

DESIGN OPTIMIZATION, ANALYSIS, AND CONTROL
OF WALKING ROBOTS

by

Wankun Sirichotiyakul



A thesis

submitted in partial fulfillment

of the requirements for the degree of

Master of Science in Mechanical Engineering

Boise State University

August 2019

BOISE STATE UNIVERSITY GRADUATE COLLEGE

DEFENSE COMMITTEE AND FINAL READING APPROVALS

of the thesis submitted by

Wankun Sirichotiyakul

Thesis Title: Design Optimization, Analysis, and Control of Walking Robots

Date of Final Oral Examination: 30th May 2019

The following individuals read and discussed the thesis submitted by student Wankun Sirichotiyakul, and they evaluated the presentation and response to questions during the final oral examination. They found that the student passed the final oral examination.

Aykut Satici, Ph.D.	Chair, Supervisory Committee
John Chiasson, Ph.D.	Member, Supervisory Committee
John Gardner, P.E., Ph.D.	Member, Supervisory Committee
Joseph Guarino, P.E., Ph.D.	Member, Supervisory Committee
Trevor Lujan, Ph.D.	Member, Supervisory Committee

The final reading approval of the thesis was granted by Aykut Satici, Ph.D., Chair of the Supervisory Committee. The thesis was approved by the Graduate College.

ACKNOWLEDGMENTS

My greatest appreciation naturally goes to my advisor, Professor Aykut C. Satici. In the past two years, his patient guidance, encouragement, and a vast amount of academic knowledge have been invaluable contributions to my success as a graduate student. I am forever grateful for this opportunity.

I would like to extend my gratitude to the other members of my thesis committee: Professor John Chiasson, Professor John Gardner, Professor Joseph Guarino, and Professor Trevor Lujan for generously offering their time, guidance, and goodwill throughout the preparation and review of this document.

To my friends Jason Kuwada, Oliver Alvarez, Kalin Gibbons, and my lab mate Nardos Ashenafi, thank you for the lunch breaks and for listening to my endless raves and rantings. You all have provided the much needed escapes from stress and made my experience at Boise State fun and exciting.

Last but not least, I wish to express my deepest gratitude and appreciation to my father, my mother, and my sisters for their unconditional love and relentless support. None of my accomplishments would have been possible without my lovely family. I am grateful for them much more than I could ever express.

ABSTRACT

Passive dynamic walking refers to the dynamical behavior of mechanical devices that are able to naturally walk down a shallow slope in a stable manner, without using actuation or sensing of any kind. Such devices can attain motions that are remarkably human-like by purely exploiting their natural dynamics. This suggests that passive dynamic walking machines can be used to model and study human locomotion; however, there are two major limitations: they can be difficult to design, and they cannot walk on level ground or uphill without some kind of actuation.

This thesis presents a mechanism design optimization framework that allows the designer to find the best design parameters based on the chosen performance metric(s). The optimization is formulated as a convex problem, where its solutions are globally optimal and can be obtained efficiently.

To enable locomotion on level ground and uphill, this thesis studies a robot based on a passive walker: the rimless wheel with an actuated torso. We design and validate two control policies for the robot through the use of scalable methodology based on tools from mathematical analysis, optimization theory, linear algebra, differential equations, and control theory.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	iv
ABSTRACT	v
LIST OF TABLES	ix
LIST OF FIGURES	x
1 Introduction	1
1.1 Mechanism Design Optimization	2
1.1.1 Literature Survey	2
1.1.2 Contributions	4
1.2 Control of a Torso-Actuated Rimless Wheel	5
1.2.1 Literature Survey	5
1.2.2 Contributions	7
2 Background	8
2.1 Sums of Squares Optimization	8
2.2 Analyzing Periodic Motions of Dynamical Systems	13
2.2.1 Limit Cycles	13
2.2.2 Poincaré Section and Poincaré Map	15
2.2.3 Transverse Coordinates and Transverse Dynamics	17
2.3 Linear Quadratic Regulator	19

2.3.1	Continuous-Time LQR	19
2.3.2	Discrete-Time LQR	20
2.3.3	Region of Attraction Estimation	21
3	Convex Optimization of Robotic Manipulator Designs via Sum-of-Squares Programming	23
3.1	Kinematic and Dynamic Performance Indices	23
3.2	Polynomial Manipulator Equations	25
3.2.1	Kinematic Jacobian and the Manipulability Matrix	26
3.2.2	Mass Matrix	28
3.3	The Optimization Problem	29
3.4	Methods	30
3.4.1	Unconstrained semidefinite optimization of GII and GDI	31
3.4.2	Formulating the constraints	33
3.4.3	Multicriteria Optimization	33
3.5	Optimization Results	33
3.6	Conclusions	35
4	Orbital Stabilization of a Torso-Actuated Rimless Wheel	37
4.1	System Model	37
4.1.1	Swing model – continuous dynamics	39
4.1.2	Impact model – discrete transition	40
4.1.3	Hybrid model	41
4.2	Control Design	42
4.2.1	Discrete-Time (DT) Controller	42
4.2.2	Continuous-Time (CT) Controller	44

4.2.3	Implementation of the Controllers	46
4.2.4	Region of Attraction Estimation	47
4.2.5	Impact Map Analysis and Controller Improvements	48
4.3	Simulation Studies	51
4.3.1	Discrete-Time (DT) Controller	51
4.3.2	Continuous-Time (CT) Controller	52
4.3.3	Comparison of the CT and DT Controllers	54
4.4	Experimental Studies	55
4.4.1	Prototype	56
4.4.2	Results	57
4.5	Conclusion	58
5	Conclusions	59
	REFERENCES	60

LIST OF TABLES

3.1	Indeterminates for the two-link manipulator	26
3.2	Design variables α and parameters β	30
3.3	Results of independent optimizations with respect to GII and GDI	34
4.1	Parameters of the system	52
4.2	Coefficients of Poincaré map approximation	52

LIST OF FIGURES

2.1	Visualization of Poincaré section	18
3.1	Schematic of a two-link serial manipulator in elbow-out posture	27
3.2	Pareto-front curve generated via sums-of-squares optimization	35
4.1	Schematic of rimless wheel with torso	38
4.2	Continuous-time Controller Computation	53
4.3	Convergence of the DT and CT controllers	54
4.4	Regions of attraction of the DT and CT controllers	55
4.5	Experimental prototype	56
4.6	Experimental results	57

CHAPTER 1

INTRODUCTION

The problem of robot locomotion is traditionally approached by studying and mimicking people's walking motions in terms of their joint angles (knees, ankles, etc.) and the characteristics of the ground reaction forces. While walking robots designed using this approach can execute robust motions, they are not versatile and not very human-like. Furthermore, the control paradigm associated with this approach typically imposes the desired dynamics to the system without using the natural dynamics to its advantage. This demands high control authority and leads to large energy requirements [1].

A more clever approach is to study mechanisms that exhibit passive dynamic walking behavior with gait appearance that is comparable to human locomotion. By designing robots that exploit their natural dynamics, the energy requirement for locomotion is significantly reduced, and the walking motion is more natural.

Building such robots require, in particular, judicious mechanism and control system designs. Kinematic and dynamic performance of a robot are highly dependent on its geometrical parameters. This gives an opportunity to approach the mechanism design task as an optimization problem. Control systems need to be designed to expend control effort only when needed, and allow the natural dynamics to perform most of the work for locomotion.

In this thesis, we tackle these two issues: (1) the design optimization of mechanisms using efficient methods that can guarantee global optimality of the solution, and (2) a control problem for hybrid dynamical systems whose motions are periodic.

1.1 Mechanism Design Optimization

The design of robotic manipulators may be achieved by optimizing certain performance measures, which are dependent on the robot's geometrical parameters. Traditionally, the design optimization problems have been formulated using nonlinear and nonconvex techniques. This is largely due to the nonlinear nature of the kinematic and dynamic equations of most robotic manipulators and the performance metrics.

Finding solutions for nonlinear, nonconvex optimization problems are difficult and inefficient. To address these issues, this thesis presents a general framework for casting the design optimization problem as a convex one, where the solution is guaranteed to be globally optimal and can be obtained efficiently.

1.1.1 Literature Survey

The majority of the literature on design optimization of robotic manipulators use nonlinear and nonconvex methods to attack the problem. A brief account of such methods include the culling algorithm [2, 3, 4], different variations of the genetic algorithm [5, 6, 7, 8], the differential evolution algorithm [9], performance-chart based methods [10], workspace atlases [11], controlled random search technique [12], and Monte Carlo method [13, 14]. These methods have relative strengths and weaknesses; however, a common undesirable characteristic that plagues them is their tendency to

get stuck at a local optimum. Branch-and-bound methods overcome this problem at the cost of limited accuracy and the need to perform a much more expensive search.

Convex optimization methods circumvent all the issues described above. However, a complete formulation of the design optimization problem as a convex problem has eluded the researchers so far. Lou *et. al.* [15] discretize the space of design parameters into a set of discrete nodes to formulate and solve a convex optimization problem constrained by a number of linear matrix inequalities (LMIs) at each discretized point. Combining the results of each problem yields a better design, but not necessarily the global optimum since the designer does not know how smooth the design objective is, *a priori*. Convex optimization has also been used to find the largest ellipse contained in the singularity-free workspace of a manipulator [16], where the design parameters are kept as constants.

An optimal design of a robotic manipulator often can only be achieved by considering many competing objectives [2]. There exist several studies in which multiple design criteria have been addressed for this purpose. Hayward *et al.* define the relationship between multiple criteria and utilize sensitivities of these criteria to conduct a hierarchical optimization study [17]. Multiple objectives are considered sequentially in [18, 19, 20, 4] by searching for parameter sets resulting in near optimal kinematic performance and then selecting the design exhibiting the best dynamic performance from this reduced parameter space. Task-priority [21], probabilistic weighting [22], composite index [23], and tabular methods [24] are among the other approaches that consider multiple criteria.

Pareto methods incorporate all optimization criteria within the optimization process and address them simultaneously to find a set of efficient solutions. Each design alternative in the solution set corresponds to a non-dominated design in the objective

space. In other words, these methods aim to construct the Pareto-front hypersurface representing the design trade-offs between multiple criteria. Once such a hypersurface resolving the design trade-offs is obtained, an appropriate design on this hypersurface can be selected, taking into account the other design requirements of the particular application in consideration.

1.1.2 Contributions

The main contribution of this work, presented in Chapter 3, is the reformulation of the problem of design optimization of robotic manipulators as a convex optimization problem by invoking sum-of-squares (SoS) techniques [25]. This allows for global optimization of many performance indices efficiently with accuracy up to machine precision. Moreover, sum-of-squares optimization scales exceptionally well as the number of design parameters increase, contrary to branch and bound methods whose computational complexity increases exponentially. In this problem, we consider the multi-criteria optimization of two performance measures: the global kinematic and dynamic isotropy indices, GII and GDI , respectively [4]. The Pareto-front curve is obtained by using the scalarization (weighted-sum) method to turn the multi-objective optimization problem into a series of single-objective ones. We use SoS programming to solve each individual single-objective problem. A proof-of-concept case-study that formulates and solves the design optimization problem for the planar two-link manipulator is presented. This mechanism is also representative of the rimless wheel with a torso undergoing the swing phase, a robot based on a passive dynamic walking device of interest in this thesis.

This material has been published in the Third IEEE International Conference on Robotic Computing (IRC 2019) [26].

1.2 Control of a Torso-Actuated Rimless Wheel

Passive dynamic walking robots locomote without sensing or actuation of any kind. Perhaps the simplest and most popular of its kind is the *rimless wheel*, a multi-spoked wheel with the rim removed, first developed by McGeer [27], almost two decades ago. When a rimless wheel is launched on a downhill ramp, it is able to sustain steady locomotion as the kinetic energy lost through impacts is supplied back to the system from the stored potential energy over the swing phase. Robots with such fortuitous natural dynamics are highly energy efficient because no external energy needs to be provided for locomotion. On the other hand, we still want to actuate walking robots to overcome at least the following two major issues: (1) sustenance of dynamically stable walking uphill or on level ground [28], and (2) robustness to external disturbances.

This thesis presents the methodology to devise control laws that address these two major issues by utilizing the control authority at our disposal due to the addition of actuator(s).

1.2.1 Literature Survey

One way to enable sustained locomotion of a rimless wheel on level ground is to have a design that incurs zero kinetic energy losses during impacts, which is achievable by reducing the spoke collision velocity to zero. Gomes [29] uses an inertial disk that is coupled to the rimless wheel through a torsional spring, thus storing and releasing energy to the robot at appropriate times. Such a robot will need a shallow ramp to sustain walking motion in reality to compensate for the dissipation in the springs, inertia disk, and frictional losses.

Another way to enable walking is to have an actively powered rimless wheel.

Some extensions to the rimless wheel of this sort can be found in [30, 31] where the spokes expand and contract, thus supplying energy to the robot. A drawback of this implementation is that it requires multiple actuators. Other examples include rimless wheels with a wobbling mass [32] or a rotating disc [33] that transfers energy through dynamic coupling.

Bhounsule [34] designed a rimless wheel with a torso which may be actuated to provide the necessary energy for continuous walking on level or uphill slopes. The torso position is regulated at a certain moment during the gait by an event-based, discrete controller. This controller assumes the unrealistic ability to instantaneously change the torso angle. In this thesis, we show that there are performance benefits of continuously moving the torso including energy efficiency and robustness.

The rimless wheel with torso is modeled as a hybrid dynamical system, which exhibits both continuous and discrete phases [35, 36]. A common approach to study the periodic orbit of such systems involves an analysis of the Poincaré map. Local orbital stability is characterized by the stability of an associated “first-return map” that describes the repeated passes of the system trajectories on a single, predefined transversal hypersurface. This map relates one pass to another through a difference equation, allowing for analysis of orbital stability and for design of feedback policy in discrete-time.

Related to the Poincaré map analysis is the *transverse coordinates and transverse dynamics* method [37, 38, 39], also known as the “Moving Poincaré section” method. This construction provides a continuous description of the system’s dynamics as it traverses along the periodic orbit, thus allowing for analysis of orbital stability and for design of feedback policy in continuous-time.

1.2.2 Contributions

In this work, we provide the design, analysis, and implementation of controllers for robotic locomotion. The contributions of this work are summarized below:

- Design two novel controllers for a reduced system,
- Estimate and compare the regions of attractions of these controllers,
- Provide an implementation of these controllers on the full system dynamics that guarantee its stability,
- Provide an analysis of the impact map that yields an optimization procedure to find energetically-preferable nominal walking gaits,
- Provide extensive simulation and partial experimental support for the theory.

This material has been accepted for publication in the ASME 2019 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE 2019).

CHAPTER 2

BACKGROUND

This chapter aims to provide the reader with the preliminary background to formulate the optimization problem for mechanism design (Chapter 3) and to design control laws for the rimless wheel actuated by a torso (Chapter 4).

Section 2.1 introduces the sums-of-squares (SoS) optimization technique, one application of which is to certify the nonnegativity of polynomials over semialgebraic sets (regions defined by polynomial inequalities). We use this technique to optimize the design of robotic manipulators in Chapter 3 and to approximate the region of attraction of dynamical systems in Chapter 4.

Section 2.2 provides an overview on a number of common techniques used to study periodic motions of dynamical systems. Then, in conjunction with Section 2.3, which gives an overview on the Linear Quadratic Regulator (LQR), these techniques are used in Chapter 4 to devise control laws for the rimless wheel with torso.

2.1 Sums of Squares Optimization

The 17th of David Hilbert's 23 "Hilbert problems" is

Given a multivariate polynomial that takes only non-negative values over the reals, can it be represented as a sum of squares of rational functions?

The solution of this problem goes back to David Hilbert and Emil Artin [40]. This branch of mathematics is called real algebraic geometry or real algebra. In this section, we provide the necessary background on positive semidefinite matrices and SoS polynomials. We emphasize the aspect of computing the SoS decomposition. The interested reader can refer to [41, 42] for more information.

Definition 1 *A polynomial $p \in \mathbb{R}[x]$ of degree $d = \eta_1 + \dots + \eta_n$, $\eta_i \in \mathbb{N}$, i.e.,*

$$p(x) = \sum_{\eta_1 + \dots + \eta_n \leq d} c_\eta x_1^{\eta_1} \dots x_n^{\eta_n} \quad (2.1)$$

is a sum-of-squares if there exist a finite number of polynomials $p_i \in \mathbb{R}[x]$ such that p can be written as

$$p(x) = \sum_i p_i^2(x). \quad (2.2)$$

Note that if $p(x)$ is SoS, then $p(x) \geq 0 \forall x \in \mathbb{R}^n$. Hilbert proved that not every positive semidefinite polynomial can be written as a sum-of-squares. However, in 1927, Artin's Theorem answered Hilbert's seventeenth problem by stating that every semidefinite polynomial is a sum-of-squares of rational functions [40]. This leads us to the question: When can a polynomial be written as a sum-of-squares?

Theorem 2.1.1 [43] *A polynomial $p \in \mathbb{R}[x]$ of degree $2d$ has a sum-of-squares decomposition if and only if there exists a positive semidefinite matrix Q such that*

$$p(x) = m^\top(x)Qm(x), \quad (2.3)$$

where m is the vector of all monomials in x_1, \dots, x_n of degree less than or equal to d , i.e. $m(x) = \begin{bmatrix} 1 & x_1 & x_2 & \dots & x_n & x_1x_2 & \dots & x_n^d \end{bmatrix}$. There exist $\binom{n+d}{n}$ such monomials.

This representation theorem, based on the Gram matrix method, implies that the set of sum-of-squares polynomials are parametrized by the (convex) set of positive semidefinite matrices. SoS programming is then an optimization problem that

searches for a positive semidefinite matrix Q that will ensure the nonnegativity of $p(x)$ for any $x \in \mathbb{R}^n$ while satisfying a set of affine constraints. Note that this is a *finite-dimensional, convex optimization problem*, which can be solved efficiently via semidefinite programming.

We are often interested in whether a polynomial p is positive semidefinite on only a subset of \mathbb{R}^n , which is described by inequalities of the form $g_i(x) \geq 0$, $g_i \in \mathbb{R}[x]$, $\forall i = 1, \dots, m$, i.e., the set $S = \{x \in \mathbb{R}^n : g_i(x) \geq 0 \forall i = 1, \dots, m\}$. We can pose this problem as another semidefinite optimization problem by employing what is called the *S-procedure*. Following [25], we have the following lemma

Lemma 2.1.2 *A polynomial $p \in \mathbb{R}[x]$ is a sum-of-squares polynomial on the set S if there exist SoS polynomials $s_0, s_1, \dots, s_m \in \mathbb{R}[x]$ such that*

$$p(x) = s_0(x) + \sum_{i=1}^m s_i(x)g_i(x). \quad (2.4)$$

These results are especially interesting because there exist efficient numerical algorithms, called semidefinite programming algorithms, which allow us to translate these theorems to be solved on computer. Parrilo [42] established the connection between the SoS decomposition of Theorem 2.1.1 and semidefinite programming. A semidefinite program, which has the following form, may be interpreted as a generalization of a linear program

$$\begin{aligned} & \underset{x}{\text{minimize}} && c^\top x \\ & \text{subject to} && F_0 + \sum_{i=1}^k x_i F_i \preceq 0, \\ & && Ax = b, \end{aligned} \quad (2.5)$$

where $x \in \mathbb{R}^k$ is the decision variable, $c \in \mathbb{R}^k$, $F_i = F_i^\top \in \mathbb{R}^{m \times m}$, $A \in \mathbb{R}^{p \times k}$ and $b \in \mathbb{R}^p$. The inequality sign \preceq designates negative semidefiniteness, i.e., the symmetric

matrix $F_0 + \sum_{i=1}^k x_i F_i$ has no positive eigenvalues. A semidefinite program with more than 100,000 variables can be efficiently and reliably solved with the help of interior point methods in polynomial time and non-heuristic stopping criteria. An extensive discussion of semidefinite programming and its applications can be found in [44, 25]. We bridge the gap between the Gram matrix method and semidefinite programming by the following theorem

Theorem 2.1.3 [42] *The existence of a sum-of-squares decomposition of a polynomial in n indeterminates of degree $2d$ can be decided by solving a semidefinite program.*

An example taken from [42] illustrates the theorem. Consider the polynomial in two indeterminates of degree 4

$$p(x) = 2x_1^4 + 2x_1^3x_2 - x_1^2x_2^2 + 5x_2^4.$$

We want to check whether p can be written as a sum-of-squares polynomial. By Theorem 2.1.1, we look for a matrix Q such that with the vector of monomials $m(x) = \begin{bmatrix} x_1^2 & x_2^2 & x_1x_2 \end{bmatrix}^\top$ we have

$$\begin{aligned} p(x) &= m^\top(x)Qm(x) \\ &= m^\top(x) \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{12} & q_{22} & q_{23} \\ q_{13} & q_{23} & q_{33} \end{bmatrix} m(x) \\ &= q_{11}x_1^4 + q_{22}x_2^4 + q_{33}x_1^2x_2^2 + 2q_{12}x_1^2x_2^2 + 2q_{13}x_1^3x_2 + 2q_{23}x_1x_2^3. \end{aligned}$$

Equating the coefficients leads to

$$q_{11} = 2, q_{22} = 5, q_{33} + 2q_{12} = -1, q_{13} = 1, q_{23} = 0.$$

We are left with one unknown parameter, q_{12} (or q_{33}). Whether $p \in \mathbb{R}[x]$ is an SoS polynomial or not is equivalent to the question whether a q_{12} exists such that $Q \succeq 0$.

The feasibility of the following semidefinite program then would verify that p is a sum-of-squares polynomial

$$Q = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 5 & 0 \\ 1 & 0 & -1 \end{bmatrix} + q_{12} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -2 \end{bmatrix} \succeq 0.$$

One solution to this feasibility problem is $q_{12} = -3$, which implies that p is indeed an SoS polynomial. The author of this thesis uses the freely available Julia packages `SumOfSquares.jl` along with `JuMP.jl` and `Mosek.jl` to solve such semidefinite optimization problems [45].

Finally, recall that a matrix $P \in \mathbb{R}^{m \times m}$ is nonnegative definite if and only if P can be factored as $P = M^\top M$. The same characterization can be made for matrices whose entries are polynomials, i.e. $P(x) \in \mathbb{R}[x]^{m \times m}$. We say that a symmetric polynomial matrix $P(x) \in \mathbb{R}[x]^{m \times m}$ is positive semidefinite if $P(x) \succeq 0$ for all $x \in \mathbb{R}^n$. Further, we define an SoS matrix as follows

Definition 2 [25] *A symmetric polynomial matrix $P(x) \in \mathbb{R}[x]^{m \times m}$, $x \in \mathbb{R}^n$, is an SoS matrix if there exists a polynomial matrix $M(x) \in \mathbb{R}[x]^{s \times m}$ for some $s \in \mathbb{N}$, such that $P(x) = M^\top(x)M(x)$.*

This definition reduces to the standard SoS notion when $m = 1$. An $m \times m$ matrix is a representation of an m -variate quadratic form. Thus, an SoS matrix can be interpreted in terms of a polynomial with m additional variables.

Lemma 2.1.4 [25] *Let $P(x) \in \mathbb{R}[x]^{m \times m}$ be a symmetric polynomial matrix with $x \in \mathbb{R}^n$. Let $p(x, z) := z^\top P(x)z$ be the associated scalar polynomial in $m+n$ variables $[x; z]$, where $z = \begin{bmatrix} z_1 & \cdots & z_m \end{bmatrix}$.*

1. *The matrix $P(x)$ is positive semidefinite if and only if $p(x, z)$ is nonnegative.*
2. *The matrix $P(x)$ is an SoS matrix if and only if $p(x, z)$ is an SoS polynomial in $\mathbb{R}[x; z]$.*

This characterization of SoS matrices suggests that semidefinite programming can also be used to verify whether a symmetric polynomial matrix is positive semidefinite. We take advantage of this result to perform matrix norm minimization in order to optimize the design of robotic manipulators, as elaborated in Section 3.4. We also use this result to maximize the estimated volume of the region of attraction of control systems, as elaborated in Section 4.2.4.

2.2 Analyzing Periodic Motions of Dynamical Systems

In this section, we provide an overview of a number of techniques that are used to study periodic motions of dynamical systems. In particular, we study the concepts of limit cycles, Poincaré section, the first return (Poincaré) map, transverse coordinates, transverse dynamics and its linearization.

2.2.1 Limit Cycles

Consider the following general nonlinear control system described by:

$$\dot{x} = f(x, u), \tag{2.6}$$

where $x \in \mathbb{R}^n$ is a state vector, $u \in \mathbb{R}^m$ is a vector of control inputs, and f is a sufficiently smooth vector field. We assume that the control input is fixed, $u = \pi(x)$, which yields an autonomous dynamical system of the form

$$\dot{x} = f(x). \quad (2.7)$$

where by abuse of notation we use the symbol f for the closed-loop system as well. The solution (or flow), $\phi(\cdot, x) : \mathbb{R} \rightarrow \mathbb{R}^n$ to the ordinary differential equation (ODE) (2.7) starting at $x \in \mathbb{R}^n$ is said to be an *orbit* of the system. A *periodic orbit* or cycle is an orbit that is topologically closed.

Definition 3 *A solution $x^*(t)$ of (2.7) is a cycle or periodic orbit if there exists a finite $T > 0$ such that*

$$x^*(t) = x^*(t + T), \quad \forall t \geq 0. \quad (2.8)$$

We can think of a periodic orbit of a walking robot as a “walking gait”. If the robot starts on this gait, it will be followed indefinitely. We want to know what happens if the robot starts away from the walking gait. This leads us to the notion of limit cycles, which characterizes if system trajectories near a periodic orbit converge to it.

The definition of limit cycles involves the ideas of ω - and α -limit points and sets.

Definition 4 [46] *Given a trajectory $\gamma : t \mapsto x(t)$ of the system in (2.7). An ω -limit point is a point p for which there exists a sequence $t_n \rightarrow \infty$ such that*

$$\lim_{n \rightarrow \infty} x(t_n) = p. \quad (2.9)$$

Similarly, if there exists a sequence $t_n \rightarrow -\infty$ such that

$$\lim_{n \rightarrow \infty} x(t_n) = q, \quad (2.10)$$

then the point q is called an α -limit point of the trajectory γ . The set of all ω -limit points of γ is called the ω -limit set of γ , denoted by $\omega(\gamma)$. The set of all α -limit points of γ is called the α -limit set of γ and it is denoted by $\alpha(\gamma)$.

Using the notion of the ω - and α -limit set, we can now rigorously define the concept of limit cycles.

Definition 5 [46] *A limit cycle Γ of (2.7) is a periodic orbit which is the α - or ω -limit set of some trajectory of (2.7) other than Γ itself. If a limit cycle Γ is the ω -limit set of every other trajectory in its neighborhood, Γ is said to be an ω -limit cycle or a **stable** limit cycle. Likewise, if Γ is the α -limit set of neighboring trajectories, Γ is said to be an α -limit cycle or an **unstable** limit cycle.*

This idea is not to be confused with trajectory tracking, which requires $x(t) \rightarrow x^*(t)$ as $t \rightarrow \infty$. Consider two solutions starting from different points on the periodic orbit: $x_1(0) = x^*(t_1)$, $x_2(0) = x^*(t_2)$, with $t_1 \neq t_2$. To certify asymptotic stability of this system to a particular trajectory (trajectory tracking), we would require that $\|x_1(t) - x_2(t)\| \rightarrow 0$ as $t \rightarrow \infty$. But after each period, the two solutions will each return to their starting points: $x_1(kT) = x^*(t_1)$ and $x_2(kT) = x^*(t_2)$, which cannot be equal for all k because of the condition $t_1 \neq t_2$ imposed earlier.

The following two subsections describe the methods used to certify stability of limit cycles. These results, along with the results in Section 2.3, are used to a design feedback control policy $\pi(x)$ so that a limit cycle of the system in (2.6) is stable.

2.2.2 Poincaré Section and Poincaré Map

The construction of a Poincaré section and the Poincaré first-return map is a useful technique to analyze the behavior of periodic orbits of dynamical systems. We summarize these ideas following [47, 48].

Let γ be a periodic orbit of the n -dimensional system in (2.6). Let S be an $n - 1$ dimensional hypersurface transversal to the flow at a point $p \in \gamma$. If $n(x)$ is the normal vector to S at x , the transversality ensures that the condition $f^\top(p)n(p) \neq 0$ is satisfied, i.e., the hypersurface S is not tangent to the flow at p . The hypersurface S , visualized in Figure 2.1(a), is referred to as the Poincaré section.

The Poincaré map $P : S \rightarrow S$ is a map from S to itself, obtained by following trajectories from one intersection with S to the next. If $x_k \in S$ denotes the k th intersection, then the Poincaré map relates x_{k+1} to x_k through the difference equation

$$x_{k+1} = P(x_k). \quad (2.11)$$

If x^* is a *fixed point* of P , i.e., $P(x^*) = x^*$, then a trajectory γ starting at x^* on S must return to x^* after some time T . The trajectory γ is therefore a periodic orbit for the original system. By looking at the behavior of P near this fixed point, we can determine the stability of the orbit γ .

If $P(x)$ exists for all $x \in S$, then this method converts stability analysis of limit cycles to the stability analysis of fixed points of a map. It is typically impossible to find a closed-form formula for P , but it can often be obtained through numerical approximation. Once P is obtained, we can infer local stability of a limit cycle with an eigenvalue analysis of the linearization of P . The limit cycle is considered locally exponentially stable if all eigenvalues of the linearization of P have magnitude less than one, i.e., $|\lambda_i| < 1$ for all $i \in \{1, 2, \dots, n-1\}$.

One disadvantage of the Poincaré map analysis is that it lacks a continuous representation of the system's evolution along the periodic orbit, but rather focuses at only a *single* section along the orbit. This limits its usefulness for control design for robot locomotion. Furthermore, approximating the Poincaré map requires integration of the dynamics over an entire cycle, which demands additional computational resources. The following subsection introduces another set of tools for analyzing the stability of limit cycles that overcomes these shortcomings.

2.2.3 Transverse Coordinates and Transverse Dynamics

To facilitate the analysis of the stability of a periodic orbit, we introduce a new coordinate system adapted to the periodic orbit of the system. In these coordinates, the said stability coincides with the stability of the origin in the “transverse coordinates”. In order to define this adapted coordinate system, we start by explaining the idea of a moving Poincaré section following [37, 38, 49].

Given a state $x \in \mathbb{R}^n$ in the original coordinate system, define τ , a scalar coordinate aligned with the flow of the system along a periodic orbit x^* . At each τ , we define a hypersurface transversal to this orbit, resulting in a family of hypersurfaces $\mathcal{S}(\tau)$, mimicking the behavior of a Poincaré section that is *moving* along the orbit. This is visualized in Figure 2.1b.

The remaining coordinates, denoted $x_\perp \in \mathbb{R}^{n-1}$ and referred to as the *transverse coordinates*, are defined such that they represent the location of the state x on each $\mathcal{S}(\tau)$ relative to the periodic orbit, with $x_\perp = 0$ implying that $x = x^*(\tau)$. Once the new coordinate system is constructed, we then define a smooth mapping $\Pi : x \mapsto (x_\perp, \tau)$ that relates original coordinate system to the transversal coordinates

$$x_\perp(\tau) = \Pi(\tau) x, \quad x \in \mathcal{S}(\tau). \quad (2.12)$$

Having defined the mapping $\Pi(\cdot)$, and assuming that it is continuously differentiable, the dynamics (2.6) can now be rewritten in terms of the new coordinate system (x_\perp, τ) as

$$\dot{\tau} = f_1(x_\perp, \tau, u) \quad (2.13a)$$

$$\dot{x}_\perp = f_2(x_\perp, \tau, u). \quad (2.13b)$$

This construction gives us a continuous representation of the system’s evolution along the periodic orbit, as opposed to at a single section when using the Poincaré map.

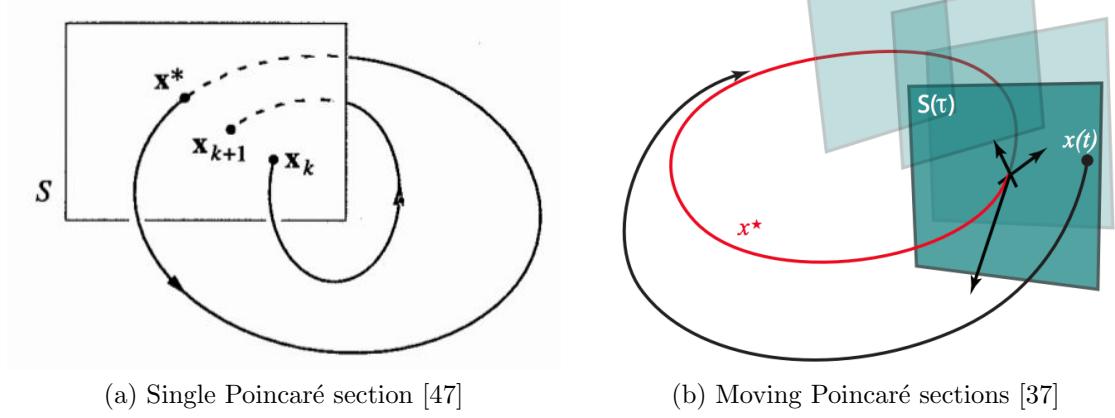


Figure 2.1: Visualization of Poincaré section

It allows for the use of a Lyapunov function $V(x_{\perp}, \tau)$ to prove that a periodic orbit of the system is a stable limit cycle by requiring that $V(x_{\perp}, \tau)$ vanishes everywhere along the orbit, and is strictly positive everywhere away from the orbit.

Theorem 2.2.1 [50] *Given a dynamical system $\dot{x} = f(x)$ with $x \in \mathbb{R}^n$ where f is continuous, $x^*(\tau)$ is a periodic solution, and a smooth mapping $\Pi : x \mapsto (x_{\perp}, \tau)$ where x_{\perp} vanishes on x^* . If there exists a function $V(x_{\perp}, \tau)$ such that the following hold:*

$$\begin{aligned} V(0, \tau) &= \dot{V}(0, \tau) = 0, \quad \forall \tau, \\ V(x_{\perp}, \tau) &> 0, \quad \forall \tau, \quad \forall x_{\perp} \neq 0, \\ \dot{V}(x_{\perp}, \tau) &< 0, \quad \forall \tau, \quad \forall x_{\perp} \neq 0, \end{aligned}$$

then the periodic orbit $x^(t)$ is an asymptotically stable limit cycle.*

The linearization of the transverse dynamics (2.13b), referred to as the *transverse linearization*, is given by

$$\dot{x}_{\perp} = \left(\frac{\partial f_2}{\partial x_{\perp}}(x^*(t), u^*(t)) \right) x_{\perp} + \left(\frac{\partial f_2}{\partial u}(x^*(t), u^*(t)) \right) u. \quad (2.14)$$

Stabilization of this linearized system is equivalent to the local exponential stabilization of the periodic orbit of the nonlinear system, and can be approached using

standard techniques from linear control theory. Furthermore, there is a standard method for constructing Lyapunov functions for linear systems, which will be useful for estimating the region of attraction, as elaborated in Section 2.3.3.

2.3 Linear Quadratic Regulator

The fundamental idea in optimal control theory is to cast the problem of finding a control policy as the long-term optimization of some scalar cost function. Coming up with an optimal control policy is equivalent to finding a solution to Hamilton-Jacobi-Bellman (HJB) equation. However, finding such a solution is an intractable task for most nonlinear systems. One important special case, where a closed-form solution is available, is when the dynamics is linear and the cost function is quadratic in the states. This control policy is called the Linear Quadratic Regulator (LQR).

In the previous sections, we considered the time-varying linearization of the transverse dynamics and the Poincaré map. We make use of the continuous, time-varying LQR to stabilize the linearization (2.14) of the transverse dynamics, and the discrete-time LQR to stabilize the linearization of the Poincaré map.

2.3.1 Continuous-Time LQR

Consider the following continuous-time linear time-varying control system

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

with the finite-horizon cost function given by

$$J = \int_{t_0}^{t_1} (x^\top Q(t)x + u^\top R(t)u) dt, \quad Q(t) \succeq 0, \quad R(t) \succ 0,$$

where $\succeq 0$ denotes positive-semidefiniteness and $\succ 0$ denotes positive-definiteness. The process of finding the optimal control policy involves finding the optimal cost-to-

go function $J^*(x)$, i.e., the cost that will be accumulated when running the optimal controller, which satisfies the Hamilton-Jacobi-Bellman (HJB) equation:

$$0 = \min_u \left(x^\top Q(t)x + u^\top R(t)u + \frac{\partial J^*}{\partial x} (A(t)x + B(t)u) \right), \quad \forall x.$$

It turns out that $J^*(x) = x^\top P(t)x$, and that the optimal control policy is given as

$$u = -R^{-1}(t)B^\top(t)P(t)x,$$

where $P(t)$ is the solution to the *differential Riccati equation*

$$-\dot{P}(t) = P(t)A(t) + A^\top(t)P(t) - P(t)B(t)R^{-1}(t)B^\top(t)P(t) + Q(t),$$

and is obtained by solving the equation backward in time from $t = t_1$. Although this equation is quadratic in $P(t)$, making its solution non-trivial, it is well known that the equation can be solved efficiently and has a single positive-definite solution as long as the system is controllable.

This result is used to design a control policy that stabilizes a periodic orbit based on transverse linearization as described in Section 2.2.3.

2.3.2 Discrete-Time LQR

For the following discrete-time linear time-invariant system:

$$x_{k+1} = Ax_k + Bu_k,$$

with the infinite-horizon cost function

$$J = \sum_{k=0}^{\infty} (x_k^\top Qx_k + u_k^\top Ru_k), \quad Q \succeq 0, R \succ 0.$$

The result from discrete LQR algorithm gives the optimal control policy

$$u_k = -(R + B^\top PB)^{-1} (B^\top PA) x_k,$$

where P is the solution the *discrete-time algebraic Riccati equation*

$$0 = -P + A^\top P A - (A^\top P B)^\top (R + B^\top P B)^{-1} (B^\top P A) + Q.$$

This result is used to design a control policy that stabilizes a periodic orbit based on the Poincaré map as described in Section 2.2.2.

2.3.3 Region of Attraction Estimation

When a feedback policy for a nonlinear system is designed based on its linearization, the following question is of particular importance: what is the region of attraction of the closed-loop system?

Definition 6 [51] *Suppose the dynamical system (2.7) is asymptotically stable about the origin. Let $\gamma(t)$ be a solution of (2.7) that starts at the initial state x at time $t = 0$. Then, the set of all points x such that $\gamma(t)$ is defined for all $t > 0$ and $\lim_{t \rightarrow \infty} \gamma(t) = 0$ is the **region of attraction** of (2.7).*

Region of attraction is then the neighborhood of the origin in which the controller is guaranteed to stabilize the system. The following theorem characterizes approximations to the region of attraction of the system to the periodic orbit in terms of the sublevel sets of a Lyapunov function.

Theorem 2.3.1 [37] *For some compact set $D := \{(x_\perp, \tau) : V(x_\perp, \tau) \leq \rho\}$, if V satisfies the conditions in Theorem 2.2.1 for all $(x_\perp, \tau) \in D$, then D is an inner approximation for the region of attraction of the system in (2.13) to the stable limit cycle $x^*(\tau)$.*

If the feedback policy $u(\tau, x_\perp)$ for (2.13) is designed using LQR, then there is a natural candidate for the Lyapunov function $V(x_\perp, \tau) = x_\perp^\top P x_\perp$, where P , which is positive definite, is the solution to the corresponding Riccati equation. In the

discrete-time case, P is the solution to the discrete algebraic Riccati equation. In the continuous-time case, $P = P(\tau)$, the solution to the differential Riccati equation.

Once we fix a polynomial form for the Lyapunov function, we can formulate a semidefinite optimization problem that looks for the best inner approximation to the region of attraction as a sublevel set of this Lyapunov function. We start by sampling a fine sequence $\{\tau_i\}$, $i = 1, 2, \dots, N$ such that $\tau_1 = 0$ and $\tau_N = T$, where T denotes the period of the periodic orbit. The following optimization problem is solved for each fixed τ_i to arrive at the estimation of the region of attraction of the system (2.13).

$$\begin{aligned} & \text{maximize} && \rho \\ & \text{subject to} && (V(x_\perp) - \rho) - \lambda(x_\perp) dV(x_\perp) \text{ is SoS,} \\ & && \lambda(x_\perp) \text{ is SoS,} \end{aligned}$$

where the term SoS denotes a sum-of-squares constraint on the corresponding polynomial. In the discrete-time LQR case, dV denotes the change in V between each successive pass of the trajectory at the Poincaré section. In the continuous-time LQR case, dV denotes the Lie derivative of V along the trajectories of the transverse system (2.13b). The formulations in this section are used in Section 4.2.4 to approximate the region of attraction of each controller under study.

CHAPTER 3

CONVEX OPTIMIZATION OF ROBOTIC MANIPULATOR DESIGNS VIA SUM-OF-SQUARES PROGRAMMING

This chapter addresses the framework for casting the robotic manipulator design as a convex optimization problem using Sum-of-Squares programming. All earlier formulations of this problem used nonlinear optimization techniques, which have poorer convergence and efficiency characteristics.

The organization of this chapter is as follows: Section 3.1 starts with the introduction to quantification of kinematic and dynamic performance for robotic manipulators. Section 3.2 presents the derivation of kinematic and dynamic equations of a robotic manipulator in polynomial form. In Section 3.3, we present the general optimization problem that maximizes the two performance metrics presented in Section 3.1. Lastly, in Section 3.4, we present a convex formulation of the optimization problem introduced in Section 3.3. We provide a discussion of the results and conclusions of the work in the last two sections.

3.1 Kinematic and Dynamic Performance Indices

To achieve robotic manipulators with low parasitic effects, both kinematic and dynamic performance need to be optimized. One way to enhance dynamic performance

is to design a device with low effective inertia so that the impedance values are kept low. A disadvantage of this approach is that it may impose unwanted restrictions to the overall workspace of the device. Similar effects of low inertia may be achieved through active inertia cancellation. The applicability of this approach is limited by actuator bandwidth, as well as issues with sensor and quantization noise.

A more clever way to quantify device performance is to consider how *isotropic* it is. The relationship between the robot's actuators and its end-effector varies with position and direction. Mechanical isotropy refers to the absence of such variations. A design with minimal variations, i.e. maximal isotropy, will result in a more uniform performance throughout the workspace, while simultaneously making most efficient use of the available actuators.

To measure the isotropy of a robotic manipulator, Salisbury and Craig [52] proposed considering the condition number of the kinematic Jacobian matrix at a given configuration. This measure locally characterizes the directional isotropy for both force/motion transmission accuracy and actuator utilization. Since the objective of the design problem is to minimize the parasitic effects of the manipulator over the workspace, we are interested in a *global* performance metric that quantifies the isotropy of robotic manipulators.

For kinematic performance, the metric we choose to use in this work is an extension to the condition number method called the Global Isotropy Index (*GII*), as introduced in [4]. A manipulator with maximal *GII* corresponds to a design with best worst-case kinematic performance, increasing the efficiency of actuator utilization. *GII* is expressed as

$$GII = \inf_{\theta \in W} \frac{\underline{\sigma}(J(\alpha, \theta))}{\bar{\sigma}(J(\alpha, \theta))},$$

where W denotes the workspace, J the kinematic Jacobian matrix. $\bar{\sigma}, \sigma$ are the largest and smallest singular values of J , α is a known function of link lengths, and θ denotes configuration variables.

To optimize dynamic performance, we choose to use the worst-case dynamic performance measure: the Global Dynamic Index (*GDI*), as introduced in [4]. The eigenvalues of the mass matrix capture the relationship between actuator force/torque and end-effector acceleration. The largest effect of mass on dynamic performance can be measured by extracting the largest singular value of the mass matrix and finding its maximum value over the workspace. *GDI* is calculated as the inverse of this quantity. Optimizing *GDI* corresponds to finding a design with minimal inertial interference by the system. *GDI* is mathematically expressed as

$$GDI = \inf_{\theta \in W} \frac{1}{1 + \bar{\sigma}(H(\alpha, \theta))},$$

where W denotes the workspace, H the mass matrix, $\bar{\sigma}$ the largest singular value of H , α is a known function of link lengths, and θ denotes configuration variables.

These performance metrics are chosen for ease of comparison with the literature. The SoS optimization techniques provide enough flexibility to analyze mechanisms with other performance metrics that are relevant to the designer.

3.2 Polynomial Manipulator Equations

The aim of this section is to present the necessary steps to derive the kinematic Jacobian and the mass matrix of a robotic manipulator in polynomial form.

We start by defining indeterminates for the i^{th} joint variable. If joint i is prismatic, we let the indeterminate x_i denote the joint displacement d_i . If joint i is revolute, we define the indeterminates as $x_i = \cos(\theta_i)$ and $y_i = \sin(\theta_i)$, where θ_i is the angle

Table 3.1: Indeterminates for the two-link manipulator

x_1	y_1	x_2	y_2
$\cos \theta_1$	$\sin \theta_1$	$\cos \theta_2$	$\sin \theta_2$

of rotation of joint i . In this case, the additional constraint $x_i^2 + y_i^2 = 1$ must be enforced to generate an accurate kinematic description of the manipulator. These indeterminate definitions are summarized in Table 3.1.

Following [53], after these definitions of the indeterminates, all the homogeneous transformations relating various frames become polynomial functions of x_i 's and y_i 's. Consequently, every partial derivative of these position-level kinematic quantities will also be polynomial functions of the same variables.

As an ongoing case-study for this thesis, we apply this method to the two-link manipulator depicted in Figure 3.1. The two-link manipulator can be characterized by the lengths l_1, l_2 of its links. In Section 3.3 we introduce a number of constraints on the decision variables, which are functions of l_1 and l_2 , to completely define the workspace (see shaded area in Figure 3.1) over which the optimization is carried out.

3.2.1 Kinematic Jacobian and the Manipulability Matrix

The i^{th} column of the kinematic Jacobian of a serial manipulator is given by

$$J_i = \begin{cases} \begin{bmatrix} z_{i-1} \times (o_n - o_{i-1}) \\ z_{i-1} \end{bmatrix} & \text{if joint } i \text{ is revolute} \\ \begin{bmatrix} z_{i-1} \\ 0 \end{bmatrix} & \text{if joint } i \text{ is prismatic} \end{cases},$$

where z_{i-1} is the axis of actuation of joint $i - 1$ and o_i is the origin of frame i . Since these are position-level kinematic quantities, they are all polynomial functions

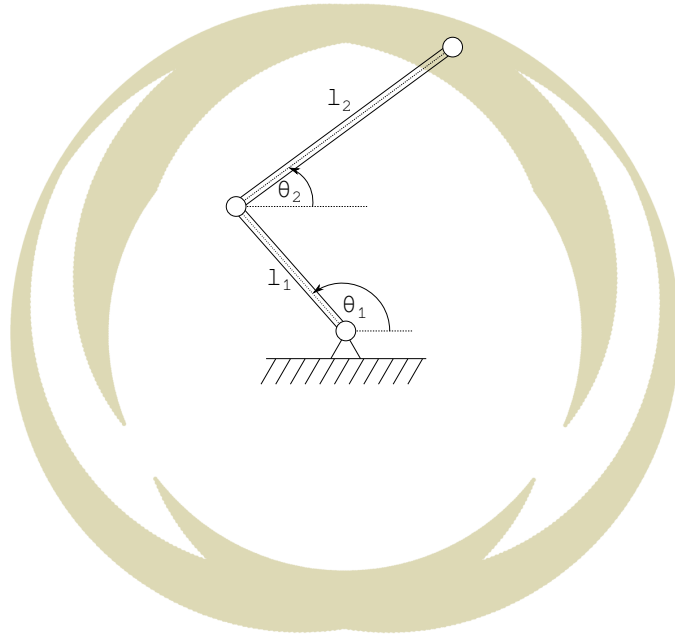


Figure 3.1: Schematic of a two-link serial manipulator in elbow-out posture
The shaded area represents the workspace over which the optimization is carried out

of the indeterminates introduced above. The manipulability matrix is then defined as $M := JJ^\top$.

For the two-link manipulator, using the definitions in Table 3.1, the kinematic Jacobian is computed to be

$$J = \begin{bmatrix} -l_1 \sin \theta_1 & -l_2 \sin \theta_2 \\ l_1 \cos \theta_1 & l_2 \cos \theta_2 \end{bmatrix} = \begin{bmatrix} -l_1 y_1 & -l_2 y_2 \\ l_1 x_1 & l_2 x_2 \end{bmatrix}, \quad (3.1)$$

from which the manipulability matrix is computed to be

$$M = JJ^\top = \sum_{k=1}^2 \begin{bmatrix} l_k^2 y_k^2 & -l_k^2 x_k y_k \\ -l_k^2 x_k y_k & l_k^2 x_k^2 \end{bmatrix}. \quad (3.2)$$

Remark 3.2.1 *Even for higher degree-of-freedom (DoF) manipulators, the kinematic manipulability matrix is a linear function of the square of the link lengths provided that*

the joint angles are defined as absolute angles. This fact allows for the formulation of the mechanism design problem as a convex optimization problem.

3.2.2 Mass Matrix

The mass matrix of a robotic manipulator is a function of the rotation matrices that relate each frame to the base frame as well as the kinematic Jacobian matrix. We have already argued that these rotation matrices and the kinematic Jacobian are polynomial functions of the indeterminates, implying that the mass matrix $H(x, y)$ is a polynomial function of the indeterminates as well.

By the operator norm definition of the largest singular value of a matrix, we have $\bar{\sigma}(H) = \|H\|_2$. It is also well-known that the Frobenius norm of a matrix dominates its two norm, i.e., $\|H\|_2 \leq \|H\|_F$. In turn, the Frobenius norm of a positive semidefinite matrix is dominated by its trace [54], i.e., $\|H\|_F \leq \text{tr}(H)$. Thus, we have the chain of inequalities

$$\bar{\sigma}(H) = \|H\|_2 \leq \|H\|_F \leq \text{tr}(H) \leq \sqrt{r}\|H\|_F,$$

where r denotes the number of degrees-of-freedom (DoF) of the manipulator. The last inequality follows from usual norm inequalities and the fact that the Frobenius norm can also be expressed as the square root of the sum of squares of the eigenvalues of H [54]. Then, it follows from matrix norm inequalities that

$$\bar{\sigma}(H) \leq \text{tr}(H) \leq r\bar{\sigma}(H). \quad (3.3)$$

To construct the mass matrix from the kinetic energy of the planar two-link manipulator, without loss of generality, we assume the center of mass of each link is situated at the geometric center of the link. With this assumption, the mass matrix is given by

$$H = \begin{bmatrix} I_1 + \frac{1}{4}ml_1^2 & \frac{1}{2}m_2l_1l_2(x_1x_2 + y_1y_2) \\ \frac{1}{2}m_2l_1l_2(x_1x_2 + y_1y_2) & I_2 + \frac{1}{4}m_2l_2^2 \end{bmatrix}, \quad (3.4)$$

where I_1, I_2 are the moments of inertia of the link about the axis perpendicular to the plane of operation, m_1, m_2 are the masses of each link and $m = (m_1 + 4m_2)$. Note that, in this case, inequalities (3.3) hold with $r = 2$.

3.3 The Optimization Problem

To optimize kinematic and dynamic performance of a two-link planar manipulator using the performance metrics discussed in Section 3.1, one can formulate an optimization problem whose objective is to maximize GII and GDI simultaneously. This implies that mechanism is optimally close to kinematic isotropy, while the effective mass is simultaneously minimized. In standard (negative null) form, the multi-objective optimization problem can be written as

$$\begin{aligned} & \underset{\alpha}{\text{maximize}} && F(\alpha, \beta, x, y) \\ & \text{subject to} && G(\alpha, \beta, x, y) \leq 0 \\ & && \alpha_l \leq \alpha \leq \alpha_u, \end{aligned} \quad (3.5)$$

where F is a column vector containing the objective functions that depend on the design variables α , parameters β , summarized in Table 3.2, and x, y are the configuration variables. G represents the inequality constraints that also depend on the design variables, parameters, and the configuration. Finally, $\alpha_l = 0.1$ and $\alpha_u = 1.0$ stand for the lower and upper bounds on the design variables, respectively. For the two-link planar manipulator, these quantities are given by

$$F = \begin{bmatrix} GII \\ GDI \end{bmatrix}, G = \begin{bmatrix} -y_2 \\ x_1y_2 - x_2y_1 \\ x_1x_2 + y_1y_2 - \epsilon \\ \epsilon - x_1x_2 - y_1y_2 \\ c_i^\top \alpha + d_i \end{bmatrix}, i = 1, \dots, m,$$

where the first two elements of G force an elbow-out posture for the manipulator, the next two elements prevent singular configurations upon selection of $0 < \epsilon < 1$, and the last m elements are affine constraints on the decision variables. We choose 2 additional affine constraints, i.e., $m = 2$.

Table 3.2: Design variables α and parameters β

	Symbol	Definition	Unit
α_1	l_1^2	squared length of link 1	m^2
α_2	l_2^2	squared length of link 2	m^2
β_1	$\epsilon = 0.5$	bound on $ \cos(\theta_1 - \theta_2) $	—
β_2	$c_1 = [1 \ 1]^\top$	parameters for affine constraints on the decision variables	—
β_3	$c_2 = -[1 \ 1]^\top$		—
β_4	$d_1 = -\frac{9}{70}$		m^2
β_5	$d_2 = \frac{1}{10}$		m^2

3.4 Methods

Theorem 2.1.3 states that SoS programming can be viewed as a special case of semidefinite programming, which is simply a linear program over the cone of positive semidefinite matrices. We need to formulate the optimization problem such that the objective function is a linear function of the decision variables and the constraints are linear matrix inequalities. The difficulty is that GII and GDI are defined in terms of the singular values of the Jacobian and the mass matrices, respectively,

which are nonlinear and nonconvex functions of the decision variables. The workspace constraints could also contain nonlinear terms. These issues must be addressed before SoS programming techniques can be invoked.

3.4.1 Unconstrained semidefinite optimization of *GII* and *GDI*

We use matrix norm minimization technique [55] to address the nonconvexity of the performance measure *GII*. Consider the problem

$$\underset{\alpha \in \mathbb{R}^n}{\text{minimize}} \quad \|A(\alpha)\|_2,$$

where $\|\cdot\|_2$ denotes the spectral norm, i.e. the maximum singular value. We use the fact that $\|A(\alpha)\|_2 \leq s$ if and only if $AA^\top \preceq s^2I$ (and $s \geq 0$) to express the problem in the form

$$\begin{aligned} & \underset{(\alpha, s) \in \mathbb{R}^{n+1}}{\text{minimize}} && s \\ & \text{subject to} && sI - A(\alpha)A(\alpha)^\top \succeq 0. \end{aligned} \tag{3.6}$$

If the $A(\alpha)A(\alpha)^\top$ is a linear function of α , then this is a semidefinite program.

We want to use this method to determine the maximum of the maximum singular value of the Jacobian matrix, i.e., we want to substitute J for A . However, J is not only a function of the decision variables, but also of the configuration of the manipulator, that is, $J = J(\alpha; x, y)$. From Section 3.2 we know that the manipulability matrix $M = JJ^\top$ is a linear function of the decision variables α and a polynomial function of the configuration variables x, y . This suggests a way to modify (3.6) to determine the maximum of the maximum singular value of the Jacobian matrix as follows

$$\begin{aligned}
& \underset{(\alpha, s) \in \mathbb{R}^{n+1}}{\text{minimize}} && s \\
& \text{subject to} && sI - M(\alpha; x, y) \succeq 0.
\end{aligned} \tag{3.7}$$

While determining the nonnegative definiteness of the matrix $sI - M(\alpha; x, y)$ over all (x, y) is an unwieldy problem, that of determining whether it is an SoS matrix is not, by virtue of Lemma 2.1.4 of Section 2.1 on page 12. It states that replacing the \succeq condition in the constraint by a SoS condition turns this optimization to a semidefinite program.

A similar procedure can be applied to maximize the minimum of the minimum singular value of the Jacobian over the decision variables by negating the constraint and objective function in (3.7). We can then find the global maximizer of GII by solving the following semidefinite optimization problem

$$\begin{aligned}
& \underset{(\alpha, s, t) \in \mathbb{R}^{n+2}}{\text{minimize}} && s - t \\
& \text{subject to} && sI - M(\alpha; x, y) \text{ is SoS} \\
& && M(\alpha; x, y) - tI \text{ is SoS.}
\end{aligned} \tag{3.8}$$

To incorporate the maximization of GDI within the semidefinite program we create in the next subsection, we recall the discussion at the beginning of Section 3.2.2. We notice that a natural linear function of the decision variables, whose minimization will also minimize $\bar{\sigma}(H)$, thus maximizing GDI , is given by the trace of H .

Remark 3.4.1 *Since $\text{tr}(H) = \sum_i \lambda_i(H)$, minimizing the trace of H is equivalent to minimizing the maximum value of the sum of the eigenvalues of H over the workspace.*

3.4.2 Formulating the constraints

The semidefinite program (3.8) is not yet a faithful optimization problem because it lacks the circle constraints $x_i^2 + y_i^2 = 1$, $\forall i = 1, 2$, and the workspace constraints, G , presented in Section 3.3.

The circle constraints, the elbow-out posture constraints, G_1 , G_2 and nonsingularity constraints G_3 and G_4 of G are polynomial constraints, which can be embedded in SoS programming by invoking the procedure described in Lemma 2.1.2. The last two elements of G define affine constraints on the decision variables and can be directly inserted into the final semidefinite program (3.9).

3.4.3 Multicriteria Optimization

Once the single criteria optimization problems have been cast as semidefinite programs, we combine the two using the weighted-sum approach to generate the Pareto-front. This involves solving the following semidefinite program for each value of the parameter $0 \leq \gamma \leq 1$.

$$\begin{aligned}
 & \underset{(\alpha, s, t) \in \mathbb{R}^{n+2}}{\text{minimize}} && \gamma c^\top \alpha + (1 - \gamma)(s - t) \\
 & \text{subject to} && sI - M(\alpha; x, y) \text{ is SoS} \\
 & && M(\alpha; x, y) - tI \text{ is SoS} \\
 & && G(\alpha; x, y) \leq 0 \\
 & && \alpha_l \leq \alpha \leq \alpha_u \\
 & && x_i^2 + y_i^2 = 1, \quad \forall i = 1, 2.
 \end{aligned} \tag{3.9}$$

3.5 Optimization Results

Table 3.3 presents the results of the SoS optimization algorithm for the single objective problems, for best kinematic ($\gamma = 0$) and best dynamic isotropy ($\gamma = 1$),

Table 3.3: Results of independent optimizations with respect to GII and GDI

	Best Design for Kinematic Isotropy	Best Design for Dynamic Isotropy	Unit
GII	0.57735	0.28021	–
GDI	0.89354	0.95012	–
l_1	0.22361	0.10000	m
l_2	0.22361	0.30000	m

respectively.

To characterize the trade-off between the single objective solutions, a Pareto-front curve for the bi-objective optimization problem is constructed in Figure 3.2. Schematics of the mechanism at the two extremes, i.e., best kinematic and dynamic isotropy, and at an approximately equal-trade-off-point are shown in the subplots within Figure 3.2. It turns out that whenever $\gamma > 0.5$, there is very little trade-off between *GII* and *GDI*. Cubically spacing γ within $[0, 0.5]$ results in a relatively uniform sampling of the Pareto-front curve. Note that any point on this curve is a non-dominated solution. The designer is free to choose to “best” design parameters respective to his or her requirements.

For performance comparisons with regards to computational demands, the multi-objective problem with 101 distinct values of γ took approximately **2 minutes** to complete using a single *mobile* Intel i7 – 7700HQ processor operating at 3.80GHz. In comparison, when the culling algorithm was used to obtain globally optimal results for a symmetric five-bar parallel mechanism in [2], it took the authors approximately **16 hours** to find 30 distinct values on the Pareto-front on a 3.40GHz Intel Xeon processor in parallel using 8-cores. Thanks to the lack of discretization requirement in SoS methods, both computation time and the solution accuracy improve. Moreover, the method scales to higher DoF manipulators in polynomial time as opposed to

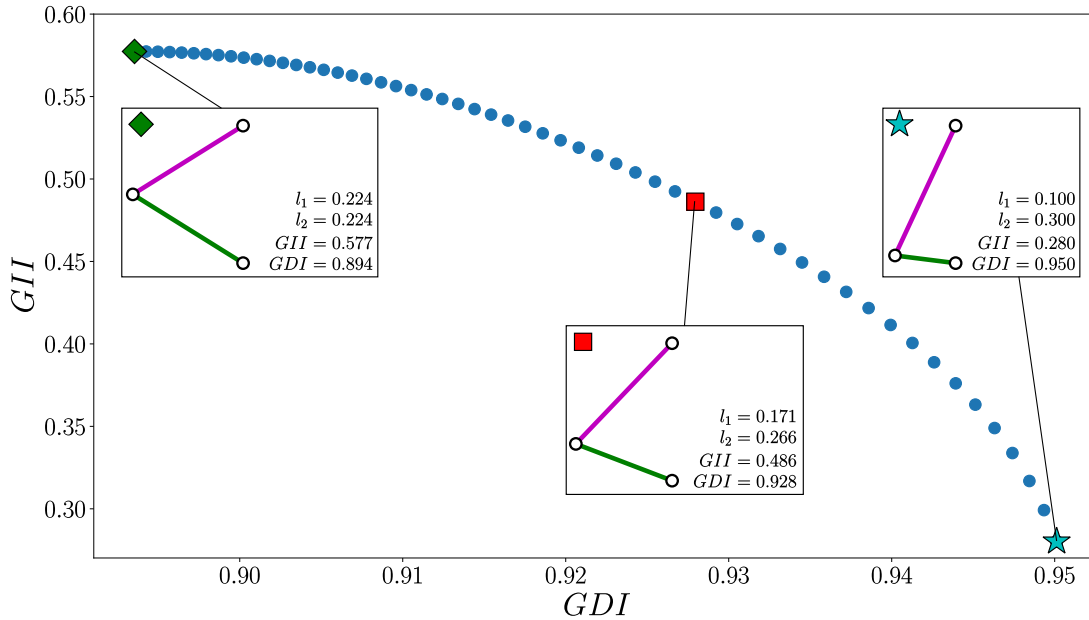


Figure 3.2: Pareto-front curve generated via sums-of-squares optimization
 Subplots: Comparisons of results for best GII (diamond), nontrivial trade-off between GII and GDI (square), and best GDI (star).

exponential. Since the symmetric five-bar linkages can be viewed as a two-link serial manipulator with an additional constraint at the end-effector, the sums-of-squares programming algorithm should result in orders of magnitude improvement in computation time. These results serve as a strong indication that the application of framework presented in this case study to more complex mechanisms promises substantial improvements in accuracy and computational performance.

3.6 Conclusions

This chapter presents a formulation of the design optimization of robotic manipulators, which was previously solved via nonlinear optimization techniques, as a convex optimization problem by employing sums-of-squares optimization. As a result, we are able to find the globally optimal solution for the single objective

problems with arbitrarily high precision (up to machine precision). Interior-point methods are used to obtain the numerical solution to a SoS optimization problem very efficiently. They scale exceptionally well with increasing number of decision variables in contrast to other methods, such as branch-and-bound algorithms, which are able to find the globally optimal solution. In future work, the same technique can be used to optimally design parallel mechanisms, which additionally require the satisfaction of loop equations.

CHAPTER 4

ORBITAL STABILIZATION OF A TORSO-ACTUATED RIMLESS WHEEL

This chapter addresses the development of control systems for a rimless wheel with torso. The organization of this chapter is as follows: Section 4.1 presents the modeling of the robot as a hybrid dynamical system. Section 4.2 brings together the ideas from Section 2.2 and Section 2.3 to design two types of control systems for the robot. This section also presents a method to estimate the regions of attraction of the two controllers using Sum-of-Squares programming techniques. Section 4.3 presents and compares the simulation results and the estimated regions of attraction of the two controllers. Section 4.4 provides some experimental results that support the theory developed in this chapter.

4.1 System Model

The robot, depicted in Figure 4.1, consists of two parts, first of which is the classical rimless wheel and the second of which is a torso attached to the rimless wheel at its center. The rimless wheel consists of $n = 10$ massless spokes, each of length l_1 . The rimless wheel has mass m_1 , which is concentrated at the hip. A point mass m_2 is attached to the hip through a massless rod of length l_2 and constitutes the torso. We place a motor between the torso and the hip and actuate the variable

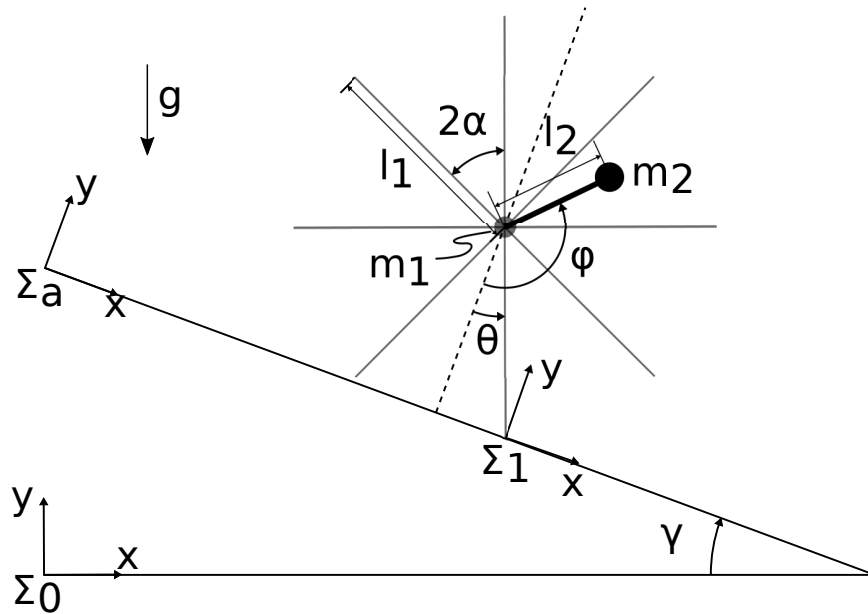


Figure 4.1: Schematic of rimless wheel with torso

This is depicted with $n = 8$ spokes to prevent clutter. The angle φ is controlled. The origin of Σ_1 is placed at the point of contact between the ground and the spoke.

φ , defined as the angle between the normal vector that penetrates the terrain and the vector that emanates from the hip and is directed to the point mass m_2 . The position of the hip is characterized by the angle θ , which is defined between the normal vector that penetrates the terrain and the vector that emanates from the hip and is directed to the point of contact. A downhill terrain is characterized by a ramp slope, $\gamma > 0$. Without loss of generality, we assume that the wheel walks from left-to-right, that is, for walking we must have $\dot{\theta}(t) < 0$ for all $t > 0$. Finally, we denote half the angle between two spokes by $\alpha = \frac{\pi}{n}$.

The rimless wheel undergoes phases of continuous flows and discrete transitions, resulting in a hybrid dynamical system with two modes. Its configuration space is $\mathcal{Q} = \mathbb{T}^2$, the two-torus. The state space is then the tangent bundle of the configuration space, denoted by $T\mathcal{Q}$.

4.1.1 Swing model – continuous dynamics

We derive the continuous dynamics of the system via the Lagrangian approach. The kinetic and potential energies of the system are given by

$$\mathcal{K} = \frac{1}{2} \sum_{i=1}^2 m_i v_i \cdot v_i + \frac{1}{2} \sum_{i=1}^2 \omega_i \cdot I_i \omega_i,$$

$$\mathcal{P} = \sum_{i=1}^2 m_i g p_{0,i} \cdot e_2,$$

where $\{v_i\}_1^2$ are the linear velocities of the hip, and the tip of the torso; $\{\omega_i\}_1^2$ are the angular velocities of the wheel and the torso with respect to and expressed in the inertial frame; $\{p_{0,i}\}_1^2$ are the position vectors from the origin of the inertial frame to the centers of mass of the wheel and the torso, $\{m_i\}_1^2$ and $\{I_i\}_1^2$ are masses and the moments of inertia of the wheel and the torso about their centers of masses, respectively. The magnitude of the gravitational acceleration is denoted by g while its opposite direction is given by $-e_2$. In coordinates (θ, φ) (see Figure 4.1), the Lagrangian is given by

$$\begin{aligned} \mathcal{L} &= \mathcal{K} - \mathcal{P} \\ &= \frac{1}{2} (I_1 + m_t l_1^2) \dot{\theta}^2 - m_2 l_1 l_2 c_{\theta\varphi} \dot{\theta} \dot{\varphi} + \frac{1}{2} (I_2 + m_2 l_2^2) \dot{\varphi}^2 - m_t g l_1 c_{\theta\gamma} + m_2 g l_2 c_{\varphi\gamma}, \end{aligned}$$

where $m_t = m_1 + m_2$, $c_{ab} := \cos(a - b)$ and $s_{ab} := \sin(a - b)$, with a and b taking values in $\{\theta, \varphi, \gamma\}$. Defining $q = (\theta, \varphi)$ and writing the Euler-Lagrange equations corresponding to this Lagrangian yields the classical robot dynamics

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = Bu, \quad (4.1)$$

where $B = \begin{bmatrix} -1 & 1 \end{bmatrix}^\top$ and

$$\begin{aligned}
M(q) &= \begin{bmatrix} I_1 + m_t l_1^2 & -m_2 l_1 l_2 c_{\theta\varphi} \\ -m_2 l_1 l_2 c_{\theta\varphi} & I_2 + m_2 l_2^2 \end{bmatrix}, \\
C(q, \dot{q}) &= m_2 l_1 l_2 s_{\theta\varphi} \begin{bmatrix} 0 & -\dot{\varphi} \\ \dot{\theta} & 0 \end{bmatrix}, \\
G(q) &= g \begin{bmatrix} -m_t l_1 s_{\theta\gamma} \\ m_2 l_2 s_{\varphi\gamma} \end{bmatrix},
\end{aligned}$$

Adding together each equation in (4.1) and setting $\dot{\varphi} \equiv 0$ in this equation yields what we will call the *reduced equations of motion*.

$$\tilde{m}(q)\ddot{\theta} + \tilde{c}(q, \dot{\theta})\dot{\theta} + \tilde{g}(q) = 0, \quad (4.2)$$

where

$$\begin{aligned}
\tilde{m}(q) &= I_1 + m_t l_1^2 - m_2 l_1 l_2 c_{\theta\varphi}, \\
\tilde{c}(q, \dot{\theta}) &= m_2 l_1 l_2 s_{\theta\varphi} \dot{\theta}, \\
\tilde{g}(q) &= -m_t g l_1 s_{\theta\gamma} + m_2 g l_2 s_{\varphi\gamma}.
\end{aligned}$$

This reduced model, with configuration space $\mathcal{Q} = \mathbb{S}^1$, where \mathbb{S}^1 stands for the circle as a topological space, neglects the transient response of the torso, whose angle φ is viewed as the control input.

4.1.2 Impact model – discrete transition

At heelstrike, the impulsive reaction force from the ground is applied at the end of the swing leg. The angular momentum of the whole system about the swing foot is not affected by this reaction force [56], implying a conservation law relating the pre-impact and post-impact velocities of the system. Similarly, the impact force affects the torso through the hip. Therefore, the angular momentum of the torso about this point is conserved through the heelstrike. Combining these two observations yields the

impact map for the system. Representing the state of the system in coordinates by $x := (q, \dot{q}) = (\theta, \varphi, \dot{\theta}, \dot{\varphi})$, the impact map, $\Delta : T\mathcal{Q} \rightarrow T\mathcal{Q}$, is given by

$$x^+ = \Delta(x^-) = \Xi(q)x^-, \quad (4.3)$$

where $\mathbb{R}^{4 \times 4} \ni \Xi(q) = \text{diag}\{\Xi_p, \Xi_v\}$ has a block diagonal structure with $\Xi_p = \text{diag}\{-1, 1\}$ and

$$\begin{aligned} \Xi_v(q) &= \begin{bmatrix} \xi_1(\varphi) & 0 \\ \xi_2(\varphi) & 1 \end{bmatrix}, \\ |\Xi_v^+| &= I_1 I_2 + I_1 m_2 l_2^2 + I_2 m_t l_1^2 \\ &\quad + m_2 l_1^2 l_2^2 (m_1 + m_2 \sin(\alpha - \varphi)^2), \\ |\Xi_v^+| \xi_1(\varphi) &= I_1 I_2 + I_1 m_2 l_2^2 + \left[I_2 m_t l_1^2 + m_2 l_1^2 l_2^2 \left(m_1 + \frac{m_2}{2} \right) \right] \cos(2\alpha) - \frac{1}{2} m_2^2 l_1^2 l_2^2 \cos(2\varphi), \\ |\Xi_v^+| \xi_2(\varphi) &= m_2 l_1 l_2 \left[I_1 (\cos(\alpha - \varphi) - \cos(\alpha + \varphi)) \right. \\ &\quad \left. + m_t l_1^2 (\cos(2\alpha) \cos(\alpha - \varphi) - \cos(\alpha + \varphi)) \right]. \end{aligned}$$

For the reduced model, we again set $\dot{\varphi} \equiv 0$, which reduces these equation to yield the *reduced impact map*, $\Xi_r(q) = \text{diag}\{-1, \xi_r(q)\}$, where

$$\begin{bmatrix} \theta^+ \\ \dot{\theta}^+ \end{bmatrix} = \Xi_r(q) \begin{bmatrix} \theta^- \\ \dot{\theta}^- \end{bmatrix}, \quad (4.4a)$$

$$\xi_r(q) = \frac{I_1 + m_t l_1^2 \cos(2\alpha) - m_2 l_1 l_2 \cos(\alpha + \varphi)}{I_1 + m_t l_1^2 - m_2 l_1 l_2 \cos(\alpha - \varphi)}. \quad (4.4b)$$

4.1.3 Hybrid model

The overall model of walking is obtained by combining the swing phase model and the impact model to form a system with impulse effects. For the full model with $x = (q, \dot{q})$, we have

$$\Sigma : \begin{cases} \dot{x} = f(x, u) & \text{if } x^- \notin \mathcal{S} \\ x^+ = \Delta(x^-) & \text{if } x^- \in \mathcal{S}, \end{cases} \quad (4.5)$$

where the switching set is chosen to be

$$\mathcal{S} := \{x \in T\mathcal{Q} : p_2^v(\theta) = 0, p_2^h(\theta) > 0\}.$$

Here p_2^h and p_2^v denote the x - and y - position of the swing leg with respect to the auxiliary frame, Σ_a , (see Figure 4.1). In the equations of motion (4.5), the flow vector field f is taken from equation (4.1). The impact map is given by $\Delta(x) = \Xi(q)x$.

For a preliminary controller design, it will be convenient to use the reduced model, where we view the torso angle φ as the control input and reduce the states to $x = (\theta, \dot{\theta})$. In this case, the flow vector field in (4.5) is taken from equation (4.2) and the reduced impact map is given by $\Delta(x) = \Xi_r(q)x$.

4.2 Control Design

In this section, we used the techniques described in Section 2.2 and Section 2.3 to present two novel controllers: an inherently discrete-time and an inherently continuous-time, derived from the reduced model and stabilized on the full system. We describe how to estimate the regions of attraction of these controllers and present an analysis that leads to further performance improvements. An overview of the control design procedure is given in Algorithm 1.

4.2.1 Discrete-Time (DT) Controller

For this type of controller, we seek to set the torso angle at a particular Poincaré section that we choose to be $\mathcal{S} = \{x \in \mathbb{S}^1 \times \mathbb{R} : \theta = 0\}$. The torso angle is then kept constant throughout the swing phase.

Algorithm 1: Controller Design Process

Input: States of the robot in the reduced model, $x = (\theta, \dot{\theta})$
Output: Desired torso angles φ_{ref} , stabilizing a nominal walking gait of the reduced model

- 1 $x^* \leftarrow$ Find a nominal limit cycle
- 2 **if** *Discrete-time controller* **then** do
 - 3 Define a single Poincaré section along x^*
 - 4 $P \leftarrow$ approximation of the Poincaré map
 - 5 $K \leftarrow$ discrete LQR controller that stabilizes the fixed point of P
 - 6 $\varphi_{\text{ref}} \leftarrow \varphi_0 + K(\dot{\theta} - \dot{\theta}^*)$
- 7 **else if** *Continuous-time controller* **then** do
 - 8 Define a *family* of Poincaré sections along x^* , each parametrized by τ
 - 9 Define a transverse coordinate x_{\perp}
 - 10 Transform original coordinates to (x_{\perp}, τ) coordinates
 - 11 Compute the nonlinear dynamics of the transverse system
 - 12 Linearize the transverse dynamics
 - 13 $k(\theta) \leftarrow$ LQR controller to stabilize the x_{\perp} system
 - 14 $\varphi_{\text{ref}} \leftarrow \varphi_0 + k(\theta)(\dot{\theta} - \dot{\theta}^*)$
- 15 **return** φ_{ref}

The set polynomial functions forms a basis for the space of continuous functions from \mathbb{R} to \mathbb{R} . We use this fact to approximate the Poincaré map by a polynomial of the form

$$P(y, u) = \sum_{\eta_1 + \eta_2 \leq m} \alpha_{\eta} y^{\eta_1} u^{\eta_2}, \quad (4.6)$$

where $\eta = (\eta_1, \eta_2)$ is a multi-index, that is, a 2-tuple of non-negative integers, whose sum is less than or equal to m , the degree of the approximating polynomial. Note that $P : U \subseteq \mathcal{S} \rightarrow \mathcal{S}$ is a map from a subset of the Poincaré section to the Poincaré section. Here $y = \dot{\theta} - \dot{\theta}^*$ is the coordinate along the Poincaré section with $\dot{\theta}^*$ denoting the nominal wheel velocity along a nominal walking gait, and u is the torso angle, φ .

We invoke the least squares estimation technique to estimate the coefficients, $\{\alpha_{\eta}\}$, of this polynomial. The estimation is performed by generating target values, $\dot{\theta}_{\text{target}}$ on the Poincaré section, by simulating forward in time a range of initial values

$0 < \dot{\theta} < -\pi$ and $\gamma < \varphi < \frac{\pi}{2} + \gamma$. We remove from data the failed simulations due to stumbling or running. These target values are then compared with the output of the approximation of the Poincaré map and the coefficients that minimize the discrepancy are selected. In other words, $\{\alpha_\eta\}$ are found by solving the following minimization problem:

$$\underset{\alpha_\eta}{\text{minimize}} \quad \|P(\dot{\theta}, \varphi) - \dot{\theta}_{\text{target}}\|^2.$$

We use a linear approximation to derive the discrete-time controller and test the controller against a second-order approximation to avoid potential overfitting. This linear approximation of the Poincaré map allows for the use of the mature theory of linear systems, and here we choose to utilize a discrete-time LQR controller for the discrete linear system described by

$$y_{k+1} = P(y_k, u_k) = \alpha_{1,0}y_k + \alpha_{0,1}u_k. \quad (4.7)$$

4.2.2 Continuous-Time (CT) Controller

Following the methods described Sections 2.2.3 and [57, 37], we present the derivation of a continuous-time (CT) controller for the reduced system. In this case, instead of sticking to a particular Poincaré section, we choose a family of Poincaré sections, one for each parameter $\tau \in \mathbb{R}$, along a periodic orbit of the system. As θ is a monotonically decreasing function of time along a periodic orbit, we use this state variable to parametrize the Poincaré sections, i.e. $t = \tau(\theta)$, where t denotes time. Using this parametrization, we have the following family of Poincaré sections

$$\mathcal{S}(\theta) = \{x \in \mathbb{S}^1 \times \mathbb{R} : \theta = c, \text{ for some } c \in [-\alpha, \alpha]\}.$$

This family of Poincaré sections can also be described as a family of hyperplanes defined as follows

$$\mathcal{S}(\theta) = \{y \in \mathbb{R}^2 : z^\top (y - x^*(\theta)) = 0\},$$

where $z = \begin{bmatrix} -1 & 0 \end{bmatrix}^\top$ and $x^*(\theta)$ is a periodic orbit, i.e., $x^*(t) = x^*(t + T)$ with $\dot{x}^*(t) \neq 0, \forall t \in [0, T)$. We construct a coordinate system on $\mathcal{S}(\theta)$ by choosing the basis on this subspace as $\begin{bmatrix} 0 & -1 \end{bmatrix}^\top$. A projection operator $\Pi(\theta)$ onto this space is then defined. as $\Pi(\theta) = \begin{bmatrix} 0 & 1 \end{bmatrix}$. This construction defines a change of coordinates $x \mapsto (x_\perp, \tau)$, where τ represents which of the transversal surfaces $\mathcal{S}(\tau)$ the current state x inhabits, and the vector x_\perp is the “transversal” state representing the location of x within the hyperplane $\mathcal{S}(\tau)$, with $x_\perp = 0$ implying that $x = x^*(\tau)$. The transverse coordinate under this construction is determined by $x_\perp = \Pi(y - x^*(\tau))$ for any $y \in \mathcal{S}(\tau)$.

The transverse dynamics for the rimless wheel with torso is given by

$$\dot{x}_\perp = \Pi f(x^*(\tau) + \Pi^\top x_\perp) - \Pi f(x^*(\tau)) \dot{\tau}, \quad (4.8a)$$

$$\dot{\tau} = \frac{z^\top f(x^*(\tau) + \Pi^\top x_\perp)}{z^\top f(x^*(\tau))}. \quad (4.8b)$$

To derive the CT controller, we first compute the linearization of the transverse dynamics (4.8).

$$\dot{\delta x}_\perp = a(t)\delta x_\perp + b(t)\delta u, \quad (4.9)$$

where

$$a(t) = \Pi \frac{\partial f}{\partial x}(x^*(t)) \Pi^\top - \Pi f(x^*(t)) \frac{\partial \dot{\tau}}{\partial x_\perp} \Big|_{x_\perp=0}, \quad (4.10a)$$

$$b(t) = \Pi \frac{\partial f}{\partial u}(x^*, u^*). \quad (4.10b)$$

For evaluating the right-hand sides of these expressions, first we select a limit cycle using the methods discussed in Section 4.2.5. The limit cycle thus provided is then interpolated to be a function of time and is inserted into equation (4.10).

We follow this procedure up by solving the differential Riccati equation corresponding to this linear time-varying (LTV) system and come up with an optimal gain such that $\delta u(t) = k(t)\delta x_{\perp}(t)$. We apply this feedback control as a function of the phase variable θ rather than time t , by setting $k(t) = k \circ \tau(\theta)$. This renders the overall closed-loop system autonomous rather than time-varying, which simplifies further stability analyses.

4.2.3 Implementation of the Controllers

The controllers we have derived so far find desired torso angles, which would stabilize a nominal walking gait for the reduced model. The DT controller yields a torso angle to be kept constant for each step, whereas the CT controller yields a time-varying, continuous reference torso angle. We will call both reference signals φ_{ref} henceforth.

In this short subsection, we describe how to make the torso angle φ exponentially track φ_{ref} . For this purpose, we employ the now-standard method of partial feedback linearization [58] on the equations (4.1). We solve the first equation in (4.1) for $\ddot{\theta}$ and substitute it into the second to get the following equation

$$\begin{aligned}\Delta_{22}\ddot{\varphi} + h(q, \dot{q}) &= bu, \\ h(q, \dot{q}) &= c_{21}\dot{\theta} - m_{12}^{\top}m_{11}^{-1}c_{12}\dot{\varphi} + g_2 - m_{12}^{\top}m_{11}^{-1}g_1,\end{aligned}$$

where Δ_{22} is the Schur complement of the block m_{22} of the matrix M , and $b = 1 + m_{12}^{\top}m_{11}^{-1}$. Expanding the expression for b , one finds out that it vanishes nowhere. Thus, we can set

$$u = \frac{1}{b} (h - \Delta_{22} (k_p(\varphi - \varphi_{ref}) + k_d(\dot{\varphi} - \dot{\varphi}_{ref}))), \quad (4.11)$$

which is well-defined on all of the state-space. Here k_p and k_d are positive control gains. Since $\varphi_{ref} = \varphi_0 + k(\theta)(\dot{\theta} - \dot{\theta}^*)$, this implies on the nominal limit cycle, $\varphi = \varphi_0$.

Combined with Proposition 2, which is given in Section 4.2.5 on page 50, this implies that the perturbation to the reduced dynamics of the motion of the torso is a vanishing perturbation that is Lipschitz in a neighborhood of $x_{\perp} = 0$. Thus by Lemma 9.1 of [51], this implies that the full system is locally exponentially stable to the nominal limit cycle.

4.2.4 Region of Attraction Estimation

We have devised two controllers for the rimless wheel with torso. Both of these controllers are derived from the linearization of a relevant dynamical quantity. It is then of interest to quantify the regions of attraction of these controllers. We solve this estimation problem by casting it as the following sum-of-squares program.

$$\begin{aligned} & \text{maximize } \rho \\ & \text{subject to } (V(y) - \rho) - \lambda(y) dV(y) \text{ is SoS,} \\ & \lambda(y) \text{ is SoS,} \end{aligned} \tag{4.12}$$

where $y = \dot{\theta} - \dot{\theta}^*$ and the term SoS implies a sum-of-squares constraint [59]. We pick the Lyapunov function as $V(y) = py^2$. In the discrete-time case, p is a positive constant and $dV(y)$ is the change in V between successive visits to the Poincaré section. In the continuous-time case, p is a positive function of the angle θ , $dV(y)$ is the Lie derivative of $V(y)$ along the trajectories of the system (4.2). The quantity p is obtained by solving the corresponding Riccati equation. This equation is either associated with a second-order estimation of the Poincaré map (4.6) or the continuous flow (4.2). In the continuous-time case, we discretize the variable along the limit-cycle, and solve a family of sum-of-squares programs to generate the values of $p(\theta)$. The

solution of optimization problem (4.12) gives an inner estimation of the region of attraction as $\Omega_r = \left\{ x \in T\mathcal{Q}_r : |y| < \sqrt{\frac{p}{p}} \right\}$.

4.2.5 Impact Map Analysis and Controller Improvements

The control laws studied so far perform reasonably well; however, the states typically jump away from the desired manifold $\mathcal{M} = \{\varphi - \varphi_{ref} = 0\}$ at the heelstrike. A considerable control effort must then be expended so that φ approaches φ_{ref} again. To remedy this situation, we analyze the impact map further to devise a family of methods to modify φ_{ref} in a desirable manner.

From the form of the impact map Ξ in equation (4.3), we observe that the post-impact torso velocity $\dot{\varphi}^+$ equals its pre-impact value, $\dot{\varphi}^-$, plus the term $\xi_2(\varphi)\dot{\theta}^-$. This suggests a robust method that precludes the impact event from disturbing the convergence $\varphi \rightarrow \varphi_{ref}$:

1. Find φ^* such that $\xi_2(\varphi^*) = 0$, $\varphi^* \in [\gamma, \frac{\pi}{2} + \gamma]$
2. Find a nominal limit cycle x^* so that $\varphi_{ref} = \varphi^*$,
3. Compute the transverse dynamics and the accompanying time-varying LQR closed-loop controller,
4. Have φ track φ_{ref} by using the control law (4.11).

In fact, we can take this observation one step further and find a family of methods that will perform favorably to a naïve nominal limit cycle selection. To that end, consider replacing point (1) above by the following steps:

- (1a) Pick a function $f(\Xi)$ of the impact map to optimize, e.g., no change in post-impact torso velocity, least decrease in post-impact wheel velocity, etc.,

(1b) Perform the optimization to find the optimizer φ^* .

The search for nominal limit cycle in step (2) may be conducted by using a nonlinear solver to find the fixed point of the Poincaré map $P(\cdot, \varphi^*) : \mathbb{R} \rightarrow \mathbb{R}$. The values of the Poincaré map are obtained by numerically simulating the reduced system forward in time for a single step.

If the optimization problem in (1b) is unconstrained, then there may exist no nominal limit cycle for the optimizer φ^* (i.e., the rimless wheel stumbles and falls back on the pre-impact stance spoke). In this case, the optimization can be repeated with an increased lower bound for φ until such a nominal limit cycle found. If any nominal limit cycle exists, this procedure will guarantee an optimal solution thanks to the following two propositions, which jointly show that increasing φ will increase the walking speed $|\dot{\theta}|$ whenever the speed is small.

Proposition 1 *For the reduced system (4.2), $\dot{\theta}$ is a decreasing function of φ for small $|\dot{\theta}|$ and $\gamma \leq \varphi \leq \frac{\pi}{2} + \gamma$.*

Proof. Straightforward calculation shows that

$$\frac{d}{dt} \frac{\partial}{\partial \dot{\theta}} \left(\frac{1}{2} \tilde{m}(q) \dot{\theta}^2 \right) = \tilde{m}(q) \ddot{\theta} + \tilde{c}(q, \dot{\theta}) \dot{\theta} = -\tilde{g}(q).$$

Integrating both sides over time yields

$$\tilde{m}(q(t)) \dot{\theta}(t) = m_t g l_1 \int_0^t \sin(\theta(\sigma) - \gamma) d\sigma - m_2 g l_2 \int_0^t \sin(\varphi - \gamma) d\sigma.$$

Differentiating both sides with respect to φ we find

$$\frac{\partial \dot{\theta}}{\partial \varphi} = -\frac{m_2 l_2}{\tilde{m}(q)} \left(g \cos(\varphi - \gamma) t + l_1 \sin(\varphi - \theta) \dot{\theta} \right).$$

Since the coefficient on the right hand side is negative and the continuous flow takes a finite amount of time ($t > 0$), we deduce that $\dot{\theta}$ is a decreasing function of φ as long as $|\dot{\theta}|$ is bounded and small.

The above proposition makes sense as long as we can prove that $\dot{\theta}$ is bounded. This is true with a mild assumption on φ .

Proposition 2 *The speed $|\dot{\theta}|$ of the rimless wheel under the reduced model (4.2) is finite as long as $\varphi > \alpha$.*

Proof. Consider the Lyapunov function candidate

$$V(\theta, \dot{\theta}) = \frac{1}{2} \tilde{m}(q) \dot{\theta}^2 + m_t g l_1 (1 + \cos(\theta - \gamma)).$$

Taking the Lie derivative along the solutions of (4.2) and manipulating gives

$$\dot{V} = m_2 l_2 \dot{\theta} \left(\frac{l_1}{2} \sin(\varphi - \theta) \dot{\theta}^2 - g \sin(\varphi - \gamma) \right).$$

Assuming $\dot{\theta} < 0$, $\dot{V} < 0$ whenever $\frac{l_1}{2} \sin(\varphi - \theta) \dot{\theta}^2 \geq g \sin(\varphi - \gamma)$. For sufficiently large $|\dot{\theta}|$, this inequality is satisfied. Since the impact map Δ contracts $|\dot{\theta}|$ as well, we conclude that $|\dot{\theta}|$ is finite.

As an example of the above procedure, consider the problem of stabilizing a walking gait that requires the least energy to maintain. In the absence of friction, the system loses energy through the impact events. In particular, observe that $\Xi_2(q; \alpha) \rightarrow I$ as $\alpha \rightarrow 0$, which implies a continuum of spokes, i.e., a circular (rimmed) wheel. When $\alpha > 0$, on the other hand, it is always the case that $\mathcal{H}(q, \dot{q}) > \mathcal{H} \circ \Delta(q, \dot{q})$, where \mathcal{H} denotes the total energy of the system. It follows that the closer Ξ_2 is to the identity map, the smaller is the energy loss of the walking gait. Therefore, we pose and solve the following optimization problem to come up with a desired nominal torso angle

$$\begin{aligned} & \underset{\varphi}{\text{minimize}} \quad \|I - \Xi_2(\varphi)\|_F^2 \\ & \text{subject to} \quad \gamma < \varphi < \frac{\pi}{2} + \gamma, \end{aligned} \tag{4.13}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. For this particular system, this objective function can be expressed as $\|I - \Xi_2(\varphi)\|_F^2 = (1 - \xi_1(\varphi))^2 + \xi_2^2(\varphi)$.

Remark 4.2.1 *Note that Ξ_2 also depends on the inertial parameters and the link lengths. Although we do not pursue it here, one can also cast the energetically-optimal walking problem into a mechanism design problem. One would seek to choose the spoke length and the torso length such that $\|I - \Xi_2(\varphi)\|_F^2$ is uniformly minimized.*

Remark 4.2.2 *Extensive simulation studies show that even if the desired nominal gait is not selected such that the nominal torso reference angle equals φ^* , where φ^* is the angle that minimizes energy loss at impact, it is energetically more efficient to smoothly deform the reference from φ_{ref} to φ^* as the walking gait nears heelstrike (use any smooth homotopy from the function $\varphi_{ref} : T\mathcal{Q} \rightarrow \mathbb{S}^1$ to the constant signal $\varphi^* : T\mathcal{Q} \rightarrow *$).*

4.3 Simulation Studies

We present and compare simulation results for the controllers we discussed in the previous Section 4.2. We obtained the simulation results using a custom software written in the programming language Julia. The parameters we use for the simulations are selected to match an actual implementation of the robot and are summarized in Table 4.1.

4.3.1 Discrete-Time (DT) Controller

To derive the DT controller, we first obtain the approximation of the Poincaré map as described in Section 4.2.1. With the system parameters given in Table 4.1, the coefficients, α_η of the Poincaré map are presented in Table 4.2. We then find the nominal limit cycle by solving the optimization problem described in Section 4.2.5.

Table 4.1: Parameters of the system

(l_1, l_2) [m]	(m_1, m_2) [kg]	(I_1, I_2) [kg · m ²]
(0.26, 0.05)	(2.32, 4.194)	(0.0784160, 0.0380256)

Table 4.2: Coefficients of Poincaré map approximation

$\alpha_{0,0}$	$\alpha_{1,0}$	$\alpha_{0,1}$	$\alpha_{2,0}$	$\alpha_{0,2}$	$\alpha_{1,1}$
-0.842	0.498	-1.813	-0.0567	0.623	-0.183

The angular velocity of the wheel at $\theta = 0$ on the nominal limit cycle is computed to be $\dot{\theta}^* = -0.65 \left[\frac{\text{rad}}{\text{s}} \right]$. The corresponding nominal torso angle is $\varphi_0 = 18.26^\circ$.

For the computed nominal limit cycle, the linear portion of the estimated Poincaré map yields a DLQR controller gain of $K = -0.212$ [s]. The reference torso angle each time the reduced system hits this section is given by $\varphi_{ref} = \varphi_0 + K(\dot{\theta} - \dot{\theta}^*)$.

The torso position so generated is taken as a reference, which is asymptotically stabilized for the full system as described in Section 4.2.3. To generate the simulation results shown in Figure 4.3, we used the initial condition $x_{\text{initial}} = (\theta, \varphi, \dot{\theta}, \dot{\varphi}) = (0, \frac{\pi}{2} + \gamma, \frac{1}{2}, 0)$, where the wheel starts moving in the reverse direction. The response of the system under the DT controller is shown by the dotted, diamond, red curves.

4.3.2 Continuous-Time (CT) Controller

The implementation of CT controller parallels that of the DT controller. We still derive a nominal limit cycle but this time interpolate it as a function of time and subsequently the wheel angle θ . We use the same nominal limit cycle characterized by $\dot{\theta}^* = -0.65 \left[\frac{\text{rad}}{\text{s}} \right]$ at $\theta = 0$ with $\varphi_0 = 18.26^\circ$.

Using a numerical interpolation of the nominal limit cycle, we derive the transverse linearization, given by the functions $a(t)$ and $b(t)$ in equation (4.9). Figure 4.2 shows

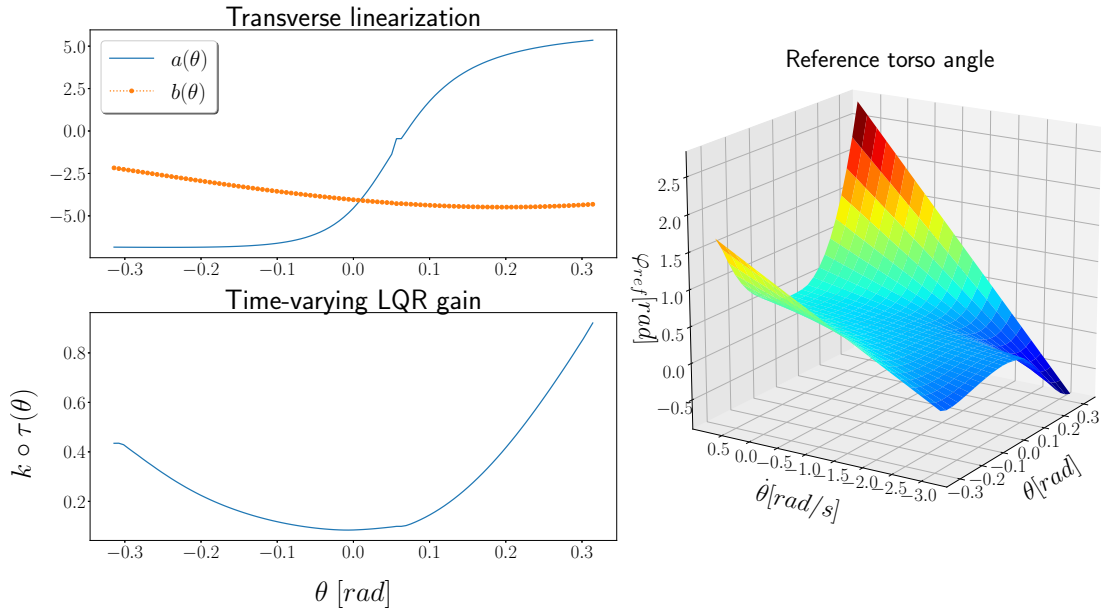


Figure 4.2: Continuous-time Controller Computation

Top left: Transverse state and input functions $a(\theta)$, $b(\theta)$. Bottom left: the optimal control gain $k(\theta)$. Right: the reference torso angle, which is computed by $\varphi_{ref} = \varphi_0 + k(\theta)(\dot{\theta} - \dot{\theta}^*)$.

the transverse linearization, the time-varying LQR gain k and the corresponding reference torso angle φ_{ref} , expressed as functions of θ and $\dot{\theta}$.

The full system uses the inverse dynamics controller (Section 4.2.3) to exponentially steer φ to this φ_{ref} . For ease of comparison with the DT controller, we start the system at exactly the same initial condition even though the CT controller affords to be started at an even more adverse initial condition (such as a larger positive initial wheel velocity, see Section 4.3.3). The behavior of the full system dynamics under the CT controller is shown in Figure 4.3 as the solid, triangle, blue curves.

We can use the CT controller to smoothly stabilize on terrains with varying slopes as well. The animation for such a simulation may be viewed at https://youtu.be/mRcX6S_rvFE.

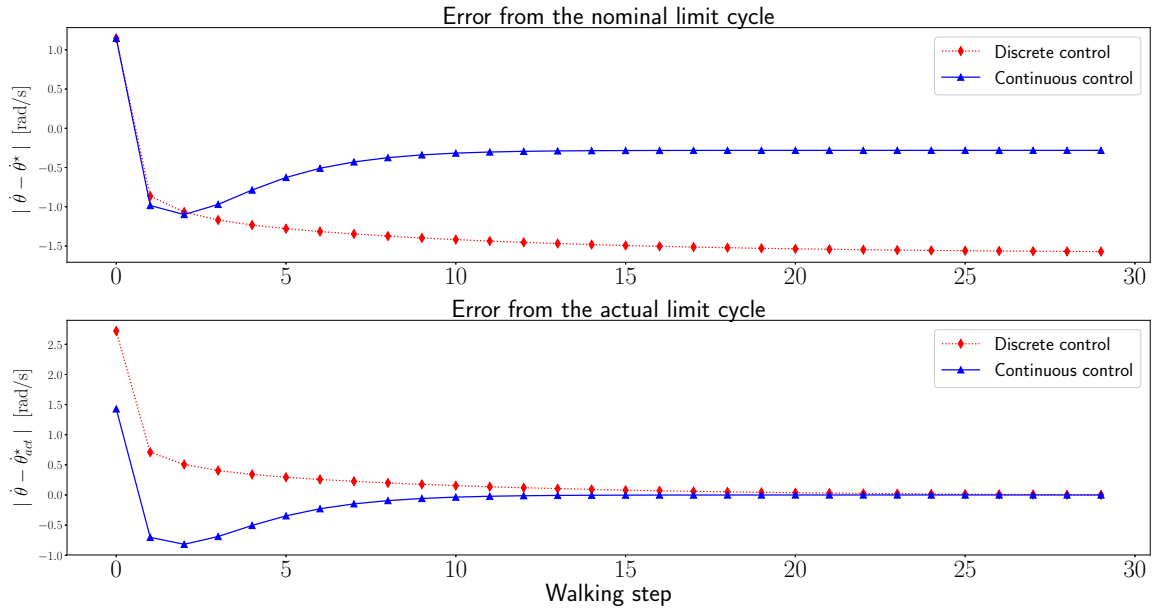


Figure 4.3: Convergence of the DT and CT controllers
 Top: convergence to the nominal. Bottom: convergence to actual limit cycles.

4.3.3 Comparison of the CT and DT Controllers

The top plot in Figure 4.3 shows the convergence of each controller when applied to the full system. The results reveal that the CT controller tracks the nominal limit cycle that was derived from the reduced model much more closely. The DT controller converges to a completely different limit cycle, which walks much faster than the nominal limit cycle.

The bottom plot in Figure 4.3 shows the transient response of the controllers as they converge to their respective actual walking gaits. We observe that the CT controller outperforms the DT controller with a faster convergence.

Figure 4.4 depicts the inner approximations of the regions of attraction for the controllers. The red diamond shows the estimated region of attraction of the DT controller, which is only computed at the Poincaré section characterized by $\theta = 0$.

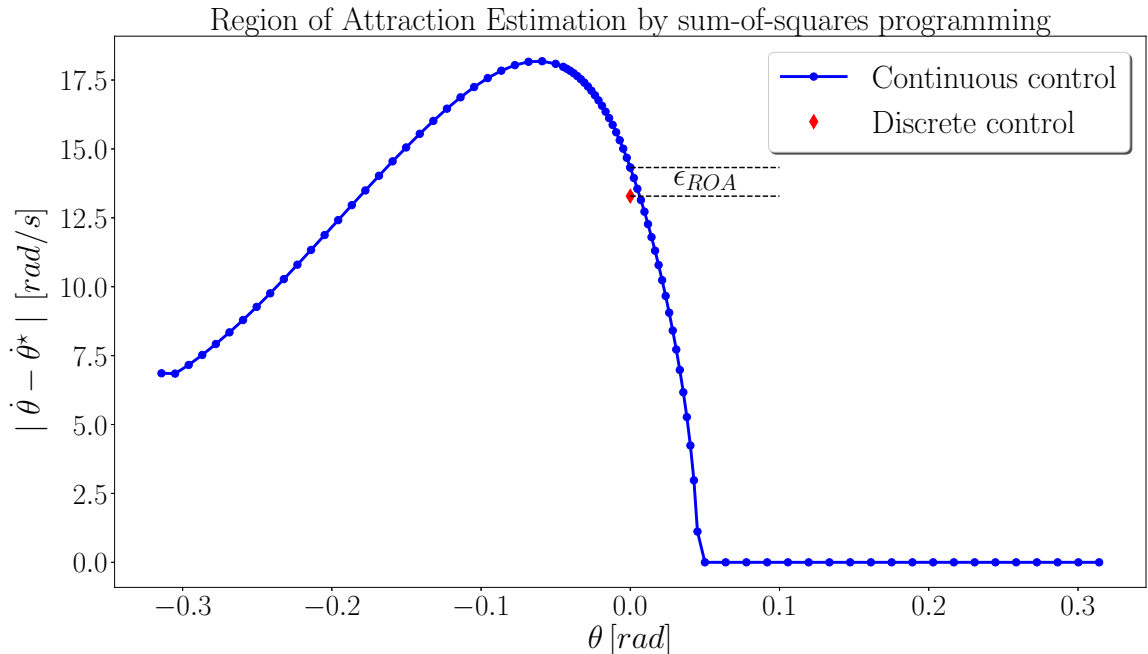


Figure 4.4: Regions of attraction of the DT and CT controllers
 $\epsilon_{ROA} = 1.038$ [rad/s]

The region of attraction of the CT controller is given for various Poincaré sections, $-\alpha \leq \theta \leq \alpha$. We observe that on the section $\theta = 0$, the region of attraction of the CT controller is greater than that of the DT controller by an amount $\epsilon_{ROA} = 1.038$ [rad/s]. We also observe that the optimization results correctly predicts that the region of attraction vanishes for the region $\theta \geq 0.05$ [rad]. In words, it is predicting that if the system is started with a wheel velocity that is too great in the opposite direction, the robot will stumble and fall back on its swing foot.

4.4 Experimental Studies

In a collaboration with Professor Bhounsule [34] and his team, we describe the hardware implementation of the rimless wheel with torso and provide experimental

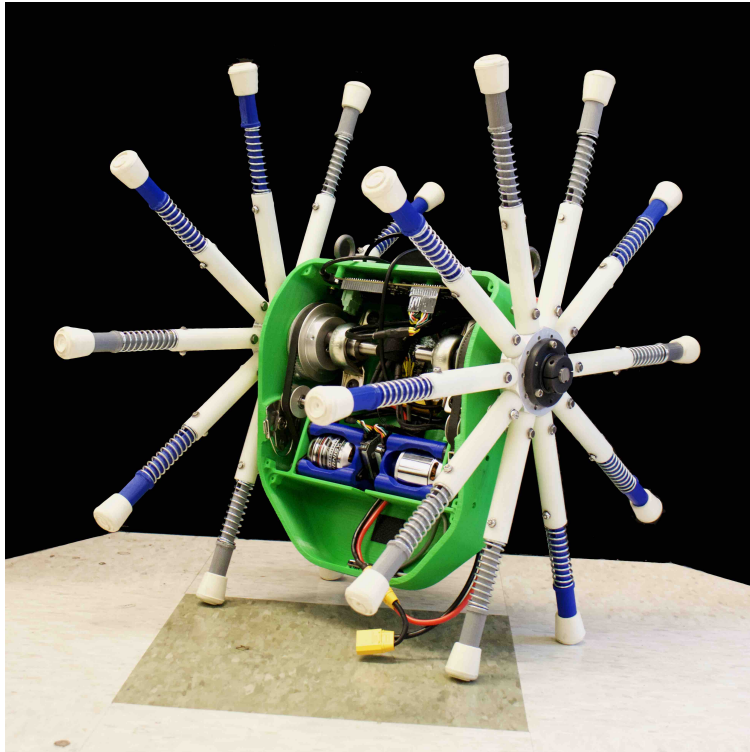


Figure 4.5: Experimental prototype

results that support the theory developed earlier.

4.4.1 Prototype

Figure 4.5 shows the experimental prototype. It consists of two rimless wheels, each with 10 spokes, attached side-to-side through a torso that houses all the electronics, motors, and batteries. The spokes and the torso are 3D printed using a hobby-grade printer (Makerbot Replicator). Each spoke of the wheel has an inline compression spring to cushion the collision. Each rimless wheel is connected to an outrunner motor (Turnigy Aerodrive SK3 5055-280 KV, Hobbyking, Hong Kong) through a belt system. Each motor has a capacitive encoder (AMT102 8192 counts per revolution, CUI Inc., Tualatin, OR, USA) and controlled by an Odrive v3.5 motor

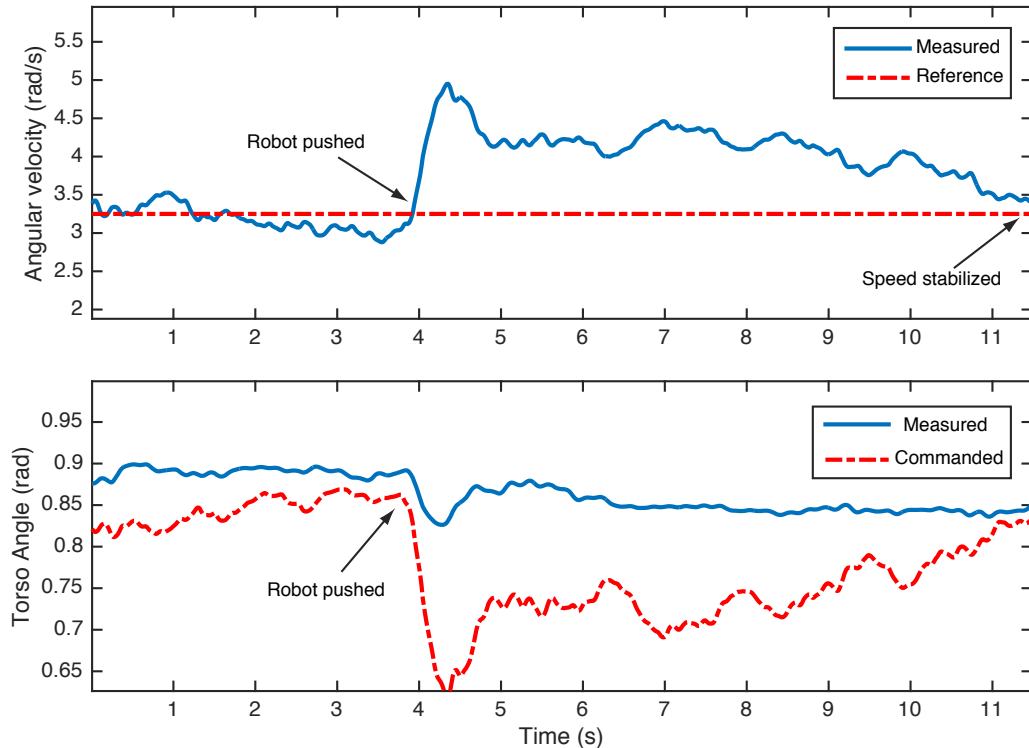


Figure 4.6: Experimental results

controller (Odrive, CA, USA). The Odrive is connected to a Raspberry Pi 3B (Raspberry Pi foundation, UK). In addition, the torso is connected to a 9-axis inertial measurement unit (Adafruit, NY, USA). The motors are powered by a Turnigy 3000mAh 6S 30C Lipo Pack w/XT-60 and the electronics and computers are powered by Turnigy 1300mAh 6S 35C Lipo Pack.

4.4.2 Results

We implemented the continuous-time controller on the hardware with constant k gain. We chose a nominal torso angle of $\varphi_0 = 0.873$ [rad] (50°). This corresponds to a nominal linear hip speed of $0.8[\frac{\text{m}}{\text{s}}]$ or an angular speed of $\dot{\theta}^* = 3.25[\frac{\text{m}}{\text{s}}]$ at $\theta = 0$. Figure 4.6 (a) shows the angular velocity of the stance leg ($\dot{\theta}$), and (b) shows

torso angle (φ), both as a function of time. The robot is manually pushed at $t = 4$ sec, thus increasing its mid-stance speed to 5 rad/s. The controller compensates by reducing the torso angle. It takes the controller about 7 seconds or about 50 steps to get the system back to the nominal limit cycle, The slow response is because the proportional-integral-derivative controller for the torso is conservatively tuned and takes appreciable time to change the set-point.

4.5 Conclusion

This chapter presents a family of controllers for robustly stabilizing walking gaits of a rimless wheel with torso on various ground slopes. These controllers are derived by asymptotically stabilizing the transverse dynamics of the reduced system along a nominal limit cycle, judiciously selected to minimize the energy loss at impacts. Several other sensible choices for the nominal limit cycles exist, such as one for which the pre-impact and post-impact angular velocity of the torso coincide, etc. We then support the theoretical development by numerical simulations and an eventual implementation on hardware.

CHAPTER 5

CONCLUSIONS

This thesis explores the problem of design optimization for robotic manipulators subject to multiple performance metrics using convex optimization techniques. We have designed and validated two types of feedback controllers that stabilize selected gaits of a walking robot through a methodology that scales well to robots with higher degrees of freedom.

For the design optimization problem, we are able to efficiently find the geometric parameters for the two-link serial manipulator that are globally-optimal in terms of kinematic and dynamic isotropy. One of the remaining challenges is to formulate the constraints so that the framework can be used with parallel manipulators, for which the loop equations must be satisfied. Furthermore, this framework readily allows for the determination of inner and outer approximations to the singularity-free workspace of robotic manipulators. These two tasks will be pursued in future work.

For the work concerning orbital stabilization of the rimless wheel with torso, we show theoretically and through extensive simulation results that the control policies are able to stabilize the full dynamics of the system. In the future, we intend to add to the experimental results that we have presented in Chapter 4.

REFERENCES

- [1] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, “Efficient bipedal robots based on passive-dynamic walkers,” *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.
- [2] R. Unal, G. Kiziltas, and V. Patoglu, “A multi-criteria design optimization framework for haptic interfaces,” in *2008 Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2008, pp. 231–238.
- [3] M. A. Ergin, A. C. Satici, and V. Patoglu, “Design optimization, impedance control and characterization of a modified delta robot,” in *2011 IEEE International Conference on Mechatronics (ICM)*. IEEE, 2011, pp. 737–742.
- [4] L. Stocco, S. E. Salcudean, and F. Sassani, “Fast constrained global minimax optimization of robot parameters,” *Robotica* 16(6), pp. 595–605, 1998.
- [5] J. W. Yoon, J. Ryu, and Y.-K. Hwang, “Optimum design of 6-dof parallel manipulator with translational/rotational workspaces for haptic device application,” *Journal of mechanical science and technology*, vol. 24, no. 5, pp. 1151–1162, 2010.
- [6] S.-D. Stan, V. Maties, R. Balan, C. Rusu, and S. Besoiu, “Optimal link design of a six degree of freedom micro parallel robot based on workspace analysis,” in *2008 10th IEEE International Workshop on Advanced Motion Control*. IEEE, 2008, pp. 637–642.
- [7] S.-U. Lee and S. Kim, “Analysis and optimal design of a new 6 dof parallel type haptic device,” in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 460–465.
- [8] Y. Hou and Y. Zhao, “Workspace analysis and optimization of 3-puu parallel mechanism in medicine base on genetic algorithm,” *The open biomedical engineering journal*, vol. 9, p. 214, 2015.
- [9] M. H. Saadatzi, M. T. Masouleh, H. D. Taghirad, C. Gosselin, and P. Cardou, “On the optimum design of 3-rpr parallel mechanisms,” in *2011 19th Iranian Conference on Electrical Engineering*. IEEE, 2011, pp. 1–6.

- [10] X.-J. Liu and J. Wang, "A new methodology for optimal kinematic design of parallel mechanisms," *Mechanism and machine theory*, vol. 42, no. 9, pp. 1210–1224, 2007.
- [11] X.-J. Liu, J. Wang, and H.-J. Zheng, "Optimum design of the 5r symmetrical parallel manipulator with a surrounded and good-condition workspace," *Robotics and Autonomous Systems*, vol. 54, no. 3, pp. 221–233, 2006.
- [12] Y. Lou, G. Liu, and Z. Li, "Randomized optimal design of parallel manipulators," *IEEE Transactions on Automation Science and Engineering*, vol. 5, no. 2, pp. 223–233, 2008.
- [13] R. E. Stamper, L.-W. Tsai, and G. C. Walsh, "Optimization of a three dof translational platform for well-conditioned workspace," in *Proceedings 1997 IEEE International Conference on Robotics and Automation*, vol. 4. IEEE, 1997, pp. 3250–3255.
- [14] H.-p. LIN and S.-q. HOU, "Optimization analysis of workspace for the 3-puu parallel machine tool based on monte carlo method," *Modular Machine Tool & Automatic Manufacturing Technique*, vol. 9, p. 024, 2009.
- [15] Y. Lou, G. Liu, J. Xu, and Z. Li, "A general approach for optimal kinematic design of parallel manipulators," in *Proceedings 2004 IEEE International Conference on Robotics and Automation*, vol. 4. IEEE, 2004, pp. 3659–3664.
- [16] M. A. Mousavi, M. T. Masouleh, and A. Karimi, "On the maximal singularity-free ellipse of planar 3-rpr parallel mechanisms via convex optimization," *Robotics and Computer-Integrated Manufacturing*, vol. 30, no. 2, pp. 218–227, 2014.
- [17] V. Hayward, J. Choksi, G. Lanvin, and C. Ramstein, "Design and multi-objective optimization of a linkage for a haptic interface," in *Advances in robot kinematics and computational geometry*. Springer, 1994, pp. 359–368.
- [18] G. Alici and B. Shirinzadeh, "Optimum synthesis of planar parallel manipulators based on kinematic isotropy and force balancing," *Robotica*, vol. 22, no. 1, pp. 97–108, 2004.
- [19] M. Kreff and J. Hesselbach, "Elastodynamic optimization of parallel kinematics," in *IEEE International Conference on Automation Science and Engineering, 2005*. IEEE, 2005, pp. 357–362.
- [20] A. Frisoli, M. Prisco, F. Salsedo, and M. Bergamasco, "A two degrees-of-freedom planar haptic interface with high kinematic isotropy," in *8th IEEE International Workshop on Robot and Human Interaction. RO-MAN'99 (Cat. No. 99TH8483)*. IEEE, 1999, pp. 297–302.

- [21] W. Chen, Q. Zhang, Z. Zhao, and W. Gruver, “Optimizing multiple performance criteria in redundant manipulators by subtask-priority control,” in *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century.*, vol. 3. IEEE, 1995, pp. 2534–2539.
- [22] S. McGhee, T. F. Chan, R. V. Dubey, and R. L. Kress, “Probability-based weighting of performance criteria for a redundant manipulator,” in *1994 IEEE International Conference on Robotics and Automation*. IEEE, 1994, pp. 1887–1894.
- [23] J. H. Lee, K. S. Eom, and I. I. Suh, “Design of a new 6-dof parallel haptic device,” in *Proceedings 2001 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1. IEEE, 2001, pp. 886–891.
- [24] J. Yoon and J. Ryu, “Design, fabrication, and evaluation of a new haptic device using a parallel mechanism,” *IEEE/ASME Transactions on mechatronics*, vol. 6, no. 3, pp. 221–233, 2001.
- [25] G. Blekherman, P. A. Parrilo, and R. R. Thomas, *Semidefinite optimization and convex algebraic geometry*. SIAM, 2012.
- [26] W. Sirichotiyakul, V. Patoglu, and A. C. Satici, “Convex multi-criteria design optimization of robotic manipulators via sum-of-squares programming,” in *3rd IEEE International Conference on Robotic Computing, IRC*. IEEE, 2019, pp. 439–440.
- [27] T. McGeer, “Passive dynamic walking,” *The International Journal of Robotics Research*, vol. 9, no. 2, pp. 62–82, 1990.
- [28] A. Chatterjee and M. Garcia, “Small slope implies low speed for mcgeer’s passive walking machines,” *Dynamics and Stability of Systems*, vol. 15, no. 2, pp. 139–157, 2000.
- [29] M. Gomes and A. Ruina, “Walking model with no energy cost,” *Physical Review E*, vol. 83, no. 3, p. 032901, 2011.
- [30] J. B. Jeans and D. Hong, “Impass: Intelligent mobility platform with active spoke system,” in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 1605–1606.
- [31] F. Asano and J. Kawamoto, “Passive dynamic walking of viscoelastic-legged rimless wheel,” in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2331–2336.

- [32] F. Asano, T. Sogawa, K. Tamura, and Y. Akutsu, “Passive dynamic walking of rimless wheel with 2-dof wobbling mass,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 3120–3125.
- [33] F. Asano and X. Xiao, “Output deadbeat control approaches to fast convergent gait generation of underactuated spoked walker,” in *2012 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2012, pp. 265–270.
- [34] P. A. Bhounsule, E. Ameperosa, S. Miller, K. Seay, and R. Ulep, “Dead-beat control of walking for a torso-actuated rimless wheel using an event-based, discrete, linear controller,” in *Proc. 40th mechanisms and robotics conference*. ASME, 2016.
- [35] T. A. Henzinger, “The theory of hybrid automata,” in *Verification of Digital and Hybrid Systems*. Springer, 2000, pp. 265–292.
- [36] R. Goebel, J. Hespanha, A. R. Teel, C. Cai, and R. Sanfelice, “Hybrid systems: generalized solutions and robust stability,” in *Proc. 6th IFAC symposium in nonlinear control systems*. Citeseer, 2004, pp. 1–12.
- [37] I. R. Manchester, “Transverse dynamics and regions of stability for nonlinear hybrid limit cycles,” *arXiv preprint arXiv:1010.2241*, 2010.
- [38] G. A. Leonov, “Generalization of the andronov-vitt theorem,” *Regular and chaotic dynamics*, vol. 11, no. 2, pp. 281–289, 2006.
- [39] A. S. 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*, vol. 32, no. 2, pp. 200–211, 2008.
- [40] A. Prestel and C. Delzell, *Positive polynomials: from Hilberts 17th problem to real algebra*. Springer Science & Business Media, 2013.
- [41] D. Henrion and A. Garulli, *Positive polynomials in control*. Springer Science & Business Media, 2005, vol. 312.
- [42] P. A. Parrilo, “Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization,” Ph.D. dissertation, California Institute of Technology, 2000.
- [43] M.-D. Choi, T. Y. Lam, and B. Reznick, “Sums of squares of real polynomials,” in *Proceedings of Symposia in Pure mathematics*, vol. 58. American Mathematical Society, 1995, pp. 103–126.

- [44] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. Siam, 1994, vol. 15.
- [45] I. Dunning, J. Huchette, and M. Lubin, “Jump: A modeling language for mathematical optimization,” *SIAM Review*, vol. 59, no. 2, pp. 295–320, 2017.
- [46] L. Perko, *Differential Equations and Dynamical Systems*. Springer Science & Business Media, 2013, vol. 7.
- [47] S. Strogatz, M. Friedman, A. J. Mallinckrodt, and S. McKay, “Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering,” *Computers in Physics*, vol. 8, no. 5, pp. 532–532, 1994.
- [48] S. Sastry, *Nonlinear systems: analysis, stability, and control*. Springer Science & Business Media, 2013, vol. 10.
- [49] J. Hauser and C. C. Chung, “Converse lyapunov functions for exponentially stable periodic orbits,” *Systems & Control Letters*, vol. 23, no. 1, pp. 27–34, 1994.
- [50] R. Tedrake, “Underactuated robotics: Algorithms for walking, running, swimming, flying, and manipulation,” Course Notes for MIT 6.832, August 2018. [Online]. Available: <http://underactuated.mit.edu>
- [51] H. Khalil, *Nonlinear Systems*, ser. Pearson Education. Prentice Hall, 2002.
- [52] J. K. Salisbury and J. Craig, “Articulated hands: Force control and kinematic issues,” *The International Journal of Robotics Research*, vol. 1(1), pp. 4–17, 1982.
- [53] M. W. Spong, S. Hutchinson, M. Vidyasagar, *et al.*, *Robot modeling and control*. Wiley New York, 2006, vol. 3.
- [54] E. de Klerk, *Aspects of Semidefinite Programming: Interior Point Algorithms and Selected Applications*, ser. Applied Optimization. Springer US, 2006.
- [55] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [56] E. R. Westervelt, C. Chevallereau, J. H. Choi, B. Morris, and J. W. Grizzle, *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2007.
- [57] A. S. Shiriaev and L. B. Freidovich, “Transverse linearization for impulsive mechanical systems with one passive link,” *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2882–2888, 2009.

- [58] M. W. Spong, “Partial feedback linearization of underactuated mechanical systems,” in *International Conference on Intelligent Robots and Systems (IROS)*, vol. 1. IEEE, 1994, pp. 314–321.
- [59] W. Tan, A. Packard, *et al.*, “Stability region analysis using polynomial and composite polynomial lyapunov functions and sum-of-squares programming,” *IEEE Transactions on Automatic Control*, vol. 53, no. 2, p. 565, 2008.