot{\theta}_*, \mathbf{0}, \mathbf{0}), \tag{5.17}$$

$$b_1(t) = a(t) g_v(\theta_*, \mathbf{0}). \tag{5.18}$$

The linearized transverse dynamics in the continuous swing phase thus has the form

$$\dot{x}_\perp = \mathcal{A}(t) x_\perp + \mathcal{B}(t) v, \quad t \in [0, T). \tag{5.19}$$

with $x_\perp = [I(\cdot), y_2, y_3, \dot{y}_2, \dot{y}_3]^T$. It thus remains to find the discrete mapping $\mathcal{L} := dF^{TS}$ from (2.84) to account for the change of the coordinates due to impact.

In order to find this mapping, one can first note that the impact surface Γ_- from (3.21) can be equivalently written as

$$\Gamma_- = \{x = [q; \dot{q}] \in TQ \subseteq \mathbb{R}^{2n} : \; 2q_1 + q_2 + q_3 = 0\}, \tag{5.20}$$

which is already on linear form, meaning that $T\Gamma_- = \Gamma_-$. The normal vector $\vec{m}_-$ of $T\Gamma_-$ is then simply found by taking the gradient; hence

$$\vec{m}_- = [2, 1, 1, 0, 0, 0]^T. \tag{5.21}$$

By further noting from (2.86) that the normal vector $\vec{n}(t)$ of the moving Poincarè section $S(t)$ is given by the vector field of $x(t)$ along the nominal solution $(\theta_*, \dot{\theta}_*) = (\theta_*(t), \dot{\theta}_*(t))$, one can see that it can be written as

$$\vec{n}(t) = \begin{bmatrix} \Phi^{'}(\theta_*)\dot{\theta}_* \\ \Phi^{''}(\theta_*)\dot{\theta}_*^2 + \Phi^{''}(\theta_*)\ddot{\theta}_* \end{bmatrix}, \tag{5.22}$$

where $\ddot{\theta}_*$ can be found from (2.34) using θ_* and $\dot{\theta}_*$.

The next step is then to find some relation between the states of the system $x(t) = [q; \dot{q}]$ and the transverse coordinates. As stated in [12], one can find the linear mapping matrix $\Lambda(t)$ of (2.85) between small increments of the transverse coordinates in terms of the state variables as

$$\Lambda(t) = \begin{bmatrix} -2\ddot{\theta}_*\alpha(\theta_*)^2 & 0 & 0 & 2\dot{\theta}_*\alpha(\theta_*)^2 & 0 & 0 \\ -\phi_2^{'}(\theta_*) & 1 & 0 & 0 & 0 & 0 \\ -\phi_3^{'}(\theta_*) & 0 & 1 & 0 & 0 & 0 \\ -\phi_2^{''}(\theta_*)\dot{\theta}_* & 0 & 0 & -\phi_2^{'}(\theta_*) & 1 & 0 \\ -\phi_3^{''}(\theta_*)\dot{\theta}_* & 0 & 0 & -\phi_3^{'}(\theta_*) & 0 & 1 \end{bmatrix}, \tag{5.23}$$

such that $\Delta x_\perp = \Lambda(t)\Delta x$. Herein, the relations

$$\frac{\partial I}{\partial \theta} = -2\ddot{\theta}\alpha(\theta)^2 \quad \text{and} \quad \frac{\partial I}{\partial \dot{\theta}} = 2\dot{\theta}\alpha(\theta)^2$$

have been used.

Now, since the change of impact velocity mapping $P_{\dot{q}}(q_-)$ is linear and constant as q_- is assumed to be fixed, the mapping of the states dF from (2.83) is given by

$$dF = \begin{bmatrix} P_q & \mathbf{0}_3 \\ \mathbf{0}_3 & P_{\dot{q}}(q_-) \end{bmatrix}. \tag{5.24}$$

Thus, the discrete mapping for the transverse coordinates is found to be

$$\mathcal{L} := \Big[P^+_{\vec{n}(0)}\Big] dF \Big[P^-_{\vec{n}(T_h)}\Big], \tag{5.25}$$

with $P^+_{\vec{n}(0)}$ and $P^-_{\vec{n}(T_h)}$ defined in (2.95) and (2.92), respectively. Hence, the linearized transverse dynamics for the hybrid system takes the form

$$\begin{aligned} \dot{x}_\perp &= \mathcal{A}(\tau)x_\perp + \mathcal{B}(\tau)v, \quad \tau \in [0, T), \\ x_\perp(0) &= \mathcal{L}x_\perp(T), \quad \tau = T, \end{aligned} \tag{5.26}$$

with the time variable, τ, computed $\tau = t$ modulo$T$.

5.1.4 Construction of a Stabilizing Controller

In order find a stabilizing controller for the system (5.26), the LQR method introduced in Section 2.5 will be utilized with a few modifications and simplifications. Firstly, the matrix S, giving weighting the deviation of the states at the final time T, will be set to zero. This is due to the difficulty of determining a suitable final cost for the transient states. Furthermore, the matrix functions $Q(t)$ and $\Gamma(t)$ will be given as constant, positive definite matrices, such that the solution, $R(t)$, to the Riccati differential

$$\dot{R}(t) + \mathcal{A}(t)^T R(t) + R(t)\mathcal{A}(t) + Q = R(t)\mathcal{B}(t)\Gamma^{-1}\mathcal{B}(t)^T R(t), \tag{5.27}$$

if it exists, is positive definite for all $t \in [0, T]$. However, even if such a solution exists, it is not sufficient as to say that one can stabilize the hybrid system (5.26) due to the mapping $x_\perp(0) = \mathcal{L}x_\perp(T)$. An additional condition to assure that the system can be stabilized is addressed next.

Theorem 5.1.1. *Suppose there exists a symmetric, positive definite solution $R(t)$ of* (5.27) *on $[0, T]$ with $Q, \Gamma \succ 0$, which satisfies*

$$\mathcal{L}^T R(0)\mathcal{L} \leq R(T). \tag{5.28}$$

Then the controller

$$v(t) = -\Gamma^{-1}\mathcal{B}(t)^T R(t)x_\perp \tag{5.29}$$

renders the system hybrid linear system (5.26) *exponentially stable.*

Proof. If a solution to (5.27) exists for $t \in [0, T]$, then the pair $\big(\mathcal{A}(t), \mathcal{B}(t)\big)$ is already known to be stabilizable over $[0, T]$. Moreover, introducing the Lyapunov function candidate

$$V\big(x_\perp(t), t\big) = x_\perp^T R(t)x_\perp, \tag{5.30}$$

which, since it is equivalent to the cost-to-go of (2.100), is strictly decreasing on $[0, T]$ as for all $\|x_\perp\| \neq 0$ its derivative w.r.t. time is

$$\dot{V}\big(x_\perp(t), t\big) = x_\perp^T\big[-Q - R(t)\mathcal{B}(t)\Gamma^{-1}\mathcal{B}^T(t)R(t)\big]x_\perp < 0, \quad \forall t \in [0, T]. \tag{5.31}$$

Now, let $x_\perp(t_+^0) \in TS(0)$ be the initial value of the transverse states and let $x_\perp(t_+^1) \in TS(0)$ be the value at the next iteration of the system (5.26), i.e.,

$$x_\perp(t_+^1) = \mathcal{L}x_\perp(t_+^0 + T).$$

It is clear that for the hybrid system (5.26) to be stable over consecutive discrete mapping, the relation $V\big(x_\perp(t_+^1), t_+^1\big) \leq V\big(x_\perp(t_+^0), t_+^0\big)$ must hold, i.e,

$$x_\perp^T(t_+^1)R(0)x_\perp(t_+^1) \leq x_\perp^T(t_+^0)R(0)x_\perp(t_+^0), \tag{5.32}$$

such that the Lyapunov function candidate (5.30) is nonincreasing over the discrete mapping. By noting that $V\big(x_\perp(T), T\big) \leq V\big(x_\perp(0), 0\big)$, it follows that the relation (5.32) is ensured if

$$x_\perp^T(t_+^0+T)R(T)x_\perp(t_+^0+T) \geq x_\perp^T(t_+^1)R(0)x_\perp(t_+^1) = x_\perp^T(t_+^0+T)\mathcal{L}^T R(0)\mathcal{L}x_\perp(t_+^0+T), \tag{5.33}$$

yielding (5.28). Thus, uniform asymptotic stability follows from [57, Th. 3.1], from which exponential stability follows as the controller (5.29) makes $\mathcal{A}(t) - \Gamma^{-1}\mathcal{B}(t)R(t)$ Hurwitz for all $t \in [0, T]$; see Definition 4. □

The relation (5.28) is not new, and was, to the knowledge of the author, first stated in [26] without a formal derivation. Moreover, it should be emphasized that it only provides a sufficient conditions for exponential stability of (5.26), with the necessary condition given by (5.32).

It is not difficult, however, to see that the controller (5.29) does not stabilize the solution $(q_*(t), \dot{q}_*(t))$ of the closed-loop nonlinear system (2.19), (5.6) as the the matrix $\mathcal{B}(t)$ in (5.29) is uncoupled from the states of the system. For this reason, the modified *ad-hoc* controller

$$v(t) = -\Gamma^{-1}\mathcal{B}(\theta, \dot{\theta}, y)^T R(t)x_\perp \tag{5.34}$$

with

$$\mathcal{B}(\theta, \dot{\theta}, y) = \begin{bmatrix} 2\dot{\theta}\alpha(\theta)g_v(\theta, y) \\ \mathbf{0}_2 \\ I_2 \end{bmatrix},$$

exponentially orbitally stabilizes the periodic solution [46]

$$\mathcal{O}_* := \Big\{[q; \dot{q}] \in TQ : \; q_*(t) = \Phi(\theta_*), \; \dot{q}_*(t) = \Phi'(\theta_*)\dot{\theta}_*, \; t \in [0, T]\Big\}.$$

The stabilizing controller, given by (5.6) and (5.34), is still time-variant due to the time dependence of $R(t)$. In order to make the controller fully time-independent, one must introduce some projection operator as (2.65) which retrieves the time-stamp of the moving Poincarè section to which the system states belongs. With $\theta = q_1$ as the motion

generator, which is positive and monotonically decreasing over the step, one can find some function

$$\mathcal{T} : \theta \mapsto \tau, \quad \tau \in [0, T], \tag{5.35}$$

such that the controller can be directly computed just knowing the states of the system:

$$v(q, \dot{q}) = -\Gamma^{-1}\mathcal{B}(\theta, \dot{\theta}, y)^T R\big(\mathcal{T}(\theta)\big)x_\perp(q, \dot{q}). \tag{5.36}$$

The projection operator (5.35) will be dependent on the virtual holonomic constraints and must thus be approximated for each individual gait.

5.1.5 Implementation and Results from Numerical Simulations

The gait chosen to be stabilized in this thesis is the one found using third order trigonometric polynomials as and which is depicted Figure 4.5. Note, however, that all the previously presented gaits with the motion generator as q_1 were successfully stabilized with the method.

The differential Riccati equation (DRE) was solved using the method of (2.112) with a few modifications due to the system being of hybrid nature and that the solution to DRE is not T-periodic. To this end, the approximation of the approximation $\bar{R}(t)$ of $R(t)$ was represented using a Nth-order Beziè r polynomial:

$$\bar{R}(t) = \sum_{k=0}^{N} R_k \frac{N!}{k!(N-k)!}\tau^k(1-\tau)^{N-k}, \tag{5.37}$$

where the $N+1$ symmetric matrices R_j are the optimization variables and $\tau = t/T$. An order of $N = 12$ was used in the presented results as it was found that the performance dropped for lower order, while the increase in performance for higher order was close to negligible.

The LMI was given by

$$\bar{\mathcal{R}}\big(R(t), t\big) = \begin{bmatrix} \dot{R}(t) + \mathcal{A}^T(t)R(t) + R(t)\mathcal{A} + Q + 2\delta R(t) & R(t)\mathcal{B}(t) \\ \mathcal{B}(t)^T R(t) & \Gamma(t) \end{bmatrix}, \tag{5.38}$$

where the additional term $2\delta R(t)$ arises by using the matrix function $\bar{\mathcal{A}}(t) = \mathcal{A}(t) + \delta I_5$ in the problem rather than just $\mathcal{A}(t)$ from (5.26). This makes the system, in a sense, "less stable", resulting in a more aggressive controller the larger the offset parameter δ is chosen. Thus δ acts as an additional tuning parameters in addition to Q and Γ in order to get the desired performance. To easily compare the results and not put to much emphasis on "tuning" the controllers, the matrices Q and Γ were set to the 5×5 and 2×2 identity matrices, respectively, while δ was chosen in order to get a desirable performance.

Lastly, the constraint of (5.28) had to be added to the problem, which, using Beziè r polynomials, is equivalent to

$$\mathcal{L}^T R_0 \mathcal{L} \leq R_N. \tag{5.39}$$

However, it was found that the solvers struggled to find solution using this relation and needed strict bounds on $\bar{R}(t)$, i.e.,

$$-\sigma I_5 \leq \bar{R}(t) \leq \sigma I_5 \tag{5.40}$$

with σ not chosen too large[2]. Restricting the constraint to

$$\mathcal{L}^T R_0 \mathcal{L} = R_N \tag{5.41}$$

greatly increased the likelihood of the solver finding a feasible solution. For this reason, the presented results are given were found with the constraint (5.41) used, which in turn means that the value of the Lyapunov function candidate (5.30) does not change due to the impact.

The procedure used for generating a controller is outlined as follows:

1. Simulate the reduced dynamics (4.14) using a fixed step solver with an integration step of $dt = 1\,\mathrm{ms}$.
2. Set the matrices Q and Γ to some value of choice, as well as the parameter δ.
3. Take 100 evenly spaced time samples from the simulation of the reduced dynamics and formulate the optimization problem (2.112) with $\bar{R}(t)$ from (5.37) and the additional constraints (5.41),(5.40) using the YALMIP toolbox. Note that there seemed to be no increase in performance when increasing the number of time steps.
4. Solve the SDP problem using a solver of choice. For the results presented, the DSDP solver [3] from the OPTI toolbox [7] was used to solve the SDP.

Since the resulting controller is time-dependent, the projection operator (5.35) is needed to make it time-invariant. It was found that time could be well represented in terms of θ by the function

$$\mathcal{T}(\theta) = Ts + a\sin(2\pi s), \quad s = \frac{\theta - \theta_+}{\theta_- - \theta_+}, \tag{5.42}$$

where T is the step period and a is a parameter found using least squares from the measured relation between the actual time t and θ. For the gait from Figure 4.5, a was calculated to -0.1092. The resulting projection along the nominal solution can be seen in Figure 5.1.

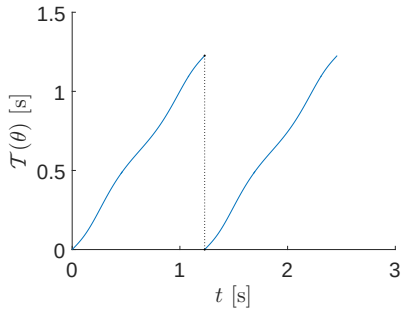
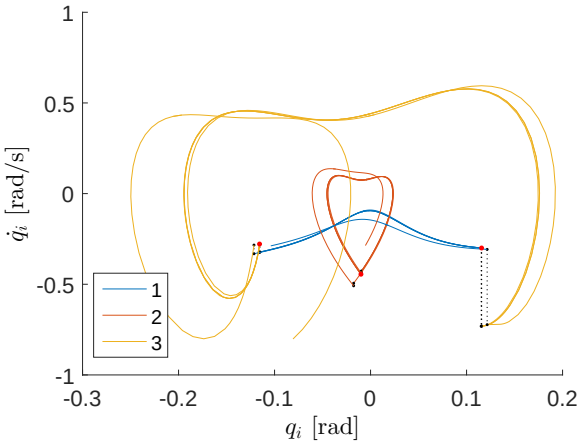


Figure 5.1: Relation between the projection operator and time over two steps.

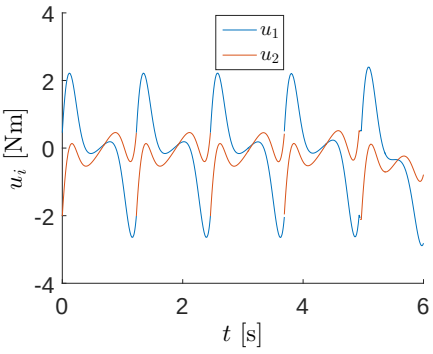
[2]Most solvers tested struggled when $\sigma > 10^4$ was used. With the constraint (5.41), the presented results were found with $\sigma = 10^6$

Open-loop vs stabilizing controller

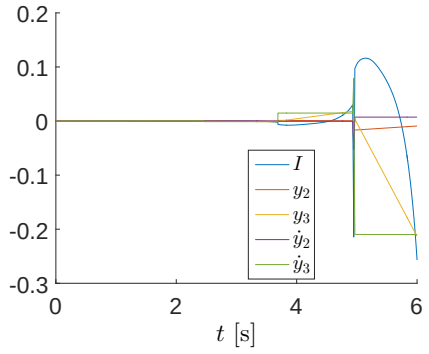
In order to show the necessity of the stabilizing controller (5.34), the biped was simulated with no perturbations in the initial conditions with only the PFL controller (5.6) with $v = [0; 0]$, i.e., open-loop control. The results are shown in Figure 5.2 where the initial states are marked by red dots. Although the open-loop controller manages to make the biped follow the nominal gait over a few steps, small numerical deviations due to the impact quickly builds up making the gait unstable. After the fourth step the biped is already about to fall.



(a) Phase portrait in absolute angles.



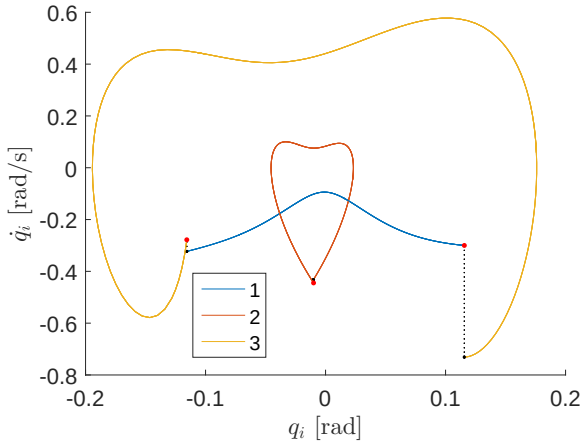
(b) Actuator outputs.



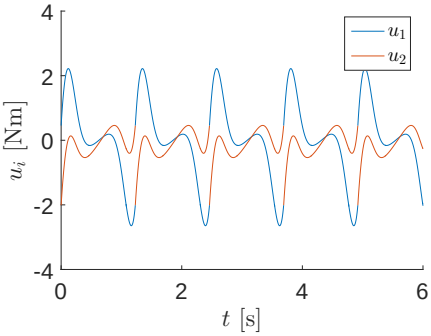
(c) Values of the transverse states .

Figure 5.2: The gait in open loop with only the PFL controller and unperturbed initial conditions.

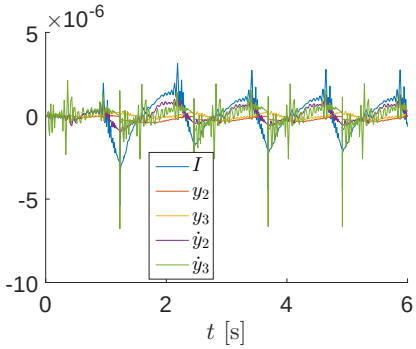
With the stabilizing controller v from (5.6) with $\delta = 1$, the nominal gait was stabilized, as seen in Figure 5.3. Indeed, the values of the transverse states have a maximum deviation from the origin of about 10^{-5}, with only some spikes occurring due to the impacts. Herein, the integral function $I(\cdot)$ was calculated using Algorithm 1 using 100 integration steps, which on average took approximately $1\,\mathrm{ms}$.



(a) Phase portrait in absolute angles.



(b) Actuator outputs.

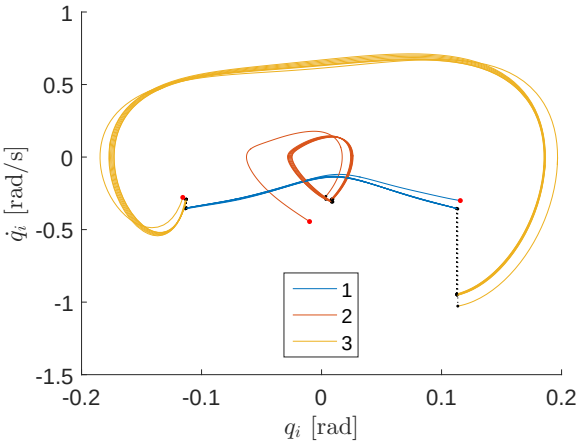


(c) Values of the transverse states .

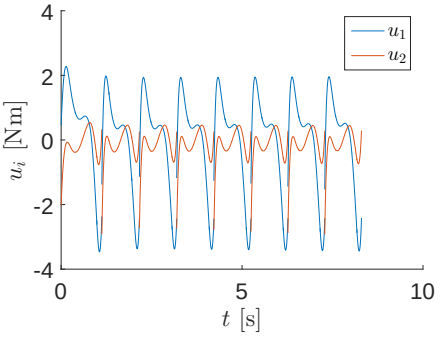
Figure 5.3: The gait with unperturbed initial conditions and the controller generated with $\delta = 1$.

Performance with different system parameters

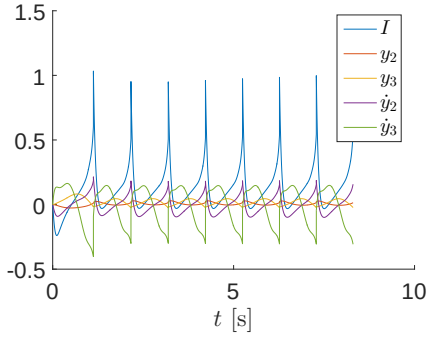
For a real system, it is likely that the parameters used in the construction of the controller will deviate from the real parameters of the system. Because of this, the controller was tested with the system parameters from Table 4.1 scaled up or down by 10%. The results can be seen in Figures 5.4 and 5.5, respectively. When the parameters were scaled up, δ was set to 2 in order to get the desired performance, while it needed to be increased to 3 when the parameters were scaled down in order to stabilize the biped. The controller manages to stabilize the system in both cases. Although, as one would expect, the biped settles to different gaits than the nominal. Figures 5.4c and 5.5c also show that the main deviations of the transverse states happens just prior to and after impact due to the impact map also being changed. Furthermore, Figure 5.5b shows that the reference torque from the controller jumps at impact. Thus, one would need to pass the reference signal through a low-pass filter for implementation on a real system.



(a) Phase portrait in absolute angles.

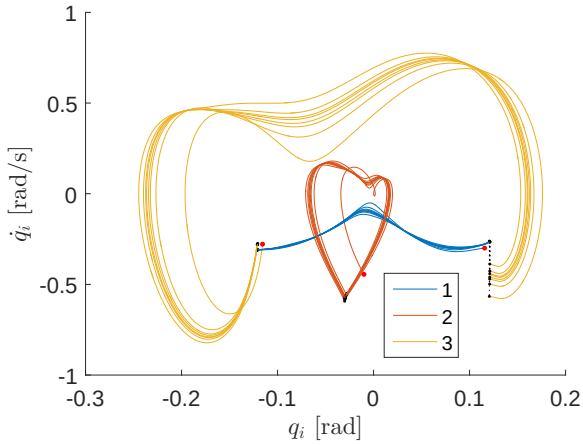


(b) Actuator outputs.

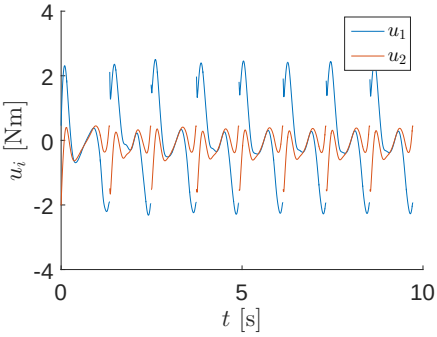


(c) Values of the transverse states .

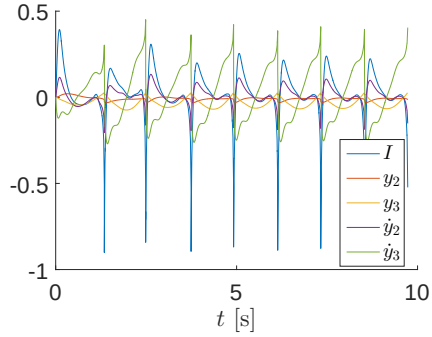
Figure 5.4: The gait with the parameters of the system scaled up by 10% compared to the design phase and a controller with $\delta = 2$.



(a) Phase portrait in absolute angles.



(b) Actuator outputs.



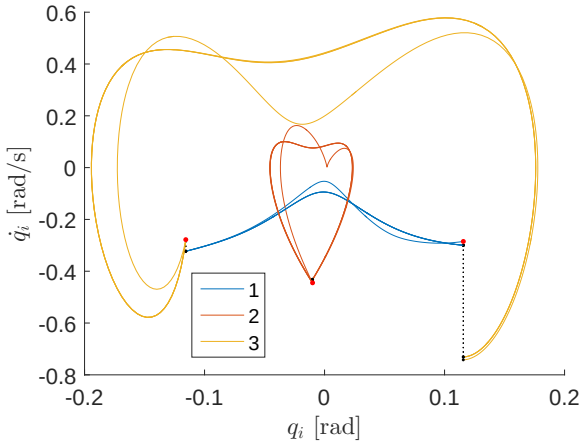
(c) Values of the transverse states .

Figure 5.5: The gait with the parameters of the system scaled down by 10% compared to the design phase and a controller with $\delta = 3$.

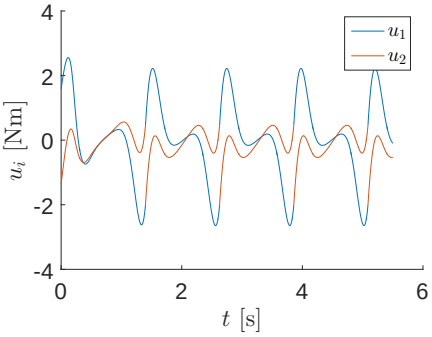
Pertubred initial velocity of the stance leg

Since the stance leg is passive throughout the gait, it is becomes difficult for the controller to correct perturbations from its nominal trajectory. For this reason, the controller was tested when the initial angular velocity of the stance leg was perturbed from its nominal trajectory.

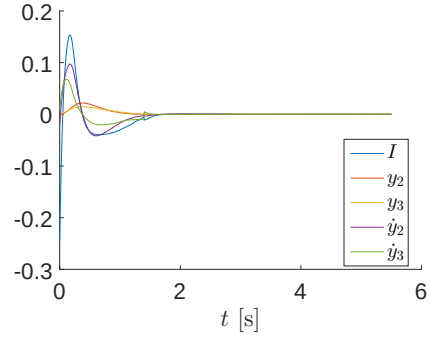
In Figure 5.6 the initial angular velocity was perturbed from the nominal value by $0.015\,\mathrm{rad\,s^{-1}}$ for which δ had to be at least 3. If the perturbation was increased or δ was lowered, the controller was unable to stabilize the gait in time such that the angular velocity of the stance leg became positive, resulting in the biped falling backwards. As seen in Figure 5.6a, the angular velocity of the stance leg, $\dot{q}_1$, almost crosses the horizontal zero-line during the first step but eventually manages to stabilize the gait, with almost perfect tracking for the remaining steps.



(a) Phase portrait in absolute angles.



(b) Actuator outputs.

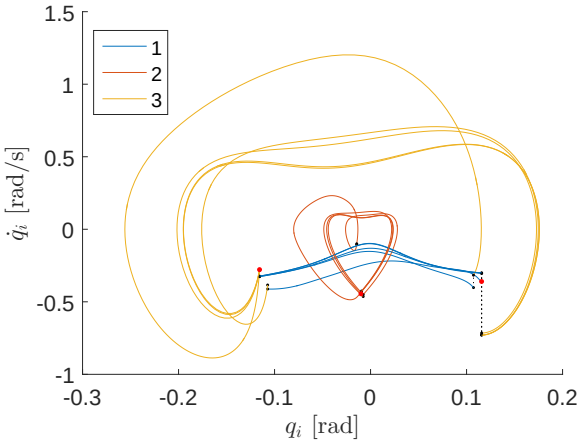


(c) Values of the transverse states .

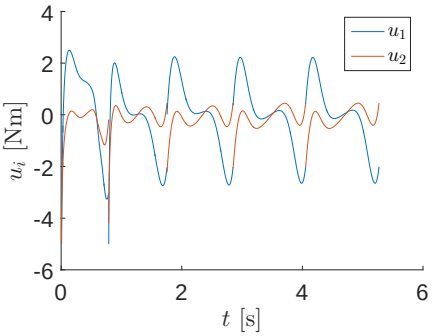
Figure 5.6: The gait with the initial perturbation $\dot{q}_1(t_0) = \dot{\theta}_*(t_0) + 0.015$ and $\delta = 3$.

With negative perturbations, the controller could stabilize higher greater perturbations

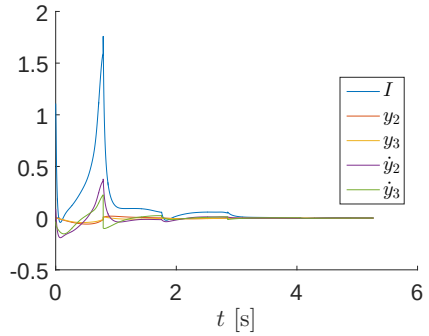
and the δ-parameter could be lowered. However, due to the increase of energy of the system compared to the nominal gait, the controller require high spikes in the torques in order to stabilize the motion. Because of this, saturation of the actuators of $\pm 5\,\mathrm{N\,m}$ was added to keep to realistic torques. In Figure 5.7 the initial perturbation is given by $\dot{q}_1(t_0) = \dot{\theta}_*(t_0) - 0.06$ with $\delta = 2$. As seen by Figure 5.7b, the actuators saturate at both the beginning and the end of the step, but still manages to stabilize the motion even tough, as seen in Figure 5.7c, the absolute value of the transverse coordinates are actually bigger in the beginning of the second step than the first. Indeed, the motion does not converge to the nominal before the third step.



(a) Phase portrait in absolute angles.



(b) Actuator outputs.

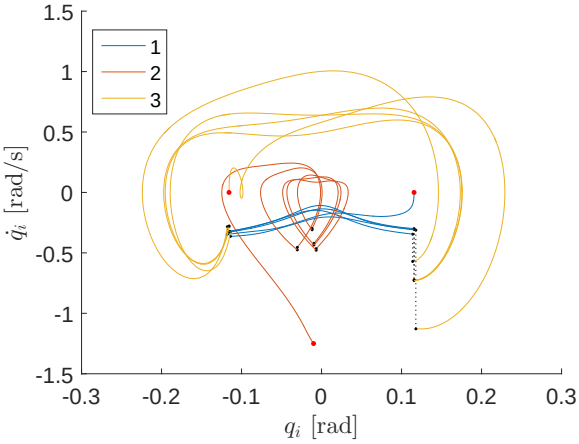


(c) Values of the transverse states .

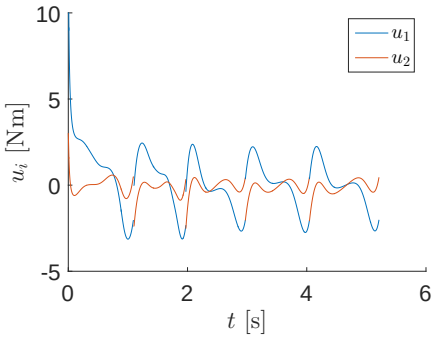
Figure 5.7: The gait with the initial perturbation $\dot{q}_1(t_0) = \dot{\theta}_*(t_0) - 0.06$ of the stance leg and controller with $\delta = 2$ with enabled saturation of the actuators at $\pm 5\,\mathrm{N\,m}$.

Zero initial angular velocity of the stance and swing leg

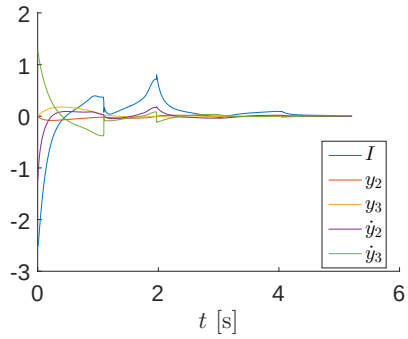
An important aspect of walking is how to initialize a gait from a stand still. To this end, the biped was initialized with zero angular velocity of the stance and the swing legs, with only initial angular velocity of the torso to emulate the the biped being pushed into starting. In Figure 5.8c, the initial angular velocity of the torso was set to $-1.25\,\mathrm{rad\,s^{-1}}$ with δ set to 1. Increasing δ led to very high torque requirements at the beginning of the motion. Furthermore, if the saturation of the actuators was lowered bellow $\pm 10\,\mathrm{N\,m}$ the controller was unable to stabilize the motion.



(a) Phase portrait in absolute angles.



(b) Actuator outputs.



(c) Values of the transverse states .

Figure 5.8: The gait with the initial conditions perturbed such that, measured in absolute angles, the swing and stance leg have zero velocity, while the torso is initialized with $\dot{q}_2^a(t_0) = -1.25\,\mathrm{rad\,s^{-1}}$. The controller was generated with $\delta = 1$ and saturation was set to $\pm 10\,\mathrm{N\,m}$.

5.2 Motion Generator found through a Projection from the Configuration

In this section, a controller will be constructed for the case where the scalar MG is not directly known as a function of the configuration, but can be closely approximated using a projection from the generalized coordinates. Such a method was first presented in [37] for a fully actuated industrial manipulator. Therefore, this method will here be transferred to an underactuated system with hybrid dynamics.

5.2.1 Excessive Transverse Coordinates

Suppose one has found a solution of the Euler-Lagrange (2.8) described by the geometric relations

$$q_*(t) = \Phi(\theta_*), \tag{5.43a}$$

$$\dot{q}_*(t) = \Phi^{'}(\theta_*)\dot{\theta}_*, \tag{5.43b}$$

where θ is a strictly monotonic MG which can not directly be found from the configuration variables of the system, but which can be approximated by the projection

$$\theta \approx P(q), \tag{5.44}$$

such that

$$\dot{\theta} \approx \frac{\partial P(q)}{\partial q}\dot{q}, \tag{5.45}$$

$$\ddot{\theta} \approx \frac{\partial P(q)}{\partial q}\ddot{q} + \dot{q}^T\left[\frac{\partial^2 P(q)}{\partial q^2}\right]\dot{q}. \tag{5.46}$$

Since the projection is only an approximation of the MG, using (5.45) to retrieve the velocity might lead to large inaccuracies compared to what the velocity of the MG should truly be for a given θ, especially for perturbation from the nominal trajectory. For this reason, one can introduce a function $h(\theta)$ which retrieves the correct velocity for a given value of the MG, i.e.,

$$\dot{\theta}_*(t) = h\big(\theta_*(t)\big). \tag{5.47}$$

For the biped, this function can be found from the integral function (4.51) as

$$h(\theta) = -\frac{1}{\alpha(\theta)}\left(\Big(\alpha(\theta_0)\dot{\theta}_0\Big)^2 - 2\int_{\theta_0}^{\theta}\alpha(\tau)\gamma(\tau)d\tau\right)^{1/2}. \tag{5.48}$$

Noting from (4.17) that $\ddot{\theta} = \frac{1}{2}\frac{\partial}{\partial\theta}\big(h(\theta)\big)^2$, its partial derivatives are found to be

$$h^{'}(\theta) := \frac{\partial h(\theta)}{\partial\theta} = -\frac{1}{\alpha(\theta)h(\theta)}\left[\gamma(\theta) + \beta(\theta)h(\theta)^2\right], \tag{5.49}$$

$$h^{''}(\theta) := \frac{\partial^2 h(\theta)}{\partial\theta^2} = -\frac{1}{\alpha(\theta)h(\theta)}\left[\gamma^{'}(\theta) + \beta^{'}(\theta)h(\theta)^2 + 3\beta(\theta)h(\theta)h^{'}(\theta) + \alpha(\theta)h^{'}(\theta)^2\right] \tag{5.50}$$

where

$$\gamma^{'}(\theta) := \frac{\partial \gamma(\theta)}{\partial \theta} \quad \text{and} \quad \beta^{'}(\theta) := \frac{\partial \beta(\theta)}{\partial \theta}.$$

Using the function $h(\theta)$, one can introduce the output functions

$$y = q - \Phi(\theta), \tag{5.51a}$$

$$\omega = \dot{q} - \Phi^{'}(\theta)h(\theta), \tag{5.51b}$$

acting as the new transverse coordinates, and whose time derivatives are

$$\dot{y} = \dot{q} - \Phi^{'}(\theta)\dot{\theta}, \tag{5.52a}$$

$$\dot{\omega} = \ddot{q} - \mathcal{F}(\theta)\dot{\theta}, \tag{5.52b}$$

where column matrix function $\mathcal{F}(\theta)$ is defined as

$$\mathcal{F}(\theta) := [\mathcal{F}_1(\theta); \mathcal{F}_2(\theta); \mathcal{F}_3(\theta)] = \Phi^{''}(\theta)h(\theta) + \Phi^{'}(\theta)h^{'}(\theta). \tag{5.53}$$

Thus, one has $2n$ transverse coordinates. However, for any state space of dimension $2n$, there can only be $(2n-1)$ independent transverse coordinates [37]. This follows directly from the fact that any Poincarè surface, which is transverse to the trajectory by Definition 2, is of $(2n-1)$-dimensions. Thus, the coordinates of (5.51) are excessive transverse coordinates. Nonetheless, asymptotic stability of the linearized transverse dynamics of the form

$$\frac{d}{dt}\begin{bmatrix}\delta y \\ \delta \omega\end{bmatrix} = A(t)\begin{bmatrix}\delta y \\ \delta \omega\end{bmatrix} \tag{5.54}$$

about the trajectory (5.43) ensures local asymptotic stability of the trajectory [37].

5.2.2 Transverse Linearization

In order to find the linearized transverse dynamics of (5.51), one can note that (5.52) can be rewritten as

$$\dot{y} = f_1(q, \dot{q}) := \dot{q} - \Phi^{'}(\theta)\frac{\partial P(q)}{\partial q}\dot{q}, \tag{5.55}$$

$$\dot{\omega} = f_2(q, \dot{q}) := \ddot{q} - \mathcal{F}(\theta)\frac{\partial P(q)}{\partial q}\dot{q}, \tag{5.56}$$

such that $f_1(q, \dot{q})$ and $f_2(q, \dot{q})$ equals zero on the nominal trajectory (5.43). Thus, for small deviations from the nominal trajectory, a linearization of the evolution of these $2n$ transverse coordinates can be written on the form

$$\frac{d}{dt}\begin{bmatrix}\delta y \\ \delta \omega\end{bmatrix} = \mathcal{A}(t)\begin{bmatrix}\delta y \\ \delta \omega\end{bmatrix} + \mathcal{B}(t)v, \tag{5.57}$$

where $\mathcal{A}(t)$ and $\mathcal{B}(t)$ are found from

$$\mathcal{A}(t) := \left.\begin{bmatrix}\frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial \omega} \\ \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial \omega}\end{bmatrix}\right|_{\substack{y=0 \\ \omega=0}} \quad \text{and} \quad \mathcal{B}(t) := \left.\begin{bmatrix}\frac{\partial f_1}{\partial v} \\ \frac{\partial f_2}{\partial v}\end{bmatrix}\right|_{\substack{y=0 \\ \omega=0}}, \tag{5.58}$$

with $v := v(y, \omega)$ denoting a control input which satisfies $v\big(0, 0\big) = [0; 0]$.

However, computing the Jacobian of the functions $f_1(\cdot)$ and $f_2(\cdot)$ in terms of the transverse coordinates is not straightforward. Nevertheless, it can be done by noting that for each of the functions, the following relations between the expansions must hold

$$\delta f_i - \frac{\partial f_i}{\partial v}\delta v = \frac{\partial f_i}{\partial y}\delta y + \frac{\partial f_i}{\partial \omega}\delta\omega = \frac{\partial f_i}{\partial q}\delta q + \frac{\partial f_i}{\partial \dot{q}}\delta\dot{q}, \quad i \in \{1, 2\}. \tag{5.59}$$

Thus, one can write

$$\mathcal{A}(t)\begin{bmatrix}\delta y \\ \delta\omega\end{bmatrix} = C(t)\begin{bmatrix}\delta q \\ \delta\dot{q}\end{bmatrix} \tag{5.60}$$

with

$$C(t) := \begin{bmatrix}\frac{\partial f_1}{\partial q} & \frac{\partial f_1}{\partial \dot{q}} \\ \frac{\partial f_2}{\partial q} & \frac{\partial f_2}{\partial \dot{q}}\end{bmatrix}\Bigg|_{\substack{\theta=\theta_*(t) \\ q=q_*(t) \\ \dot{q}=\dot{q}_*(t)}}. \tag{5.61}$$

Further noting that small increments of the transverse states about the nominal trajectory can be written in terms of small increments of the states as

$$\begin{bmatrix}\delta y \\ \delta\omega\end{bmatrix} = \begin{bmatrix}I_n - \Phi^{'}(\theta)\frac{\partial P(q)}{\partial q} & \mathbf{0}_n \\ -\mathcal{F}(\theta)\frac{\partial P(q)}{\partial q} & I_n\end{bmatrix}\Bigg|_{\substack{\theta=\theta_*(t) \\ q=q_*(t)}}\begin{bmatrix}\delta q \\ \delta\dot{q}\end{bmatrix} = T(t)\begin{bmatrix}\delta q \\ \delta\dot{q}\end{bmatrix}, \tag{5.62}$$

one obtains the relation

$$\mathcal{A}(t)T(t) = C(t). \tag{5.63}$$

Unfortunately, the matrix function $T(t)$ can not be directly inverted as its rank is not necessarily[3] $2n$ due to the fact that there are only $(2n-1)$ independent transverse coordinates; yet the solvability of (5.63) is guaranteed due to the existence of the expansions (5.59) [37]. Moreover, one can observe that from (5.63) that

$$\frac{\partial f_1}{\partial \omega} = \frac{\partial f_1}{\partial \dot{q}} \quad \text{and} \quad \frac{\partial f_2}{\partial \omega} = \frac{\partial f_2}{\partial \dot{q}},$$

such that one only needs to solve the equations

$$\frac{\partial f_i}{\partial y}\left(I_n - \Phi^{'}(\theta)\frac{\partial P(q)}{\partial q}\right) - \frac{\partial f_i}{\partial \dot{q}}\mathcal{F}(\theta)\frac{\partial P(q)}{\partial q} = \frac{\partial f_i}{\partial q}, \quad i = 1, 2, \tag{5.64}$$

where the term $I_n - \Phi^{'}(\theta)\big(\partial P(q)/\partial q\big)$ can not always be inverted since it has a rank of $(n-1)$ when the rank of $T(t)$ is $(2n-1)$. This also suggest that the equation (5.64) might have multiple solutions when $T(t)$ has insufficient rank. In order to solve it, one can use methods like, e.g., taking the pseudoinverse, or by minimizing the least-square error.

[3] If the projection $P(q)$ exactly represent θ, then $T(t)$ has rank $(2n-1)$ and is thus singular. This is not necessarily the case if $P(q)$ is a non-exact approximation.

The remaining step is therefore finding the expansions of the functions $f_i(\cdot)$ in terms of the states and the control input v. Herein, the expansion of $f_1(\cdot)$ is trivial, and is given by

$$\frac{\partial}{\partial q} f_1(q,\dot{q}) = -\Big[\Phi^{''}(\theta)\frac{\partial P(q)}{\partial q}\Big]\frac{\partial P(q)}{\partial q}\dot{q} - \Phi^{'}(\theta)\Big[\frac{\partial^2 P(q)}{\partial q^2}\dot{q}\Big]^T, \tag{5.65a}$$

$$\frac{\partial}{\partial \dot{q}} f_1(q,\dot{q}) = I_n - \Phi^{'}(\theta)\frac{\partial P(q)}{\partial q}, \tag{5.65b}$$

$$\frac{\partial}{\partial v} f_1(q,\dot{q}) = \mathbf{0}_{3\times 2}. \tag{5.65c}$$

The expansion of $f_2(q,\dot{q},v)$ is, however, not trivial due to the term

$$\ddot{q} = M(q)^{-1}\Big[Bu - C(q,\dot{q})\dot{q} - G(q)\Big].$$

Herein, the control input u should be of the form (2.55). This is achieved with the semi-partial feedback linearizing controller

$$u = K(q)^{-1}\big(v + B^T\mathcal{F}(\theta)h(\theta) - R(q,\dot{q})\big), \tag{5.66}$$

where

$$K(q) := B^T M(q)^{-1} B, \tag{5.67}$$

$$R(q,\dot{q}) := B^T M(q)^{-1}\Big(- C(q,\dot{q})\dot{q} - G(q)\Big), \tag{5.68}$$

in which the invertibility of $K(q)$ is guaranteed as $M(q)$ is positive definite and the columns of B are non-zero. Thus, by writing $f_2(q,\dot{q},v) = [f_2^1; f_2^2; f_2^3]$, the expansions of the second and third functions are given by

$$\begin{aligned}\Big[\frac{\partial f_2^2}{\partial q};\ \frac{\partial f_2^3}{\partial q}\Big] =& B^T\mathcal{F}(\theta)\Big(h^{'}(\theta)\frac{\partial P(q)}{\partial q} - \dot{q}^T\frac{\partial^2 P(q)}{\partial q^2}\Big) \\ &+ B^T\mathcal{F}^{'}(\theta)\frac{\partial P(q)}{\partial q}\Big(h(\theta) - \frac{\partial P(q)}{\partial q}\dot{q}\Big),\end{aligned} \tag{5.69a}$$

$$\Big[\frac{\partial f_2^2}{\partial \dot{q}};\ \frac{\partial f_2^3}{\partial \dot{q}}\Big] = - B^T\mathcal{F}(\theta)\frac{\partial P(q)}{\partial q}, \tag{5.69b}$$

$$\Big[\frac{\partial f_2^2}{\partial v};\ \frac{\partial f_2^3}{\partial v}\Big] = I_2 \tag{5.69c}$$

In order to find the remaining expansion of f_2^1, one can start by writing

$$B^{\perp}\Big\{M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q)\Big\} = 0$$

and note that the controller (5.66) brings $\ddot{q}_{2,3} = [\ddot{q}_2;\ \ddot{q}_3]$ on the form

$$\ddot{q}_{2,3} = B^T\mathcal{F}(\theta)h(\theta) + v.$$

Hence,

$$f_2^1(q,\dot{q},v) = \frac{1}{m_{11}(q)}\Big\{ - z(q,\dot{q}) - M_1(q)Bv - C_1(q,\dot{q})\dot{q} - g_1(q)\Big\}, \tag{5.70}$$
$$z(q,\dot{q}) := m_{11}(q)\mathcal{F}_1(\theta)\frac{\partial P(q)}{\partial q}\dot{q} + m_{12}(q)\mathcal{F}_2(\theta)h(\theta) + m_{13}(q)\mathcal{F}_3(\theta)h(\theta),$$

where $M_1(q)$ and $C_1(q,\dot{q})$ denote the first rows of the matrix functions $M(q)$ and $C(q,\dot{q})$; $g_1(q)$ is the first element of the gravity vector; and $m_{ij}(q)$ denotes the i,j-th element of the inertia matrix. From this, it follows that

$$\frac{\partial f_2^1}{\partial v} = -\frac{1}{m_{11}(q)}[m_{12}(q)\ m_{13}(q)], \tag{5.71}$$

such that the matrix function $\mathcal{B}(t)$ has the form

$$\mathcal{B}(t) = \begin{bmatrix} \mathbf{0}_{3\times 2} \\ \frac{\partial f_2^1}{\partial v} \\ I_2 \end{bmatrix}\Bigg|_{q=q_*(t)}. \tag{5.72}$$

Since the actuator inputs v are linear in f_2^1, one can set $v = [0;0]$ for the remaining computations and denote $\hat{f}_2^1 := f_2^1(q,\dot{q},\mathbf{0})$. Moreover, by defining

$$\eta(q,\dot{q}) := -z(q,\dot{q}) - C_1(q,\dot{q})\dot{q} - g_1(q), \tag{5.73}$$

one can compute the Jacobian by the following relations:

$$\frac{\partial \hat{f}_2^1}{\partial q} = \frac{1}{m_{11}(q)}\left(\frac{\partial \eta(q,\dot{q})}{\partial q} - \frac{\partial m_{11}(q)}{\partial q}\hat{f}_2^1\right), \tag{5.74}$$
$$\frac{\partial \hat{f}_2^1}{\partial \dot{q}} = \frac{1}{m_{11}(q)}\frac{\partial \eta(q,\dot{q})}{\partial \dot{q}}. \tag{5.75}$$

Herein, the expansions of $\eta(q,\dot{q})$ can be computed by the formulas

$$\frac{\partial \eta(q,\dot{q})}{\partial q} = -\frac{\partial z(q,\dot{q})}{\partial q} - \dot{q}^T\frac{\partial C_1(q,\dot{q})}{\partial q} - \frac{\partial g_1(q)}{\partial q} \tag{5.76}$$

and

$$\frac{\partial \eta(q,\dot{q})}{\partial \dot{q}} = -\frac{\partial z(q,\dot{q})}{\partial \dot{q}} - \dot{q}^T\frac{\partial C_1(q,\dot{q})}{\partial \dot{q}} - C_1(q,\dot{q}). \tag{5.77}$$

By close inspection, one can see that

$$\eta\big(q_*(t),\dot{q}_*(t)\big) = -\alpha(\theta_*)\ddot{\theta}_* - \beta(\theta_*)\dot{\theta}_*^2 - \gamma(\theta_*) = 0$$

if

$$h(\theta_*) = \frac{\partial P(q_*)}{\partial q}\dot{q}_*.$$

Hence, this can be used to check the accuracy of the projection $P(q)$. Furthermore, if $\eta(\cdot)$ is exactly zero on the trajectory, then on can set $\hat{f}_2^1 = 0$ in (5.74) as it will disappear in the linearization, which is the case for perfect projections of the MG.

Thus, by solving (5.60) to find $\mathcal{A}(t)$, the linearized transverse dynamics takes the form (5.57), with the addition of an instantaneous mapping $\mathcal{L}(t)$, which is found in by the same manner as (5.25) by using the relation (5.62) and its inverse, found from, e.g., using the pseudoinverse. A stabilizing controller, if it exists, can then be constructed using the same method as in Section 5.1.4.

5.2.3 Implementation and Results from Numerical Simulations

In this section, results from numerical simulations using the aforementioned method of constructing a controller will be presented. The stabilization of the gait depicted in Figure 4.11 will be shown here, while some results of stabilizing the gait in Figure 4.5, as was done with the previous control method and which represent the perfect projection $P(q) = q_1$, is given in Appendix C.2.

For the gait shown in Figure 4.11, one needed to find a projection $P(q) \in [0, 1]$ with $P(q_+) = 0$ and $P(q_-) = 1$. This was done by noticing that by defining $\hat{q} := q_1 + 5.5q_2 + 2q_3$, then $\hat{q}$ was monotonic over the trajectory and could be used to give reasonable representation of the MG by

$$\hat{s} := \frac{\hat{q}(t) - \hat{q}(0)}{\hat{q}(T) - \hat{q}(0)} \approx \theta(t).$$

However, this projection was still not very accurate, such that a better projection was clearly needed. To this end, a 10th order Bezièr polynomial which was a function of the variable $\hat{s}$ was used. Thus, the gradient and the Hessian of the projection was found by

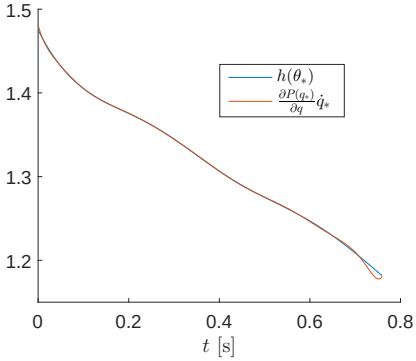
$$\frac{\partial P(q)}{\partial q} = \frac{\partial P(q)}{\partial \hat{s}} \frac{\partial \hat{s}}{\partial q} \quad \text{and} \quad \frac{\partial^2 P(q)}{\partial q^2} = \frac{\partial^2 P(q)}{\partial \hat{s}^2} \Big(\frac{\partial \hat{s}}{\partial q}\Big)^T \frac{\partial \hat{s}}{\partial q},$$

where

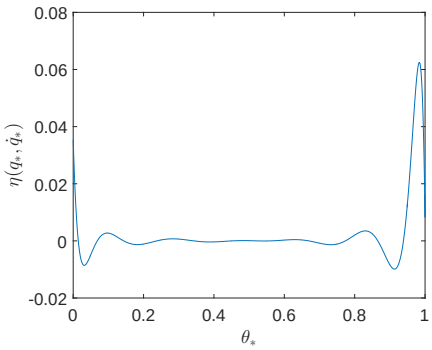
$$\frac{\partial \hat{s}}{\partial q} = [1,\ 5.5,\ 2] / \big(\hat{q}(T) - \hat{q}(0)\big).$$

The parameters for the Bezièr polynomial were found by minimizing the least square error of $\|\theta_* - P(q_*)\|$ over a given number of time steps. Over 1500 time steps, the resulting error was approximately $1 \cdot 10^{-2}$, with the error in the velocities is shown in Figure 5.9a. Although the the projection is very close, the deviation at the beginning and the end of the trajectory is very prominent as seen in Figure 5.9b, in which $\eta(q, \dot{q})$, defined in (5.73), should be zero on the whole trajectory for a perfect projection. This of course shows that a better projection would be advantageous. However, to illustrate the method with an imperfect projection, this projection will be used for the results presented.

In order to solve the equations (5.64), the pseudoinverse function provided by MATLAB was used. This was also used to find the inverse of the relation $[\delta y; \delta\omega] = T(t)[\delta q; \delta\dot{q}]$ in order to find the impact map $\mathcal{L}$ for the linearized transverse dynamics by the same method as in Section 5.1.3. Note that there might exist better methods than the pseudoinverse for solving these equations.



(a) Accuracy of the velocity.



(b) $\eta(q, \dot{q})$ against θ.

Figure 5.9: Accuracy of the projection $P(q)$ over the nominal trajectory.

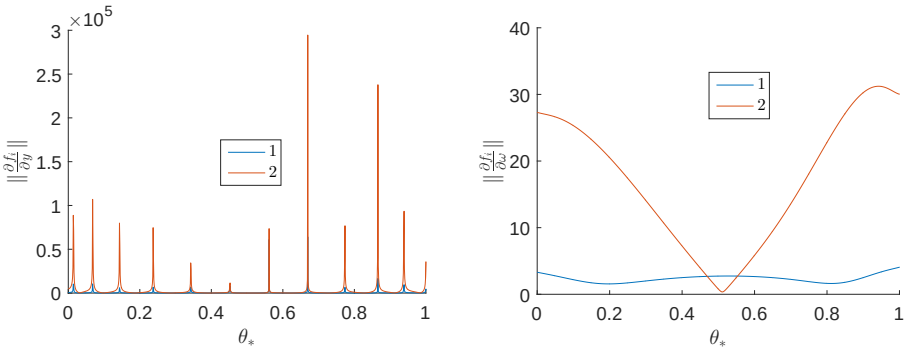
The controller was generated using the same method as described in Section5.1.5 with a Bezièr polynomial of order 12 to approximate $R(t)$. Herein, the constraint of the velocity mapping had to be given as

$$\mathcal{L}^T R_0 \mathcal{L} \leq R_N$$

in order for a solution to be found rather than the equality constraint used for the other controller that was developed. Moreover, the linearization showed that the expansions of the function f_i in terms of y resulted in much larger values of the matrix elements than that of the expansion in terms of ω. Indeed, as seen in Figure 5.10, the norm of the expansions in terms of y "blow up" several times over the step [4]. Because of this, the solver was unable to find a stabilizing solution when setting the weighting matrix Q to the 6×6 identity matrix. However, by increasing the values of the bottom right diagonal, i.e., weighting deviations of ω more than y, by using e.g., $Q = \text{diag}(1, 1, 1, 100, 100, 100)$, the solver was able to find solution. Thus, such weighting will be used in the results presented, where a number 100 time samples in the SDP have been used.

In order to make this controller time-invariant, one can note that time can be well approximated by $\theta \cdot T$, where T is the period of the step, as seen in Figure 5.11. Of course, this is far from a perfect projection, such that performance can possibly be improved by developing a better projection by, e.g, some order Bezièr polynomial in terms of θ. Yet, this simple projection operator gave satisfactory results such that the development of a better one was not attempted.

[4] Solving the equations (5.64) using the pseudoinverse or by minimizing the least squares error both resulted in the same "blow ups" of the norms and gave the same solutions. However, it shows that other methods for solving these equations might be advantageous as they can have multiple solutions when the rank of $T(t)$ is $2n - 1$.



(a) Norm of $\frac{\partial f_i}{\partial y}$.

(b) Norm of $\frac{\partial f_i}{\partial \omega}$.

Figure 5.10: Norms of the expansions of f_i over the nominal trajectory.

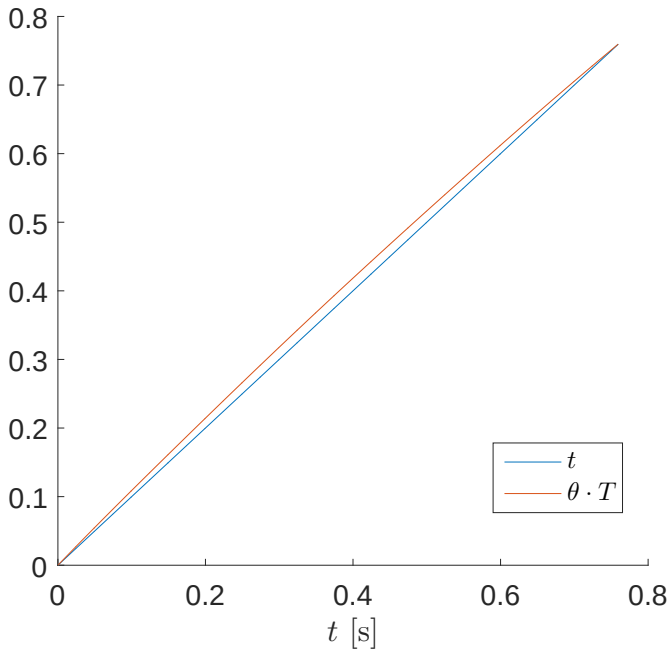
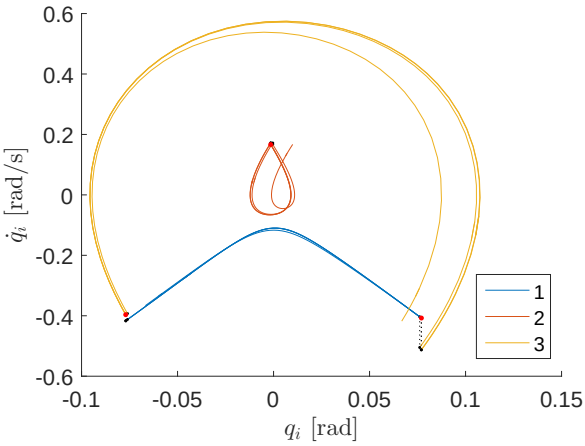


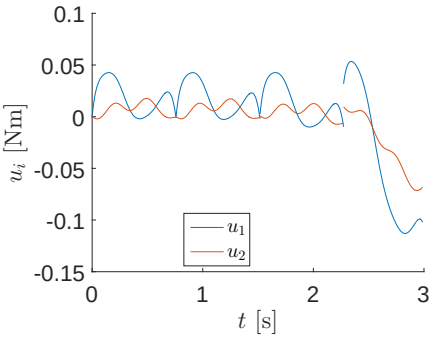
Figure 5.11: Time vs $\theta \cdot T$ over the step.

Open-loop vs stabilizing controller

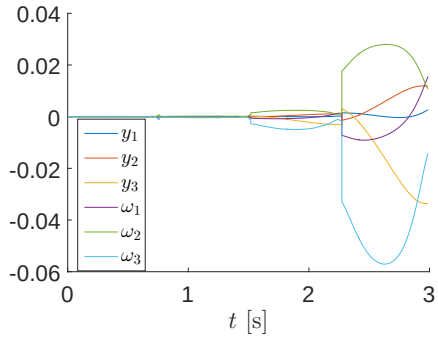
Setting $v = [0; 0]$ in the controller (5.66), i.e., in open-loop with zero perturbations, the resulting walking motion can be seen in Figure 5.12. This shows that the nominal controller is able to recreate the gait, but it quickly becomes unstable after just a few steps.



(a) Phase portrait in absolute angles.



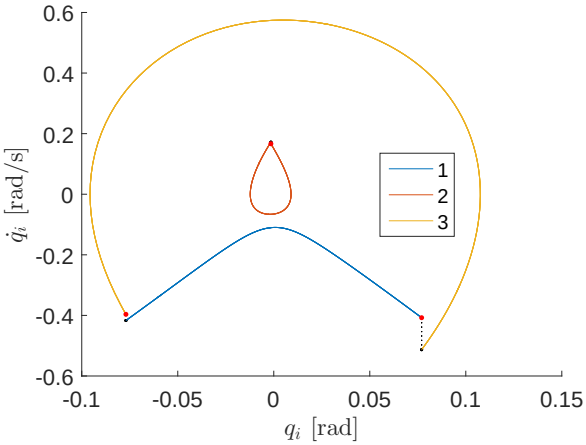
(b) Actuator outputs.



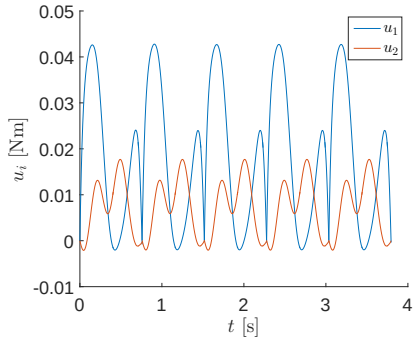
(c) Values of the transverse states .

Figure 5.12: The gait in open loop with only the semi-PFL controller and unperturbed initial conditions.

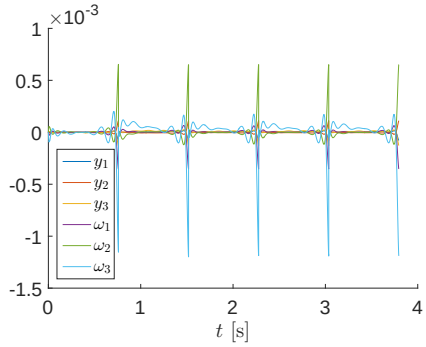
By generating a controller with $Q = \text{diag}(1, 1, 1, 100, 100, 100)$, $\Gamma = I_2$ and $\delta = 0$, the controller managed to stabilize the gait when there were no perturbations present, as seen in Figure 5.13. Indeed, the gait is stabilized even though the value of $\|\omega\|$ greatly increases just prior to the impact, as seen from Figure 5.13c, due to the imperfect projection $P(q)$.



(a) Phase portrait in absolute angles.



(b) Actuator outputs.



(c) Values of the transverse states .

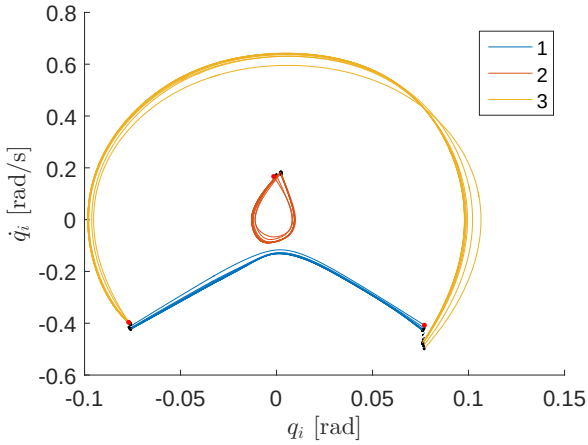
Figure 5.13: The gait in closed loop and unperturbed initial conditions.

Performance with different system parameters

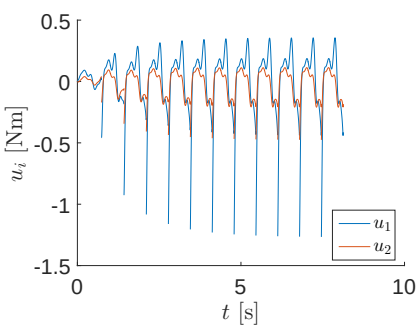
As with the controller generated using the angle of the stance leg as the MG, it is interesting to see how the controller can handle different system parameters from those used when making the controller. In order to test this, a controller was made with $Q = \text{diag}(1, 1, 1, 1000, 1000, 1000)$, $\Gamma = 10 \cdot I_2$ and $\delta = 5$, and the system parameters were increased and decreased to find the maximum deviation which the controller managed to stabilize the motion.

The controller managed to stabilize an increase in the system parameters of a maximum of 3%. The resulting motion can be seen in Figure 5.14, where the biped settles into a different gait after approximately 7 steps. Since the the offset parameter δ is quite large, the resulting control action is quite aggressive, resulting in large spikes around the time of impact as seen in Figure 5.14b,

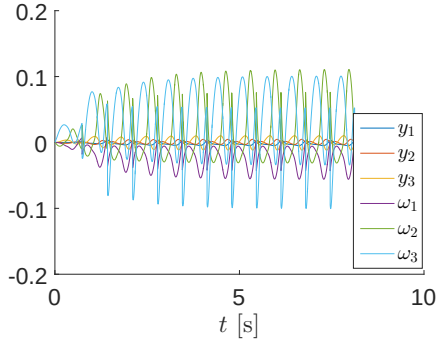
When decreasing the parameters, the controller managed a maximum decrease of 8%,



(a) Phase portrait in absolute angles.



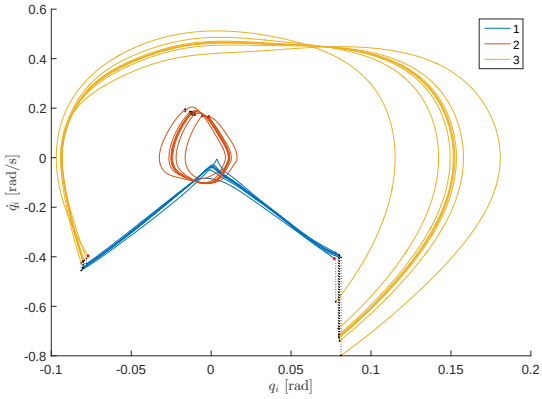
(b) Actuator outputs.



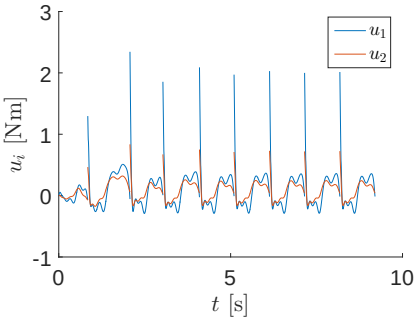
(c) Values of the transverse states .

Figure 5.14: The gait with the parameters of the system scaled up by 6% compared to the design phase .

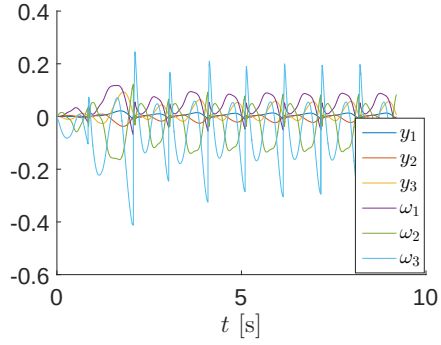
with the resulting evolution shown in Figure 5.15. Now, the biped settles into a new gait after about 4 steps. However, the aggressive control action also results in large spikes in the applied torques at impact, as seen in Figure 5.15b.



(a) Phase portrait in absolute angles.



(b) Actuator outputs.



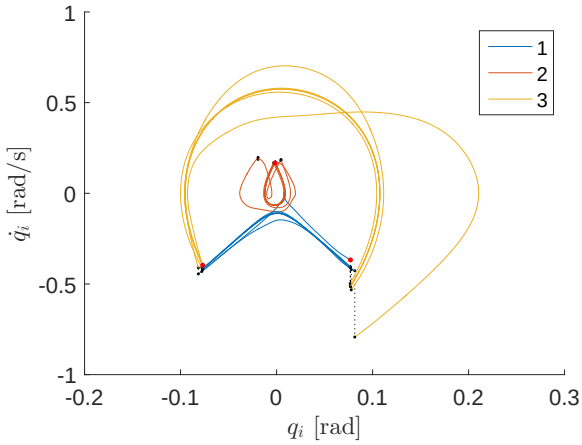
(c) Values of the transverse states .

Figure 5.15: The gait with the parameters of the system scaled down by 8% compared to the design phase

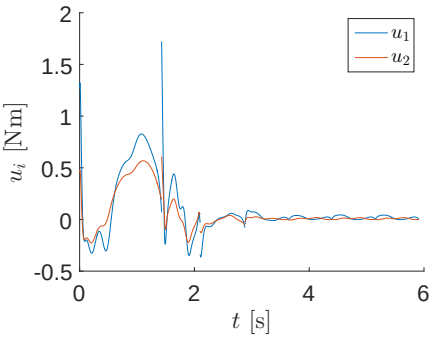
Perturbed initial velocity of the stance leg

As it was with the previous controller, the new control method was tested when the velocity of the angle of the stance leg was perturbed, as this is a passive degree of freedom. The controller was generated with the same weighting matrices as the previous subsection.

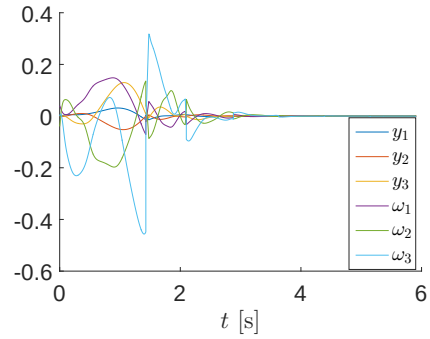
With a initial perturbation of $0.04\,\mathrm{rad\,s^{-1}}$, the controller managed to stabilize and settle into the nominal motion after approximately 3 steps, as seen in Figure 5.16. Increasing the perturbation made the robot fall backwards.



(a) Phase portrait in absolute angles.



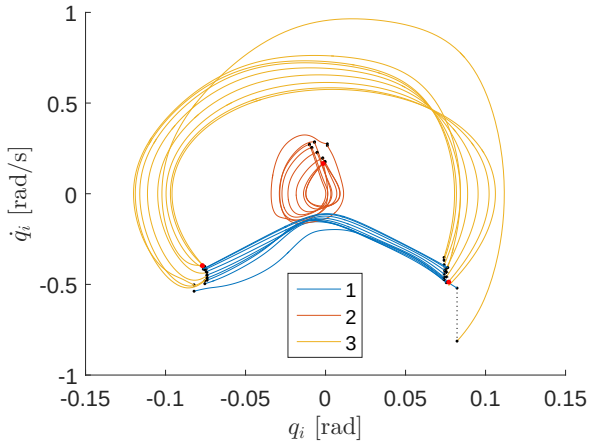
(b) Actuator outputs.



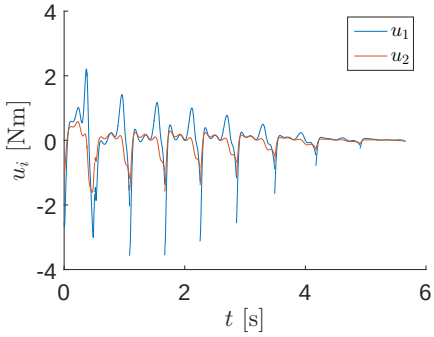
(c) Values of the transverse states .

Figure 5.16: The gait with the initial perturbation $\dot{q}_1(t_0) = \dot{\theta}_*(t_0) + 0.04$.

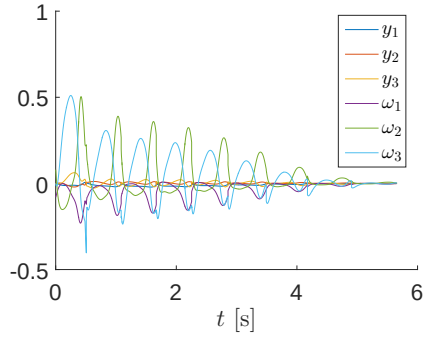
For a negative perturbation, the controller managed to stabilize the motion for a maximum perturbation of $-0.08\,\mathrm{rad\,s^{-1}}$. It took approximately 8 steps before the controller managed to settle the biped into the nominal gait, as seen in Figure 5.17.



(a) Phase portrait in absolute angles.



(b) Actuator outputs.

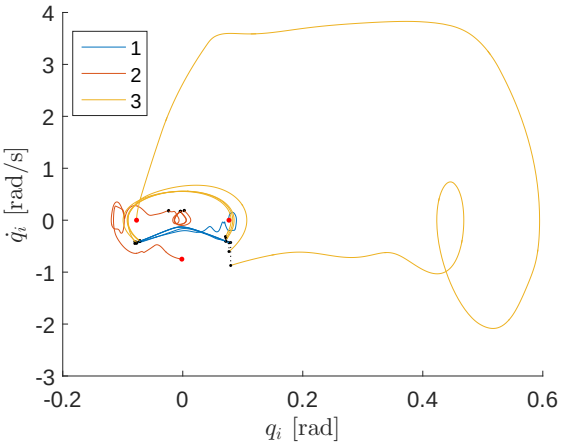


(c) Values of the transverse states .

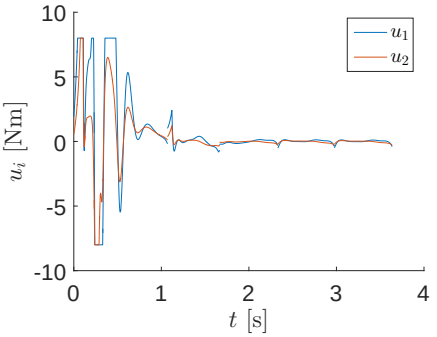
Figure 5.17: The gait with the initial perturbation $\dot{q}_1(t_0) = \dot{\theta}_*(t_0) - 0.08$.

Zero initial velocity of the stance and swing leg

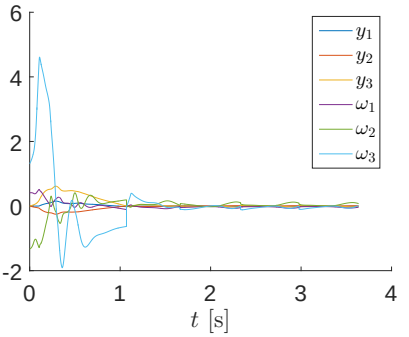
Using a controller generated with the same Q and Γ, but with $\delta = 1$ to make the controller less aggressive, the biped managed to stabilize the gait if the initial velocities of the absolute angles of the stance leg and swing leg were set to zero, while $\dot{q}_2^a = -0.75\,\mathrm{rad\,s^{-1}}$. However, this led to a spike of the torque of u_1 of almost $20\,\mathrm{N\,m}$. For this reason, saturation was enabled, which could be set to a minimum of $\pm 8\,\mathrm{N\,m}$ for the controller to being able to stabilize the motion. The result can be seen in Figure 5.18. The strange evolution of the stance leg and the torso, as seen in Figure 5.18a, is likely due to the projection $P(q)$ which becomes quite inaccurate for such large perturbations. Yet, the controller still manages to stabilize the motion after about 3 steps even with saturation enabled.



(a) Phase portrait in absolute angles.



(b) Actuator outputs.



(c) Values of the transverse states .

Figure 5.18: The gait with the initial conditions perturbed such that, measured in absolute angles, the swing and stance leg have zero velocity, while the torso is initialized with $\dot{q}_2^a(t_0) = -0.75\,\mathrm{rad\,s^{-1}}$.

5.3 Discussion

In this chapter, two methods of developing orbitally stabilizing controllers for periodic trajectories, referred to as gaits, on the planar compass biped were presented. The first used the angle of the stance leg as a motion generator (MG), which made it possible to derive the linearized transverse dynamics for five linearly independent transverse coordinates. The second method found the MG by a projection from the configuration variables, and the transverse dynamics were linearized for six excessive transverse coordinates.

Both controllers managed to stabilize the gaits tested even when subject to perturbations in the initial condition or with different system parameters than that used in the modelling of the controller. However, the stability and convergence to the nominal gait, as well as the necessary torque required, was shown to be heavily dependent on the offset parameter δ, which, for large values, resulted in a very aggressive control action. The

projection based controller was also shown to be heavily dependent of the the weighting matrix Q due to the transverse linearization, where the elements lower third diagonal had to be set to large values in order to achieve stability due to the large eigenvalues of the expansions of the function f_1. It is thus clear that the desired performance can be greatly increased by proper choice of the weighting matrices and the offset parameter; however, tuning of these was not the main focus of this thesis, so it is left as a recommendation for further work on the topic. This again ties in with the region of attraction for a found controller; finding some estimate of the region of attraction, be it precise or very conservative, will give some indication of the stability of the induced limit cycle, which can be used to determine suitable controllers in advance of implementation by changing the weighting matrices and the offset parameter. Therefore, finding an estimate of the region of attraction is also left as a recommendation for further work.

It was shown for both controllers that perturbations from the nominal trajectory could lead to large spikes in the torque required from the actuators in order to stabilize the motion. It was further shown that the controllers still managed to stabilize the motion with torque saturation enabled to some extent. However, the remaining spikes in the torques required are still not feasible to handle even for high frequency actuators and would need to be passed through a low-pass filter for implementation on a real system. Moreover, it was assumed that all the states of the biped could be perfectly measured without noise. Thus, it would be interesting to test the controllers subject to the torque reference passed through a low-pass filter, as well as the measurement of the states value with added noise or where some have states, e.g., the angular velocities, are found from observers rather than directly measured. These topics are thus left as recommendations for further work.

Although the controller developed in Section 5.1 used the angle of the stance leg as the MG, it can also be used for other choices of motion generators as long as they can be found directly as a function of the generalized coordinates of the biped. However, the construction procedure of the controller will differ for every individual choice of the MG due to the linearization of the transverse dynamics. This is not the case for the projection based controller from Section 5.2, where the transverse linearization stays the same, and one only needs to change the projection $P(q)$, its gradient and Hessian, as well as the new virtual holonomic constraints in order to linearize the transverse dynamics for different gaits. This method can be used for both gaits where the MG is known, resulting in a perfect projection as with it chosen as the angle of the stance leg, or where the MG is an unknown quantity such that the projection must be found for each individual gait. It was also shown that this control method manages to stabilize the gait even if the projection is imperfect. However, if the perturbations away from the nominal trajectory become large as in Figure 5.18, it can results in strange behaviour and instability.

Chapter 6

Conclusion and Recommendations for Further Work

6.1 Conclusion

The aim of this thesis was to plan energy efficient, symmetric gaits for an underactuated planar compass biped robot with a torso, as well as developing stabilizing controllers for these gaits. The biped was considered as a hybrid dynamical system due to the sudden change of velocities arising from the impact between the swing leg and the ground at the end of a step. This meant, that in addition to the continuous dynamics found using the method of Euler-Lagrange, an additional mapping accounting for this sudden change of velocities had to be found. For this, two discrete mappings were developed: one algebraic formulation using Newton's model of restitution and one using the conservation of angular momentum.

In order to find feasible gaits for the biped, a set of geometrical relations, called virtual holonomic constraint, between the configuration variables of the system and a scalar variable, referred to as the motion generator, were assumed to be kept invariant by feedback control. This made it possible to reduce the search space down to a two-dimensional subspace of the state space. A gait was then determined by the given virtual holonomic constraints and the evolution of the motion generator.

Three types of polynomials of a given order were tested as virtual holonomic constraints: regular polynomials, Bezièr curves and trigonometric polynomials. Gaits were found for all three types, but for the Bezièr curves it was shown that more of the polynomial variables could be determined in advance, and since the resulting curve lied in the convex hull of the polynomial parameters, the solvers needed less time to converge to a feasible gait compared to the other two types of polynomials.

Two types of motion generators were considered in the gait planning procedure: the angle of the stance leg and a quantity which was not directly found as a function of the configuration variables. A large array of feasible gaits were found using the motion gener-

ator as the angle of the stance leg. However, this choice of the motion generator restricted the search space such that the found gaits required far more energy and torque than what was feasible on the mechanical system. Using the quantity between zero and one which was not directly found from the configuration of the biped as the motion generator, gaits with very low energetic cost of transport and low torque requirements were found. It was shown that for these energy efficient gaits, the only energy added to the system was that needed in order to combat the energy dissipation due to impact.

In this thesis, two methods for generating stabilizing controllers were presented. The first method was restricted for gaits where the motion generator was chosen as the angle of the stance leg, while the second used a projection from the configuration variables in order to approximate the motion generator. Moreover, the first method used five transverse coordinates, which is the minimum number of independent transverse coordinates for the system, while the second method used six excessive transverse coordinates. For both methods, a stabilizing controller was then developed from exponentially stabilizing the linearized transverse dynamics by solving the Riccati differential equation with one jump using a semi-definite programming approach. This was done by writing the Riccati differential equation as a linear matrix inequality by taking its Schur complement. Moreover, in order to account for the instantaneous change in the states accruing at impact, a discrete mapping for the transverse coordinates was generated which then could be used to add an additional constraint to the semi-definite programming problem, which, if satisfied, ensured exponential stability of the hybrid linearized transverse dynamics. A locally exponentially orbitally stabilizing controller was generated for both methods, which was made time-invariant by recovering the time-stamp of the given moving Poincarè section to which the transverse coordinates belonged by a projection operator.

Both control methods presented were shown in numerical simulations to be able to stabilize trajectories when subject to small perturbation or changes in the system parameters compared to those used in the construction of the controllers. The found torque required for achieving stability and the convergence rate was, however, shown to be heavily dependent on the weighting matrices and the offset parameter used in the Riccati differential equation. Moreover, no method for estimating the region of attraction for a given controller was developed.

It was observed that the first method of developing a controller could only be used for gaits in which the motion generator can be found directly from the configuration. For this method, the linearization of the transverse dynamics would also differ for different motion generators. For the second method, which found the motion generator through a projection from the configuration, it was shown that it can be used for both gaits where the motion generator in a known function or where it must be approximated by a projection, with the method for finding the transverse linearization being the same regardless of the motion generator, with only the projection, as well as its gradient and Hessian, being different for different motion generators.

6.2 Recommendations for Further Work

In this section, some recommendations for further work on the topic will be presented.

Linear comparison system to the reduced dynamics

It was found that the search space of feasible gaits had many local minima, which slowed down the convergence of the solvers and could result in solutions which was far from optimal. Thus, a linear comparison system of the reduced dynamics about an already found gait, as mentioned in [36], could be used to pass approximates of the gradient and the Hessian of the objective function to the solvers. This could results in faster convergence to more optimal gaits, and possibly even be used to plan new gaits "online" on a real biped if the convergence is fast enough.

Region of attraction estimate, tuning and time-dependent weighting matrices

The controllers found and tested in this thesis managed to stabilize the trajectories subject to small perturbations and changes in systems parameters. However, there was not much emphasis on tuning of the weighting matrices Q and Γ, as well as the offset parameter δ. Clearly, the desired performance of the controller can be greatly increased by correct choice the matrices and the offset parameter. Yet, no clear method other than using the developers intuition of the system, as well as trial and error can at this time be used to better the performance. Moreover, it is likely that constant weighting matrices will not yield the optimal performance as the system is time-varying. Thus, time-dependent weighting matrices could possibly yield large increases in performance of the controllers. However, the question of finding these weighting matrices again arises. Therefore, an estimate of the region of attraction for a given controller could be used to determine the choice of the weighting matrices and the offset parameters, as well as determining the performance of a controller without rigorously testing it in numerical simulations.

Testing the controllers subject to low-pass filtered torque reference, measurement noise and imperfect measurements of some of the states

Although the controllers presented in this thesis managed to stabilize the trajectories subject to small perturbation and changes in system parameters, their robustness w.r.t. to noise in measurements or when some states can not be measured directly is not clear. Furthermore, the torques required to stabilize the gaits could have large jumps or spikes arising at impacts. Such jumps and spikes in the references sent to the actuators would not be feasible for implementation on a real system. Thus, the controllers should be tested when the torque references are passed through a low-pass filter to get a smooth reference to the actuators, as well as when there is uncertainty in the measurements of the states due to noise or as some states must be found by observers rather than measured directly.

Development of non-symmetric gaits and how to transition between them

All the gaits presented in this thesis were symmetric gaits. Thus, finding energy efficient, non-symmetric gaits is then an obvious next step for work on this topic. Moreover, as a robot must be able to turn, change pace and avoid obstacles to be viable for operation in real environments, a way to smoothly and stably change between different gaits is further left as a recommendation for further work on the topic.

Bibliography

[1] B.D.O. Anderson and J.B. Moore. *Optimal control: linear quadratic methods*. Courier Corporation, 2007.

[2] T. Anstensrud. 2-D passive compass biped walker: Analysis and robustness of stable gait. Master's thesis, 2013.

[3] S.J. Benson, Y. Ye, and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization*, 10(2):443–461, 2000.

[4] S. Bittanti, P. Colaneri, and G. De Nicolao. A note on the maximal solution of the periodic Riccati equation. *Automatic Control, IEEE Transactions on*, 34(12):1316–1319, 1989.

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

[6] S. Collins, A. Ruina, R. Tedrake, and M. Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307(5712):1082–1085, 2005.

[7] J. Currie and D.I. Wilson. OPTI: lowering the barrier between open source optimizers and the industrial MATLAB user. *Foundations of computer-aided process operations, Savannah, Georgia, USA*, pages 8–11, 2012.

[8] E.J. Davison and M.C. Maki. The numerical solution of the matrix Riccati differential equation. *Automatic Control, IEEE Transactions on*, 18(1):71–73, 1973.

[9] O. Egeland and J.T. Gravdahl. *Modeling and simulation for automatic control*. Marine Cybernetics Trondheim, Norway, 2002.

[10] G.E. Farin, J. Hoschek, and M.S. Kim. *Handbook of computer aided geometric design*. Elsevier, 2002.

[11] L.B. Freidovich, A. Shiriaev, and I.R. Manchester. Transitions between limit cycles for an underactuated system: virtual constraints approach. In *Nonlinear Control Systems*, volume 7, pages 468–473, 2007.

[12] L.B. Freidovich, A. Shiriaev, and I.R. Manchester. Stability analysis and control design for an underactuated walking robot via computation of a transverse linearization. In *Proc. 17th IFAC World Congress, Seoul, Korea*, pages 10–166, 2008.

[13] G. Gilardi and I. Sharf. Literature survey of contact dynamics modelling. *Mechanism and machine theory*, 37(10):1213–1239, 2002.

[14] R. Goebel, R.G. Sanfelice, and A. Teel. Hybrid dynamical systems. *Control Systems, IEEE*, 29(2):28–93, 2009.

[15] H. Goldstein, C.P. Poole, and J.L. Safko. *Classical Mechanics*, volume 3. Addison Wesley, 2001.

[16] J.W. Grizzle, G. Abba, and F. Plestan. Asymptotically stable walking for biped robots: Analysis via systems with impulse effects. *Automatic Control, IEEE Transactions on*, 46(1):51–64, 2001.

[17] S.V. Gusev, A. Shiriaev, and L.B. Freidovich. LMI approach for solving periodic matrix Riccati equation. 2011.

[18] S.V. Gusev, A. Shiriaev, and L.B. Freidovich. SDP-based approximation of stabilizing solutions for periodic matrix Riccati differential equations. *International Journal of Control*, (just-accepted):1–15, 2015.

[19] Y. Hurmuzlu and D.B. Marghitu. Rigid body collisions of planar kinematic chains with multiple contact points. *The international journal of robotics research*, 13(1):82–92, 1994.

[20] Y. Hürmüzlü and G.D. Moskowitz. The role of impact in the stability of bipedal locomotion. *Dynamics and Stability of Systems*, 1(3):217–234, 1986.

[21] D.W. Jordan and P. Smith. *Nonlinear ordinary differential equations: an introduction to dynamical systems*, volume 2. Oxford University Press, 1999.

[22] R. E. Kalman. Contributions to the theory of optimal control. *Bol. Soc. Mat. Mexicana*, 5(2):102–119, 1960.

[23] H.K. Khalil. *Nonlinear systems*, volume 3. Prentice hall New Jersey, 1996.

[24] Erwin Kreyszig. *Advanced engineering mathematics*. John Wiley & Sons, 10 edition, 2011.

[25] A.D. Kuo. Choosing your steps carefully. *Robotics & Automation Magazine, IEEE*, 14(2):18–29, 2007.

[26] P.X.M. La Hera, A. Shiriaev, L.B. Freidovich, U. Mettin, and S. Gusev. Stable walking gaits for a three-link planar biped robot with one actuator. *Robotics, IEEE Transactions on*, 29(3):589–601, 2013.

[27] J.R. Leigh. Functional analysis and linear control theory. 1980.

[28] G.A. Leonov. Generalization of the Andronov-Vitt theorem. *Regular and chaotic dynamics*, 11(2):281–289, 2006.

[29] J. Löfberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pages 284–289. IEEE, 2004.

[30] I.R. Manchester, U. Mettin, F. Iida, and R. Tedrake. Stable dynamic walking over uneven terrain. *The International Journal of Robotics Research*, page 0278364910395339, 2011.

[31] T. McGeer. Passive dynamic walking. *The international journal of robotics research*, 9(2):62–82, 1990.

[32] A. Mohammadi, M. Maggiore, and L. Consolini. When is a Lagrangian control system with virtual holonomic constraints Lagrangian? In *NOLCOS*, pages 512–517, 2013.

[33] J. Nestruev. *Smooth manifolds and observables*. Springer Science & Business Media, 2003.

[34] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2 edition, 2006.

[35] L. Ntogramatzidis and A. Ferrante. On the solution of the Riccati differential equation arising from the LQ optimal control problem. *Systems & Control Letters*, 59(2):114–121, 2010.

[36] S.S. Pchelkin, A. Shiriaev, U. Mettin, L.B. Freidovich, L.V. Paramonov, and S.V. Gusev. Algorithms for finding gaits of locomotive mechanisms: case studies for gorilla robot brachiation. *Autonomous Robots*, pages 1–17, 2015.

[37] S.S. Pchelkin, A. Shiriaev, A. Robertsson, L.B. Freidovich, S.A. Kolyubin, L.V. Paramonov, and S.V. Gusev. On orbital stabilization for industrial manipulators: Case study in evaluating performances of modified pd+ and inverse dynamics controllers. 2016.

[38] G. Pratt and J. Manzo. The darpa robotics challenge [competitions]. *Robotics & Automation Magazine, IEEE*, 20(2):10–12, 2013.

[39] W.J. Rugh. *Linear system theory*, volume 2. prentice hall Upper Saddle River, NJ, 1996.

[40] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura. The intelligent ASIMO: System overview and integration. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, pages 2478–2483. IEEE, 2002.

[41] A. Shiriaev and L.B. Freidovich. Transverse linearization for impulsive mechanical systems with one passive link. *Automatic Control, IEEE Transactions on*, 54(12):2882–2888, 2009.

[42] A. Shiriaev, L.B. Freidovich, and S. Gusev. Transverse linearization for controlled mechanical systems with several passive degrees of freedom. *Automatic Control, IEEE Transactions on*, 55(4):893–906, 2010.

[43] A. Shiriaev, L.B. Freidovich, and I.R. Manchester. Can we make a robot ballerina perform a pirouette? orbital stabilization of periodic motions of underactuated mechanical systems. *Annual Reviews in Control*, 32(2):200–211, 2008.

[44] A. Shiriaev, L.B. Freidovich, and I.R. Manchester. Periodic motion planning and analytical computation of transverse linearizations for hybrid mechanical systems. In *CDC*, pages 4326–4331, 2008.

[45] A. Shiriaev, J. Perram, A. Robertsson, and A. Sandberg. Explicit formulas for general integrals of motion for a class of mechanical systems subject to virtual constraints. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 2, pages 1158–1163. IEEE, 2004.

[46] A. Shiriaev, J.W. Perram, and C. Canudas-de Wit. Constructive tool for orbital stabilization of underactuated nonlinear systems: Virtual constraints approach. *Automatic Control, IEEE Transactions on*, 50(8):1164–1176, 2005.

[47] A. Shiriaev, A. Robertsson, J. Perram, and A. Sandberg. Periodic motion planning for virtually constrained Euler–Lagrange systems. *Systems & control letters*, 55(11):900–907, 2006.

[48] M.W. Spong. Partial feedback linearization of underactuated mechanical systems. In *Intelligent Robots and Systems' 94.'Advanced Robotic Systems and the Real World', IROS'94. Proceedings of the IEEE/RSJ/GI International Conference on*, volume 1, pages 314–321. IEEE, 1994.

[49] M.W. Spong and F. Bullo. Controlled symmetries and passive walking. *Automatic Control, IEEE Transactions on*, 50(7):1025–1031, 2005.

[50] M.W. Spong, J.K. Holm, and D. Lee. Passivity-based control of bipedal locomotion. *Robotics & Automation Magazine, IEEE*, 14(2):30–40, 2007.

[51] M.W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot modeling and control*, volume 3. Wiley New York, 2006.

[52] G. Teschl. *Ordinary differential equations and dynamical systems*, volume 140. American Mathematical Soc., 2012.

[53] M. Vukobratović and B. Borovac. Zero-moment point—thirty five years of its life. *International Journal of Humanoid Robotics*, 1(01):157–173, 2004.

[54] E.R. Westervelt, J.W. Grizzle, C. Chevallereau, J.H. Choi, and B. Morris. *Feedback control of dynamic bipedal robot locomotion*, volume 28. CRC press, 2007.

[55] E.R. Westervelt, J.W. Grizzle, and D.E. Koditschek. Hybrid zero dynamics of planar biped walkers. *Automatic Control, IEEE Transactions on*, 48(1):42–56, 2003.

[56] V.A. Yakubovich. A linear-quadratic optimization problem and the frequency theorem for nonperiodic systems. i. *Siberian Mathematical Journal*, 27(4):614–630, 1986.

[57] H. Ye, A.N. Michel, and L. Hou. Stability analysis of systems with impulse effects. *Automatic Control, IEEE Transactions on*, 43(12):1719–1723, 1998.

[58] F. Zhang. *The Schur complement and its applications*, volume 4. Springer Science & Business Media, 2006.

Appendix A

Various Proofs and Theorems

A.1 Proof of Theorem 2.4.1

Given a system of the form

$$\alpha(\theta)\ddot{\theta} + \beta(\theta)\dot{\theta}^2 + \gamma(\theta) = 0 \tag{A.1}$$

where $\alpha(\theta) \neq 0$ along a solution $(\theta(t), \dot{\theta}(t))$ such that one can rewrite (A.1) as

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

By setting $Y := \dot{\theta}^2(t)$, one can see that

$$\frac{dY}{dt} = \frac{d}{dt}\big(\dot{\theta}^2(t)\big) = 2\dot{\theta}\ddot{\theta}$$

and

$$\frac{dY}{dt} = \frac{\partial Y}{\partial \theta}\frac{d\theta}{dt} = \frac{\partial Y}{\partial \theta}\dot{\theta}.$$

Thus

$$\ddot{\theta} = \frac{1}{2}\frac{\partial Y}{\partial \theta}$$

holds along the solution of (A.2), giving

$$\frac{\partial Y}{\partial \theta} + 2\frac{\beta(\theta)}{\alpha(\theta)}Y + 2\frac{\gamma(\theta)}{\alpha(\theta)} = 0 \tag{A.3}$$

which is a linear Bernoulli equation which easily can be solved. By defining a function $\mu := \mu(\theta)$ such that

$$2\mu\frac{\beta(\theta)}{\alpha(\theta)} := \frac{\partial \mu}{\partial \theta}, \tag{A.4}$$

and multiplying it with (A.3), one obtains

$$\mu\frac{\partial Y}{\partial\theta}+\frac{\partial\mu}{\partial\theta}Y+2\mu\frac{\gamma(\theta)}{\alpha(\theta)}=0\ \ \Rightarrow\ \ \frac{\partial}{\partial\theta}(\mu Y)=-2\mu\frac{\gamma(\theta)}{\alpha(\theta)}. \tag{A.5}$$

Integrating both sides yields

$$Y\mu-Y(\theta_0)\mu(\theta_0)=\int_{\theta_0}^{\theta}-2\mu(s)\frac{\gamma(s)}{\alpha(s)}ds. \tag{A.6}$$

Thus, by solving (A.4) to get

$$\mu=\mu(\theta_0)\exp\left\{\int_{\theta_0}^{\theta}2\frac{\beta(s)}{\alpha(s)}ds\right\},$$

one can set

$$\psi(\theta,\theta_0)=\exp\left\{-\int_{\theta_0}^{\theta}2\frac{\beta(s)}{\alpha(s)}ds\right\}$$

and obtain from (A.6)

$$Y=\psi(\theta,\theta_0)Y(\theta_0)-\psi(\theta,\theta_0)\int_{\theta_0}^{\theta}2(\psi(s,\theta_0))^{-1}\frac{\gamma(s)}{\alpha(s)}ds. \tag{A.7}$$

Thus, the function

$$I(\theta(t),\dot\theta(t),\theta_0,\dot\theta_0)=\dot\theta^2(t)-Y(\theta(t),\dot\theta(t),\theta_0,\dot\theta_0) \tag{A.8}$$

is identically zero along the solutions of (A.2).

A.2 Proof of Theorem 2.4.2

The time derivative of the function $I=(\theta(t),\dot\theta(t),x,y)$ defined in (2.36) is given by

$$\frac{d}{dt}I=\frac{\partial I}{\partial\theta}\dot\theta+\frac{\partial I}{\partial\dot\theta}\ddot\theta. \tag{A.9}$$

where

$$\frac{\partial I}{\partial\dot\theta}=2\dot\theta \tag{A.10}$$

and

$$\begin{aligned}
\frac{\partial I}{\partial\theta}=&-\frac{\partial Y}{\partial\theta} && \text{(A.11)}\\
=&-\frac{\partial\psi}{\partial\theta}Y(x)+\frac{\partial\psi}{\partial\theta}\int_x^{\theta}2(\psi(s,x))^{-1}\frac{\gamma(s)}{\alpha(s)}ds && \text{(A.12)}\\
&+\psi\frac{\partial}{\partial\theta}\int_x^{\theta}2(\psi(s,x))^{-1}\frac{\gamma(s)}{\alpha(s)}ds && \text{(A.13)}\\
=&\frac{2\beta(\theta)}{\alpha(\theta)}\psi(\theta,x)\Big[Y(x)-\int_x^{\theta}2(\psi(s,x))^{-1}\frac{\gamma(s)}{\alpha(s)}ds\Big]+\frac{2\gamma(\theta)}{\alpha(\theta)} && \text{(A.14)}\\
=&\frac{2\gamma(\theta)}{\alpha(\theta)}-\frac{2\beta(\theta)}{\alpha(\theta)}\big[I-\dot\theta^2\big], && \text{(A.15)}
\end{aligned}$$

where Y is defined in (A.7). Thus, evaluated along the solutions of

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

the time derivative of (2.36) is given by

$$\dot{I} = \Big\{\frac{2\gamma(\theta)}{\alpha(\theta)} - \frac{2\beta(\theta)}{\alpha(\theta)}\big[I - \dot{\theta}^2\big]\Big\}\dot{\theta} + \frac{\big(u - \beta(\theta)\dot{\theta}^2 - \gamma(\theta)\big)}{\alpha(\theta)}2\dot{\theta} \tag{A.16}$$

$$= \frac{2\dot{\theta}}{\alpha(\theta)}\Big(u - \beta(\theta)I\Big). \tag{A.17}$$

A.3 Hadamard's Lemma

Lemma A.3.1 (Hadamard's Lemma, [33])**.** *Any smooth function $f(\cdot)$ in a starlike neighborhood U of a point z is representable in the form*

$$f(x) = f(z) + \sum_{i=1}^{n}(x_i - z_i)g_i(x), \tag{A.18}$$

where $g_i(\cdot)$ are smooth functions.

Proof. Consider the function

$$\varphi(t) = f(z + t(x - z)),$$

such that $\varphi(0) = f(z)$ and $\varphi(1) = f(x)$. Since $f(x)$ is smooth in a starlike (often called star convex) neighborhood U of a point z, i.e., for any point $y \in U$ all points on the interval (z, y) lies in U, then

$$\varphi(1) - \varphi(0) = \int_0^1 \frac{d\varphi(t)}{dt}dt = \int_0^1 \sum_{i=1}^{n}\frac{\partial f}{\partial x_i}(z + t(x - z))(x_i - z_i)dt$$

$$= \sum_{i=1}^{n}(x_i - z_i)\int_0^1 \frac{\partial f}{\partial x_i}(z + t(x - z))dt.$$

Since the functions

$$g_i(x) = \int_0^1 \frac{\partial f}{\partial x_i}(z + t(x - z))dt$$

are smooth in U, this concludes the proof. □

A.4 Proof of Theorem 4.2.1

For any constant annihilator $B^\perp$, one finds from (2.33) that the relation (4.20) holds as long as

$$B^\perp\left[\frac{\partial M\big(\Phi(\theta)\big)}{\partial\theta} - C\big(\Phi(\theta), \Phi'(\theta)\big)\right]\Phi'(\theta) = 0. \tag{A.19}$$

If one denotes $\hat{C}\big(\Phi(\theta),\Phi^{'}(\theta)\big)=\frac{\partial M\left(\Phi(\theta)\right)}{\partial\theta}$, one finds that its elements are given by

$$\hat{c}_{i,j}(\theta)=\sum_{k=1}^{n}\frac{\partial m_{i,j}(\Phi)}{\partial\phi_k}\phi_k^{'}.$$

Further noting from (2.10) that the elements of the matrix $C\big(\Phi(\theta),\Phi^{'}(\theta)\big)$ are given by

$$c_{i,j}\big(\Phi(\theta),\Phi^{'}(\theta)\big)=\sum_{k=1}^{n}\hat{\Gamma}_{i,j}^{k}\big(\Phi(\theta)\big)\phi_k^{'},$$

where $\hat{\Gamma}_{i,j}^{k}\big(\Phi(\theta)\big)$ is defined as

$$\hat{\Gamma}_{i,j}^{k}\big(\Phi(\theta)\big):=\frac{1}{2}\left\{\frac{\partial m_{i,j}(\Phi)}{\partial\phi_k}+\frac{\partial m_{i,k}(\Phi)}{\partial\phi_j}-\frac{\partial m_{k,j}(\Phi)}{\partial\phi_i}\right\},$$

one can define the column vector

$$D\big(\Phi(\theta),\Phi^{'}(\theta)\big)=\Big[\hat{C}\big(\Phi(\theta),\Phi^{'}(\theta)\big)-C\big(\Phi(\theta),\Phi^{'}(\theta)\big)\Big]\Phi^{'}(\theta),$$

whose elements are

$$d_i\big(\Phi(\theta),\Phi^{'}(\theta)\big)=\sum_{j=1}^{n}\sum_{k=1}^{n}\frac{1}{2}\left\{\frac{\partial m_{i,j}(\Phi)}{\partial\phi_k}-\frac{\partial m_{i,k}(\Phi)}{\partial\phi_j}+\frac{\partial m_{k,j}(\Phi)}{\partial\phi_i}\right\}\phi_k^{'}\phi_j^{'}.$$

Therefore, one can write (A.19) as

$$B^{\perp}D\big(\Phi(\theta),\Phi^{'}(\theta)\big)=0. \tag{A.20}$$

Since

$$\sum_{j=1}^{n}\sum_{k=1}^{n}\frac{1}{2}\left\{\frac{\partial m_{i,j}(\Phi)}{\partial\phi_k}-\frac{\partial m_{i,k}(\Phi)}{\partial\phi_j}\right\}\phi_k^{'}\phi_j^{'}=0$$

due to symmetry, it follows that

$$d_i\big(\Phi(\theta),\Phi^{'}(\theta)\big)=\sum_{j=1}^{n}\sum_{k=1}^{n}\frac{1}{2}\left\{\frac{\partial m_{k,j}(\Phi)}{\partial\phi_i}\right\}\phi_k^{'}\phi_j^{'}=\frac{1}{2}\Phi^{'}(\theta)^{T}\left[\frac{\partial M\big(\Phi(\theta)\big)}{\partial\phi_i}\right]\Phi^{'}(\theta).$$

Hence,

$$B^{\perp}D\big(\Phi(\theta),\Phi^{'}(\theta)\big)=\sum_{i=1}^{n}b_i^{\perp}\frac{1}{2}\Phi^{'}(\theta)^{T}\left[\frac{\partial M\big(\Phi(\theta)\big)}{\partial\phi_i}\right]\Phi^{'}(\theta), \tag{A.21}$$

from which (4.21) follows, concluding the proof.

A.5 Proof of Proposition 4.2.2

By using the relation (4.20), rewrite

$$\int_{\theta_+}^{\theta_-} \frac{\beta(\tau)}{\alpha(\tau)} d\tau$$

as

$$\int_{\theta_+}^{\theta_-} \frac{1}{\alpha(\tau)} \frac{d}{d\tau}\Big(\alpha(\tau)\Big) d\tau = \int_{\alpha(\theta_+)}^{\alpha(\theta_-)} \frac{1}{u} du,$$

which has the solution

$$\int_{\alpha(\theta_+)}^{\alpha(\theta_-)} \frac{1}{u} du = \ln\big(\alpha(\theta_-)\big) - \ln\big(\alpha(\theta_+)\big).$$

This yields

$$\exp\left\{ -\int_{\theta_+}^{\theta_-} \frac{2\beta(\tau)}{\alpha(\tau)} d\tau \right\} = \left(\frac{\alpha(\theta_+)}{\alpha(\theta_-)}\right)^2$$

and

$$\exp\left\{ \int_{\theta_+}^{s} \frac{2\beta(\tau)}{\alpha(\tau)} d\tau \right\} = \left(\frac{\alpha(s)}{\alpha(\theta_+)}\right)^2.$$

Inserting this into (4.45) and noting that $\alpha(\theta_+)$ is a constant, one obtains

$$I(\theta_+, \dot{\theta}_+, \theta_-, \dot{\theta}_-) = \dot{\theta}_-^2 - \left(\frac{\alpha(\theta_+)}{\alpha(\theta_-)}\right)^2 \left[\dot{\theta}_+^2 - \int_{\theta_+}^{\theta_-} \left(\frac{\alpha(s)}{\alpha(\theta_+)}\right)^2 \frac{2\gamma(s)}{\alpha(s)} ds \right],$$

which is equivalent to (4.48), concluding the proof.

Appendix B

Full Function Expressions

Non-zero elements of the 5 × 5 inertia matrix for the instantaneous impact model

$$
\begin{aligned}
m_{e11} =& l_1^2 m_1 + J_1 + 2L_1 \cos(q_2) l_2 m_2 + L_1^2 m_2 + l_2^2 m_2 + J_2 + L_1^2 m_3 + l_3^2 m_3 && \text{(B.1a)}\\
&+ J_3 + 2L_1 \cos(q_2 + q_3) l_3 m_3,\\
m_{e12} =& m_{e21} = L_1 \cos(q_2) l_2 m_2 + L_1 \cos(q_2 + q_3) l_3 m_3 + l_2^2 m_2 + l_3^2 m_3 && \text{(B.1b)}\\
&+ J_2 + J_3,\\
m_{e13} =& m_{e31} = L_1 \cos(q_2 + q_3) l_3 m_3 + l_3^2 m_3 + J_3, && \text{(B.1c)}\\
m_{e14} =& m_{e41} = -L_1 \sin(q_1) m_2 - L_1 \sin(q_1) m_3 - \sin(q_1) l_1 m_1 && \text{(B.1d)}\\
&- \sin(q_2 + q_1) l_2 m_2 - \sin(q_1 + q_2 + q_3) l_3 m_3,\\
m_{e15} =& m_{e51} = L_1 \cos(q_1) m_2 + L_1 \cos(q_1) m_3 + \cos(q_1) l_1 m_1 && \text{(B.1e)}\\
&+ \cos(q_1 + q_2) l_2 m_2 + \cos(q1_1 + q_2 + q_3) l_3 m_3,\\
m_{e22} =& l_2^2 m_2 + l_3^2 m_3 + J_2 + J_3, && \text{(B.1f)}\\
m_{e23} =& m_{e32} = l_3^2 m_3 + J_3, && \text{(B.1g)}\\
m_{e24} =& m_{e42} = -\sin(q_1 + q_2) l_2 m_2 - \sin(q_1 + q_2 + q_3) l_3 m_3, && \text{(B.1h)}\\
m_{e25} =& m_{e52} = \cos(q_1 + q_2) l_2 m_2 + \cos(q_1 + q_2 + q_3) l_3 m_3, && \text{(B.1i)}\\
m_{e33} =& l_3^2 m_3 + J_3, && \text{(B.1j)}\\
m_{e34} =& m_{e43} = -\sin(q_1 + q_2 + q_3) l_3 m_3, && \text{(B.1k)}\\
m_{e35} =& m_{e53} = \cos(q_1 + q_2 + q_3) l_3 m_3, && \text{(B.1l)}\\
m_{e44} =& m_1 + m_2 + m_3 && \text{(B.1m)}\\
m_{e55} =& m_1 + m_2 + m_3 && \text{(B.1n)}
\end{aligned}
$$

Functions for the second and third reduced dynamical equations

$$\begin{aligned}\alpha_2(\theta) =&(l_2^2m_2 + l_3^2m_3 + J_2 + J_3)\phi_2^{'}(\theta) + (l_3^2m_3 + J_3)\phi_3^{'}(\theta) \\ &+ L_1\cos(\phi_2(\theta))\phi_1^{'}(\theta)l_2m_2 + L_1\cos(\phi_2(\theta) + \phi_3(\theta))\phi_1^{'}(\theta)l_3m_3 \\ &+ \phi_1^{'}(\theta)l_2^2m_2 + \phi_1^{'}(\theta)l_3^2m_3 + J_2\phi_1^{'}(\theta) + J_3\phi_1^{'}(\theta),\end{aligned} \tag{B.2a}$$

$$\begin{aligned}\beta_2(\theta) =&(l_2^2m_2 + l_3^2m_3 + J_2 + J_3)\phi_2^{''}(\theta) + (l_3^2m_3 + J_3)\phi_3^{''}(\theta) \\ &+ L_1\sin(\phi_2(\theta))(\phi_1^{'}(\theta))^2l_2m_2 + L_1\sin(\phi_2(\theta) + \phi_3(\theta))l_3m_3(\phi_1^{''}(\theta))^2 \\ &+ L_1\phi_1^{''}\cos(\phi_2(\theta))l_2m_2 + L_1\phi_1^{''}(\theta)\cos(\phi_2(\theta) + \phi_3(\theta))l_3m_3 \\ &+ \phi_1^{''}(\theta)l_2^2m_2 + \phi_1^{''}(\theta)l_3^2m_3 + J_2\phi_1^{''}(\theta) + J_3\phi_1^{''}(\theta),\end{aligned} \tag{B.2b}$$

$$\gamma_2(\theta) = \cos(\phi_2(\theta) + \phi_1(\theta))gl_2m_2 + \cos(\phi_2(\theta) + \phi_3(\theta) + \phi_1(\theta))gl_3m_3, \tag{B.2c}$$

$$\begin{aligned}\alpha_3(\theta) =&(l_3^2m_3 + J_3)\phi_2^{'}(\theta) + (l_3^2m_3 + J_3)\phi_3^{'}(\theta) \\ &+ L_1\cos(\phi_2(\theta) + \phi_3(\theta))\phi_1^{'}(\theta)l_3m_3 + \phi_1^{'}(\theta)l_3^2m_3 + J_3\phi_1^{'}(\theta),\end{aligned} \tag{B.3a}$$

$$\begin{aligned}\beta_3(\theta) =&(l_3^2m_3 + J_3)\phi_2^{''}(\theta) + (l_3^2m_3 + J_3)\phi_3^{''}(\theta) + L_1\sin(\phi_2(\theta) \\ &+ \phi_3(\theta))l_3m_3(\phi_1^{'}(\theta))^2 + L_1\phi_1^{''}(\theta)\cos(\phi_2(\theta) + \phi_3(\theta))l_3m_3 \\ &+ \phi_1^{''}(\theta)l_3^2m_3 + J_3\phi_1^{''}(\theta),\end{aligned} \tag{B.3b}$$

$$\gamma_3(\theta) = \cos(\phi_2(\theta) + \phi_3(\theta) + \phi_1(\theta))gl_3m_3. \tag{B.3c}$$

Right-hand side of the reduced dynamics

$$\begin{aligned}g_{v_1}(\theta, y) =& - L_1\cos(\phi_2 + y_2)l_2m_2 - L_1\cos(\phi_2 + y_2 + \phi_3 + y_3)l_3m_3 \\ &- l_2^2m_2 - l_3^2m_3 - J_2 - J_3,\end{aligned} \tag{B.4a}$$

$$g_{v_2}(\theta, y) = - L_1\cos(\phi_2 + y_2 + \phi_3 + y_3)l_3m_3 - l_3^2m_3 - J_3, \tag{B.4b}$$

$$\begin{aligned}g_{\dot{y}_2}(\theta, \dot{\theta}, y, \dot{y}) =&(L_1\sin(\phi_2 + y_2)l_2m_2 + L_1\sin(\phi_2 + y_2 + \phi_3 + y_3)l_3m_3)\dot{y}_2 \\ &+ ((L_1l_2m_2\phi_2^{'}(\theta)\dot{\theta} + L_1l_2m_2\dot{\theta} + (L_1\phi_2^{'}(\theta)l_2m_2 \\ &+ L_1l_2m_2)\dot{\theta})\sin(\phi_2 + y_2) + (2L_1l_3m_3\dot{y}_3 + L_1l_3m_3\dot{\theta} \\ &+ L_1l_3m_3\phi^{'}(\theta)_2\dot{\theta} + L_1l_3m_3\phi^{'}(\theta)_3\dot{\theta} + (L_1\phi_2^{'}(\theta)l_3m_3 \\ &+ L_1\phi_3^{'}(\theta)l_3m_3 + L_1l_3m_3)\dot{\theta})\sin(\phi_2 + y_2 + \phi_3 + y_3)),\end{aligned} \tag{B.4c}$$

$$\begin{aligned}g_{\dot{y}_3}(\theta, \dot{\theta}, y, \dot{y}) =&L_1\sin(\phi_2 + y_2 + \phi_3 + y_3)l_3m_3\dot{y}_3 + (L_1l_3m_3\dot{\theta} \\ &+ L_1l_3m_3\phi_2^{'}(\theta)\dot{\theta} + L_1l_3m_3\phi_3^{'}(\theta)\dot{\theta} + (L_1\phi_2^{'}(\theta)l_3m_3 \\ &+ L_1\phi_3^{'}(\theta)l_3m_3 + L_1l_3m_3)\dot{\theta})\sin(\phi_2 + y_2 + \phi_3 + y_3),\end{aligned} \tag{B.4d}$$

$$
\begin{aligned}
g_{y_2}(\theta,\dot{\theta},\ddot{\theta}) =&((L_1\phi_2^{'}(\theta)l_2m_2 + 2L_1l_2m_2)\sin(\phi_2(\theta)) + (L_1\phi_2^{'}(\theta)l_3m_3 \qquad \text{(B.5a)}\\
&+ L_1\phi_3^{'}(\theta)l_3m_3 + 2L_1l_3m_3)\sin(\phi_2(\theta)+\phi_3(\theta)))\ddot{\theta}\\
&+ ((L_1\phi_2^{'}(\theta)l_2m_2 + (L_1\phi_2^{'}(\theta)l_2m_2 + L_1l_2m_2)\phi_2^{'}(\theta))\cos(\phi_2(\theta))\\
&(L_1\phi_2^{'}(\theta)l_3m_3 + L_1\phi_3^{'}(\theta)l_3m_3 + (L_1\phi_2^{'}(\theta)l_3m_3 + L_1\phi_3^{'}(\theta)l_3m_3\\
&+ L_1l_3m_3)\phi_2^{'}(\theta) + (L_1\phi_2^{'}(\theta)l_3m_3 + L_1\phi_3^{'}(\theta)l_3m_3\\
&+ L_1l_3m_3)\phi_3^{'}(\theta))\cos(\phi_2(\theta)+\phi_3(\theta)))\dot{\theta}^2\\
&+ \sin(\phi_2(\theta)+\theta)gl_2m_2 + \sin(\phi_2(\theta)+\phi_3(\theta)+\theta)gl_3m_3,\\
g_{y_3}(\theta,\dot{\theta},\ddot{\theta}) =&(L_1\phi_2^{'}(\theta)l_3m_3 + L_1\phi_3^{'}(\theta)l_3m_3 + 2L_1l_3m_3)\sin(\phi_2(\theta)+\phi_3(\theta))\ddot{\theta}\\
&(L_1\phi_2^{'}(\theta)l_3m_3 + L_1\phi_3^{'}(\theta)l_3m_3 + (L_1\phi_2^{'}(\theta)l_3m_3 + L_1\phi_3^{'}(\theta)l_3m_3\\
&+ L_1l_3m_3)\phi_2^{'}(\theta) + (L_1\phi_2^{'}(\theta)l_3m_3 + L_1\phi_3^{'}(\theta)l_3m_3\\
&+ L_1l_3m_3)\phi_3^{'}(\theta))\cos(\phi_2(\theta)+\phi_3(\theta))\dot{\theta}^2\\
&+ \sin(\phi_2(\theta)+\phi_3(\theta)+\theta)gl_3m_3. \qquad \text{(B.5b)}
\end{aligned}
$$

Appendix C

Implementations and Additional Results

C.1 The Integral Function using the Trapezoidal Rule

Consider the integral function

$$I(\theta, \dot{\theta}, \theta_0, \dot{\theta}_0) = \dot{\theta}^2 - e^{-\psi(\theta_0,\theta)} \times \left(\dot{\theta}_0^2 - \int_{\theta_0}^{\theta} e^{\psi(\theta_0,s)} \times \frac{2\gamma(s)}{\alpha(s)} ds \right), \tag{C.1}$$

where

$$\psi(x_0, x) = \int_{x_0}^{x} \frac{2\beta(\tau)}{\alpha(\tau)} d\tau. \tag{C.2}$$

Several numerical integration methods can be used to approximate the integral. Here, the trapezoidal rule will be used, which is described next.

Suppose one wants to to integrate the function $f(x)$ over the interval $[a, b]$, then the trapezoidal method is given by [24]

$$\int_a^b f(x)dx \approx h\left[\frac{1}{2}f(a) + \sum_{i=1}^{n-1} f(x_i) + \frac{1}{2}f(b)\right],$$

where $h = (b-a)/n$ defines the integration step for some integer n, chosen sufficiently large, and $x_{i+1} = x_i + hi$ such that $x_1 = a + h$.

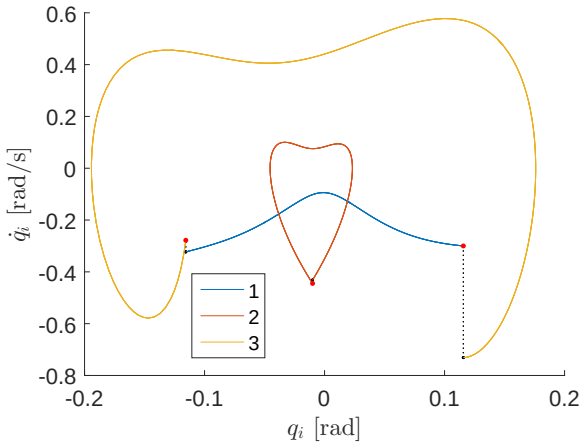
Thus, for finding an approximation of the integral function using the trapezoidal rule, Algorithm 2 is proposed.

Algorithm 2 Approximation of the integral function using the trapezoidal rule

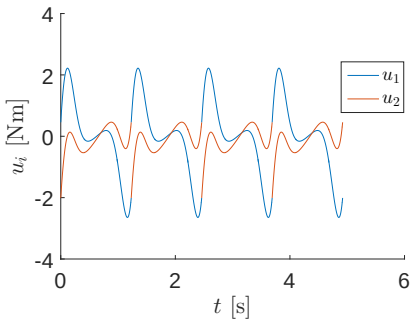
Set $h = (\theta - \theta_0)/n$ for some sufficiently large n
$B(0) = (\beta(\theta_0)/\alpha(\theta_0))/2$
$\Gamma = (\gamma(\theta_0)/\alpha(\theta_0))/2$
$x_0 = \theta_0$
for i from 0 to $n-1$ **do**
    $x_{i+1} = x_i + h$
    $\lambda = \beta(x_{i+1})/\alpha(x_{i+1})$
    $B(i+1) = B(i) + \lambda$
    $\Pi(i) = h\big(B(i) + \lambda/2\big)$
    $\Gamma = \Gamma + (\gamma(x_{i+1})/\alpha(x_{i+1}))\exp\big(2\Pi(i)\big)$
end for
$\Pi(n) = h\big(B(n) + (\beta(\theta)/\alpha(\theta))/2\big)$
$\Gamma = \Gamma + \big[(\gamma(\theta)/\alpha(\theta))\exp\big(2\Pi(n)\big)\big]/2$
$\Psi_1 = \exp\big(-2\Pi(n)\big)$
$\Psi_2 = 2h\Gamma$
Return $I(\theta, \dot{\theta}, \theta_0, \dot{\theta}_0) = \dot{\theta}^2 - \Psi_1\big(\dot{\theta}_0^2 - \Psi_2\big)$

C.2 Stabilizing a Gait with the Motion Generator as the Angle of the Stane Leg using the Projection Method

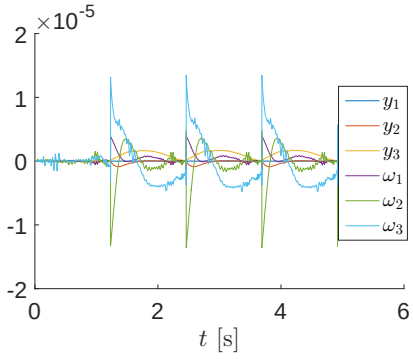
Here, some results from stabilizing the gait depicted in Figure 4.5 using the method from Section 5.2 will be presented. Since the angle of the stance leg is the motion generator, one has the perfect projection $P(q) = q_1$, with $P^{'}(q) = [1,\ 0,\ 0]$ and $P^{''}(q) = \mathbf{0}_6$. For all the results presented, the weighting matrices were $Q = I_6$ and $\Gamma = I_2$, with $\delta = 1$. The controller was made time-invariant using the same projection method as (5.42). Figure C.1 shows the system system simulated in closed loop with zero initial perturbations. Figure C.2 shows the results when the system parameters are scaled up by 10%. Figure C.3 shows the system response to a initial perturbations of the angular velocity of the stance leg by $0.15\,\mathrm{rad\,s^{-1}}$.



(a) Phase portrait in absolute angles.

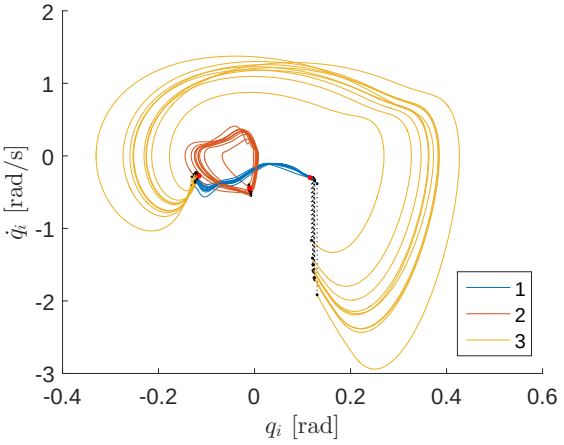


(b) Actuator outputs.

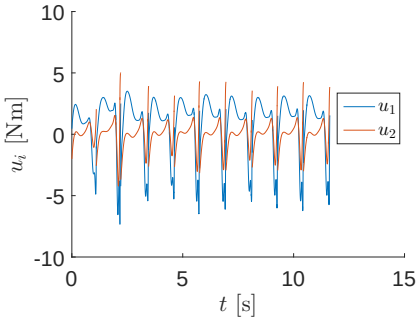


(c) Values of the transverse states .

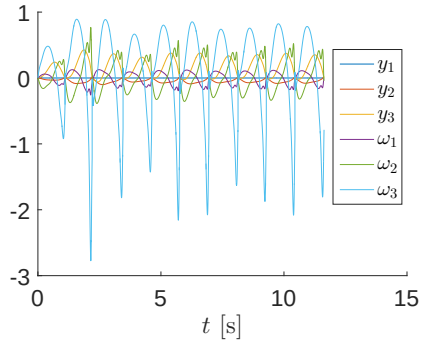
Figure C.1: The gait simulated in closed loop with zero initial perturbation.



(a) Phase portrait in absolute angles.

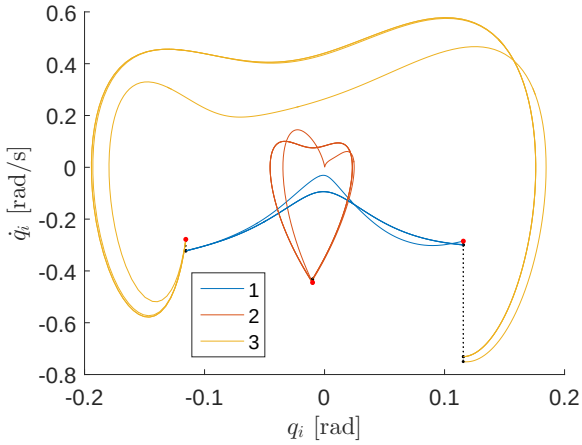


(b) Actuator outputs.

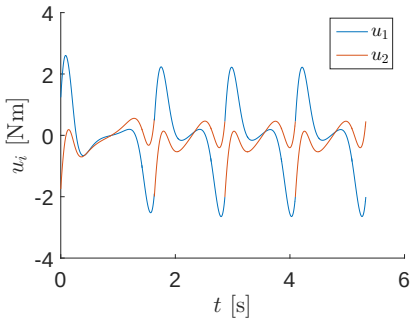


(c) Values of the transverse states .

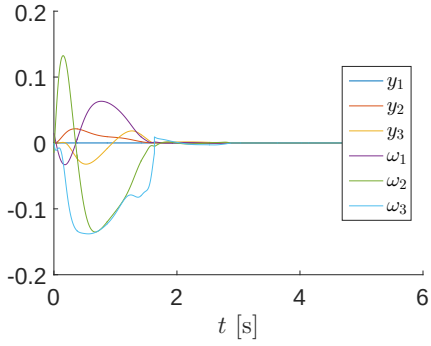
Figure C.2: Shows the evolution of the system with the parameters scaled up by 10%.



(a) Phase portrait in absolute angles.



(b) Actuator outputs.



(c) Values of the transverse states .

Figure C.3: The gait with the initial perturbation $\dot{q}_1(t_0) = \dot{\theta}_*(t_0) + 0.015$.