DYNAMIC MODELING AND CONTROL OF UNDERACTUATED PLANAR BIPEDAL WALKING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SAİT SOVUKLUK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JUNE 2022

Approval of the thesis:

DYNAMIC MODELING AND CONTROL OF UNDERACTUATED PLANAR BIPEDAL WALKING

submitted by **SAİT SOVUKLUK** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İlkay Ulusoy
Head of Department, **Electrical and Electronics Engineering** _____

Assoc. Prof. Dr. Mustafa Mert Ankaralı
Supervisor, **Electrical and Electronics Engineering, METU** _____

Prof. Dr. Uluç Saranlı
Co-supervisor, **Computer Engineering, METU** _____

Examining Committee Members:

Prof. Dr. Klaus Werner Schmidt
Electrical and Electronics Engineering, METU _____

Assoc. Prof. Dr. Mustafa Mert Ankaralı
Electrical and Electronics Engineering, METU _____

Prof. Dr. Uluç Saranlı
Computer Engineering, METU _____

Prof. Dr. Yiğit Yazıcıoğlu
Mechanical Engineering, METU _____

Assist. Prof. Dr. Yıldıray Yıldız
Mechanical Engineering, Bilkent University _____

Date: 16.06.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Sait Sovukluk

Signature :

ABSTRACT

DYNAMIC MODELING AND CONTROL OF UNDERACTUATED PLANAR BIPEDAL WALKING

Sovukluk, Sait
M.S., Department of Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. Mustafa Mert Ankaralı
Co-Supervisor: Prof. Dr. Uluç Saranlı

June 2022, 82 pages

This study demonstrates an adaptive model predictive control method for input constrained control of underactuated bipedal walking with a predefined trajectory. Our approach aims to increase the trajectory tracking performance of the system and produce realistic and applicable responses while letting a certain amount of posture change around the predefined trajectory. To do so, we employ whole-body dynamics in our control structure, include weights for the unactuated joint inside the cost function, and define input torque constraints in the solution. Obeying input torque limits decreases modeling and estimation errors, such that trajectory tracking becomes more robust and efficient. Additionally, we test this model-based controller successfully against various disturbances such as high magnitude modeling errors in its weight, significant initial condition errors, pushing and pulling the torso aggressively throughout a step, high percentage input noises, and their combinations. These disturbances are usually introduced in the torso because it is the heaviest, the longest, and the unactuated joint. Thanks to its short-horizon requirement, the controller is suitable for implementation in real-time and in 1kHz frequency, which is usually re-

quired to control high-dimensional underactuated nonlinear hybrid systems.

Keywords: adaptive model predictive control, underactuated bipedal walking, bipedal locomotion, robotic walking

ÖZ

EKSİK TAHRİKLİ DÜZLEMSEL İKİ BACAKLI YÜRÜMENİN DİNAMİK MODELLENMESİ VE KONTROLÜ

Sovukluk, Sait
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Mustafa Mert Ankaralı
Ortak Tez Yöneticisi: Prof. Dr. Uluç Saranlı

Haziran 2022 , 82 sayfa

Bu çalışma eksik tahrikli düzlemsel robotik yürümenin daha önce belirlenmiş bir yörünge etrafındaki giriş kısıtlı kontrülünün adaptif model öngürülü kontrol ile takip edilmesini incelemektedir. Çalışmadaki amacımız robotik yürüme sisteminin belirlenen yörüngeyi takip etme performasının arttırılmasını amaçlamaktadır. Bu performans artışı sağlanırken sistemin gerçekçi ve deneysel düzeneklerde tekrarlanabilir hareketler yapmasına özen gösterilmiştir. Gerekli durumlarda belirlenen yörünge etrafında küçük duruş değişikliklerine izin verilerek kontrol sırasında gerekli olan sistem girdisi miktarı azaltılmıştır. Kontrol sırasında giriş kısıtlarının göz önünde bulundurulması modelleme ve kestirim hatalarının azaltılmasına olanak sağlamış ve yörünge takibini daha verimli ve gürbüz bir hale getirmiştir. Ek olarak, bahsi geçen bu model tabanlı kontrol gövde kütlesinin arttırılması, gövdeye dışarıdan itme ve çekme gibi kuvvetlerin uygulanması, sistem girdilerinin hatalı ve gürültülü hale getirilmesi gibi çeşitli belirsizlik ve hatalar karşısında başarılı bir şekilde test edilmiştir. Gövde uzuvunun en ağır, en uzun ve eksik tahrikli olması sebebi ile genelde modelleme hataları

ve belirsizlikler bu uzuvda tanımlanmıştır. Kısa ufuk gereksinimi sayesinde, bahsi geçen bu kontrolcü gerçek zamanlı olarak 1kHz frekansında uygulanmıştır.

Anahtar Kelimeler: adaptif model kestirimli kontrol, eksik tahrikli iki bacaklı yürüme, robotik yürüme

To my family

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisors, Mustafa Mert Ankaralı and Uluç Saranlı, who has supported and guided me through my graduate studies.

I am grateful to ATLAS Laboratory supervisors Uluç Saranlı, Afşar Saranlı, Yiğit Yazıcıoğlu, and Mert Ankaralı, who kindly accepted me to their laboratory when I was an English preparatory school student just before I was a freshman, in 2015. Even though I was a newly graduated high school student, they let me participate in their projects. Alongside the projects I worked on, the laboratory environment broadened my horizon tremendously. Being friends with many graduate students and attending weekly meetings with them and professors enabled me to have ideas in various research fields related to robotics and helped me find out what I wanted to study.

I would like to thank my colleagues from the laboratory, Güner Dilşad Er, Emre Tanfener, and Furkan Davulcu, for their insightful opinions, support, and contributions to my studies. Thanks to their existence, working in the laboratory was always joyful.

Additionally, I would like to thank my colleagues and seniors at Teknolus Energy for welcoming me into their company throughout my master's education. Dealing with real-life engineering problems and developing mechatronic devices enlarged my horizon considerably. The experience I gained there helped me relate academic life to the industry, and I have learned a lot there, especially from my supervisor Burçak Özkök.

Last but not least, I would like to express my gratitude to my family for their joyful support and contribution. They always believed in me and stood with me. During this process, it was a blessing to have them with me.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

2D	2 Dimensional
3D	3 Dimensional
DOF	Degree of Freedom
HZD	Hybrid Zero Dynamics
MPC	Model Predictive Control
PD	Proportional-Derivative
Hz	Hertz
CoM	Center of Mass

CHAPTER 1

INTRODUCTION

1.1 Motivation and Background

Bipedal robotics is a subclass of legged robots. It is usually based on mimicking human-like locomotion such as walking and running. Alongside the joy this field provides to many roboticists, the main motivations behind these robots are to replace humans in hazardous environments (as simulated in DARPA Robotics Challange, which addresses humanoid robotics usage in natural and man-made disasters [1, 2, 3, 4, 5]), to rehabilitate people (for example people with paraplegia [6, 7]) in an exoskeleton form, and to develop active prosthetic leg to improve amputee gait [8, 9, 10, 11].

Unlike other robot walkers, the complexity of control of bipedal walkers with point feet originates from their underactuated nature. The underactuation requires an indirect control of unactuated joints through high-bandwidth control of actuated joints. The actuated joints follow a specially generated trajectory such that, based on the system dynamics, the unactuated joint performs the desired behavior. The combination of underactuation with high-dimensional nonlinear hybrid dynamics requires special treatment for their control. Due to the high dimensionality of the system and several gait characterization constraints, one usually needs to solve these trajectory generation optimization problems offline for one step. Some of these trajectory generation methods are Hybrid Zero Dynamics (HZD) [12], Human-Inspired Control [13], and direct trajectory optimization [14]. Online implementation of the predefined optimized trajectory employs a calculated restriction input with a local high-gain controller.

Hybrid zero dynamics (HZD), which is the method that is employed in this study,

is based on setting specially selected holonomic output functions to zero. Setting the output to zero restricts the walker's dynamics into a lower-dimensional attractive and invariant subset of its state space such that an analytically provable reliable and dynamically-stable walking is achieved. Using the underactuated planar bipedal walker RABBIT, it is shown that the method is successful in generating robust walking motions [15, 16]. Additionally, for the running with the compliance, compliant hybrid zero dynamics notion is introduced in [17], such that actuator limitation problems introduced in [18] are resolved using compliant energy storage elements. Then one degree of underactuation notion of HZD is extended for multi degree of underactuation, and walking and running with compliance is successfully implemented to a robot MABEL [19, 20, 21]. The last pioneer studies related to HZD was the extension of this method for three-dimensional (3D) walking and its realization on humanoid robots. Related models, control approaches, and realization challenges are described in [22, 23] in detail. Even though it has proven that 3D robot walking is realizable on a small-scale humanoid robot, NAO [24, 25], since the complexity of the full-order system dynamics is increased even more in three-dimensional walking (or running), reducing the gap between theory and experimental realization became an even more difficult task for full-scale humanoid robots. HZD uses entire system dynamics to generate walking gaits in the form of a constrained nonlinear optimization problem. Even though the resultant solution ends up with efficient and dynamic gaits, as the complexity increases, the difficulty of solving the problem increases along with it. Exploiting the structure of the nonlinear optimization problem allowed rapid gait generation and reduced optimization time from hours to a few minutes [26].

Locomotion on unknown stochastic terrains requires a combination of different gaits. Depending on surface and stability conditions, the next foot placement point may need the step to be longer, shorter, or suitable for different elevations. The literature employs previously calculated gait libraries with dedicated switching methods to walk over randomly generated stepping stones [27], reject velocity disturbances efficiently [28], walk over an uneven terrain [29] and parametrize bipedal robot locomotion [30]. Even though optimization problems for trajectory generation consider various gait restrictions and torque limitations, the online implementation employed in these studies uses a calculated restriction input with a local high-gain PD con-

troller, which is enough to prove exponential stability in theory. Unless one tests the trajectories against every possible source of disturbances and uncertainties, these online implementation methods tend to exceed the input limitations of the actual system due to high-gain PD and similar local controllers. Due to the complex nature of underactuated bipedal locomotion, exceeding system input limitations are usually not affordable. Exceeding torque limitations of the system causes discrepancies between the system model and the robot, errors in estimations, and instabilities or slow convergence rates. Additionally, excessive torque input usually causes a violation in the other gait restrictions, such as friction cone.

Studies addressing the input torque limitation problem in the online implementation of walking gaits usually employ modified model-based controllers using simplified or linearized dynamics. The studies in [31, 32] solve this problem employing a modified MPC around linearized closed-loop restriction dynamics, [33] employs Hybrid-Linear Inverted Pendulum (H-LIP) dynamics for MPC implementation, [34] employs simplified model of the robot dynamics projected in swing foot space for their control, [35] uses linear centroidal dynamics and [36] uses a first-order system obtained by capture point dynamics. The problem with linear system assumptions is that the accuracy of these models is valid only for a small region of configuration variables. Additionally, even input noises of the motors affect the correctness of linearized closed-loop restriction dynamics whose correction depends on a particular input. On unknown stochastic terrain, unavoidable sources of errors reduce the performance of these online implementation methods.

This study proposes an adaptive MPC implementation method with input constraints on sampled whole-body dynamics as an online trajectory tracking controller for underactuated bipedal walkers. The control method isolates commonly applied feedback linearization inputs from trajectory tracking dynamics and reduces modeling errors. However, the structure still allows the implementation of these feedback linearization inputs before generating the cost function. Additionally, the introduced cost function form includes weights for unactuated joints. As a result, the controller performs considerable posture changes around the stabilizable desired orbits such that it produces realistic and applicable responses against various disturbances while obeying input limitations. Thanks to its short-horizon requirement, the controller can

be implemented in real-time at 1kHz, which is crucial for high-dimensional underactuated nonlinear hybrid systems. We show the controller's ability against various aggressive disturbances where local high-gain PD trajectory tracking controllers failed to maintain stability under input torque saturation. We also show that this model-based controller is not too sensitive against modeling errors.

1.2 The Outline of the Thesis

Chapter II includes system dynamics formulation for five-link underactuated planar bipedal walkers. In the chapter, first, the definition of walking is discussed. Then, taking the definition as a basis, consecutive system dynamics are derived.

Chapter III discusses the definition of zero dynamics and hybrid zero dynamics notions. These methods are employed to generate periodic, stable walking gaits. In the chapter, implementation details of these methods are discussed, along with detailed simulation results.

Chapter IV covers an adaptive model predictive controller implementation for such walkers. The method is used to track the previously generated walking gait in Chapter III while constraining input and allowing some degree of posture changes.

Chapter V focuses on the adaptive power optimal control method for such walking systems. Again, the method is used to track the previously generated walking pattern in Chapter III. Differently, it includes the system's power in its cost function such that while the input torque is constrained, the input power is minimized.

Chapter VI compares controller performances that are introduced in the previous three chapters. This chapter uses the controllers' basin of attractions and Poincaré return map estimations as comparison metrics.

Chapter VII discusses different aspects of the methods introduced and implemented during the study, along with their advantages and drawbacks.

Appendix A includes the equation of motion of a planar five-link bipedal walker with point feet termination. Contents of the inertia and Coriolis matrices and the gravity

vector are provided.

Appendix B includes eigenvalue estimations of the linearized step-to-step dynamics of the walker. These metrics are used to compare the local stabilities of the controllers.

CHAPTER 2

SYSTEM DYNAMICS

Walking is a combination of two consecutive events called stance and impact (2.1), where x^- and x^+ represent states just before and just after the impact, respectively, Δ represents impact dynamics, and $\mathcal{S}$ represents impact surface. Stance dynamics is a nonlinear whole-body Lagrangian model that captures the state evolution of the system while the stance leg touches the ground and the swing leg moves in the air. The impact, $\Delta : x^- \rightarrow x^+$, on the other hand, is a momentary event that happens when the tip of the swing leg collides with the ground (2.2), causes jumps in velocities, and triggers coordination changes, i.e., the swing leg becomes stance leg, and the stance leg becomes swing leg. Switching surface (2.2) captures the moment when the tip of the swing leg hits the ground ahead of the stance leg, where $p_2 = [p_2^h; p_2^v]$ represents position of the tip of the swing leg.

$$\Sigma : \begin{cases} \dot{x} = f(x) + g(x)u & x^- \notin \mathcal{S} \\ x^+ = \Delta(x^-) & x^- \in \mathcal{S} \end{cases} \tag{2.1}$$

$$\mathcal{S} := \{(q, \dot{q}) \in T\mathcal{Q} \mid p_2^v(q) = 0, \;\; p_2^h > 0\} \tag{2.2}$$

The morphology of the bipedal walker that is employed in this study includes $n = 5$ number of links with $m = n - 1$ number of actuation with point feet termination (see Fig. 2.1). The system includes two knee and two hip joints along with two tibias, two femurs, and one torso link (see Fig. 2.2). Because of the point feet termination, no actuation is possible between the ground and the tip of the stance leg. As a result, the system is underactuated, i.e., the number of actuation is less than the number of joints. The system is represented in body coordinates that are $n-1$ relative angles and an absolute angle. This representation provides the necessary information to model

the system configuration in space. Additionally, it is a realistic representation since it is not convenient to obtain the absolute angle of each joint in real-life applications. The model parameters, which are inherited from RABBIT [15], are shown in Fig. 2.2 and given in Table. 2.1.

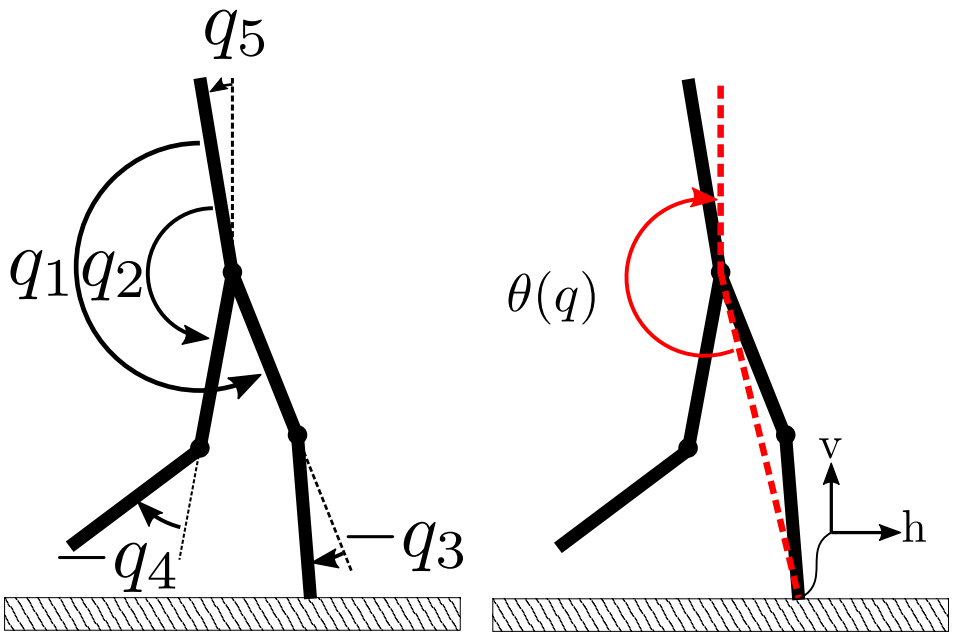

Figure 2.1: Schematic of a 5-link bipedal walker with point feet.

2.1 Stance Dynamics

Stance dynamics of the walker is based on a pinned open kinematic chain configuration. It is assumed that the stance leg is always in contact with the ground, and its end is pivoted, i.e., there is an imaginary revolute joint between the ground and the tip of the stance leg. For a set of generalized coordinates $q = (q_1; q_2; q_3; q_4; q_5)$ in an $n - dimensional$ configuration space $\mathcal{Q}$, dynamic model of the system can be obtained using the method of Lagrange [37, 38]. To begin with, Lagrangian (2.3) of the system should be calculated using total kinetic energy K (2.4) and total potential energy V (2.5) of the system. In the equations, $p^h_{cm,i}$ and $p^v_{cm,i}$ represent horizontal and vertical coordinates of the center of mass of the i^{th} link, $J_{cm,i}$ represents moment of inertia about the center of mass of the i^{th} link, and θ^{abs}_i represents absolute orientation

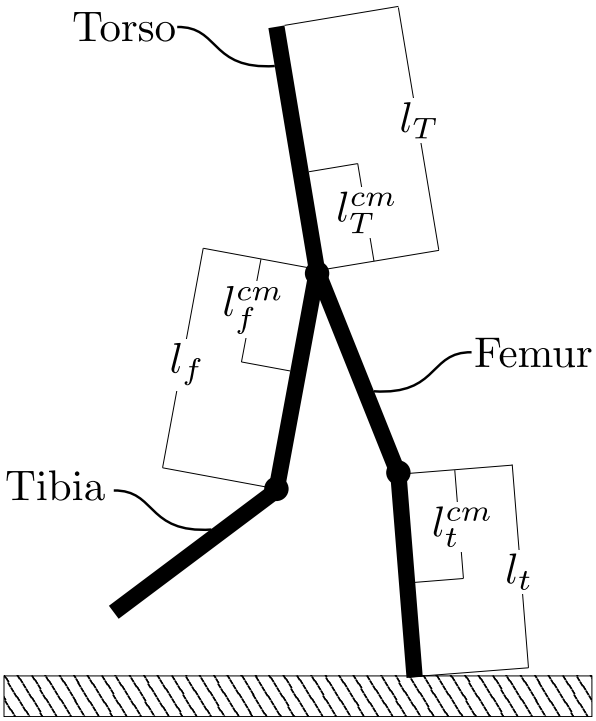

Figure 2.2: Model parameters of the bipedal walker.

of the i^{th} link with respect to the inertial frame (2.6).

$$\mathcal{L}(q, \dot{q}) = K(q, \dot{q}) - V(q) \tag{2.3}$$

$$K = \sum_{i=1}^{n} K_i = \sum_{i=1}^{n} \left(\frac{1}{2} m_i \left((\dot{p}_{cm,i}^{h})^2 + (\dot{p}_{cm,i}^{v})^2 \right) + \frac{1}{2} J_{cm,i} \left(\dot{\theta}_i^{abs} \right)^2 \right) \tag{2.4}$$

$$V = \sum_{i=1}^{n} V_i = \sum_{i=1}^{n} m_i g_0 p_{cm,i}^{v} \tag{2.5}$$

$$\begin{aligned} \theta_5^{abs} &= \frac{\pi}{2} + q_5 \\ \theta_1^{abs} &= \theta_5^{abs} + q_1 \\ \theta_2^{abs} &= \theta_5^{abs} + q_2 \\ \theta_3^{abs} &= \theta_1^{abs} + q_3 \\ \theta_4^{abs} &= \theta_2^{abs} + q_4 \end{aligned} \tag{2.6}$$

The Lagrangian is employed in Lagrange's equation,

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = \Gamma, \tag{2.7}$$

Table 2.1: Model parameters of the walker.

Model Parameter	Units	Label	Value
Mass	kg	m_T	12
		m_f	6.8
		m_t	3.2
Length	m	l_T	0.63
		l_f	0.4
		l_t	0.4
Inertia	$kg \cdot m^2$	I_T	1.33
		I_f	0.47
		I_t	0.20
Mass Center	m	l_T^{cm}	0.24
		l_f^{cm}	0.11
		l_t^{cm}	0.24
Viscous friction	$Nm \cdot s$	b_{Hip}	10
		b_{Knee}	5

where Γ represents vector of generalized torques and forces, such that it results in the second-order differential equation,

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \Gamma. \tag{2.8}$$

Matrices $D = D^\top \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{n\times n}$ and $G \in \mathbb{R}^n$ represent mass-inertia matrix, coriolis matrix and gravity vector, respectively. D, G and C are selected as shown in (2.9), (2.10) and (2.11), respectively, where C_{kj} is the kj entry of the matrix. Analytic representation of a similar system's system dynamics can be found in [39, Appendix E].

$$K(q,\dot{q}) = \frac{1}{2}\dot{q}' D(q)\dot{q} \tag{2.9}$$

$$G(q) = \frac{\partial V(q)}{\partial q} \tag{2.10}$$

$$C_{kj} = \sum_{i=1}^{n} \frac{1}{2} \left(\frac{\partial D_{kj}}{\partial q_i} + \frac{\partial D_{ki}}{\partial q_j} - \frac{\partial D_{ij}}{\partial q_k} \right) \dot{q}_i \quad \text{where} \quad 1 \leq k, \quad j \leq n \tag{2.11}$$

The sum of external generalized forces and torques Γ covers actuation, viscosity and (if any) spring effects on the dynamics. In this study, viscosity and (if any) spring dynamics are included into the coriolis matrix and gravity vector, respectively, such that Γ only represents input as in (2.12), where $B \in \mathbb{R}^{n \times m}$ represents input matrix for $m = n - 1$ number of actuation.

$$\Gamma = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}}_{u} \tag{2.12}$$

General state-space representation of the nonlinear system with configuration variables $q \in \mathcal{Q}$ is expressed as:

$$\dot{x} := f(x) + g(x)u \tag{2.13}$$

where $x(t) = (q(t); \dot{q}(t)) \in \mathcal{TQ}$,

$$f(x) = \begin{bmatrix} \dot{q} \\ D^{-1}(q)[-C(q,\dot{q})\dot{q} - G(q) + B(q)u] \end{bmatrix}$$
$$\text{and} \tag{2.14}$$
$$g(x) = \begin{bmatrix} 0 \\ D^{-1}(q)B(q) \end{bmatrix}.$$

2.2 Impact Dynamics

A widely used impact model for kinematic chains with multiple contact points [40] can be implemented for the bipedal walkers with some assumptions [12]. It is assumed that the impact is instantaneous and there is no rebound and slipping after the impact. It is also assumed that the impulsive forces cause instantaneous changes in the velocities, but it does not cause any change in positions. The impact model of

the system requires the reaction forces at the leg ends. Thus it requires unpinned $(n + 2 - DOF)$ model of the system. If the Cartesian coordinate added to the robot is labelled as $p_e = (p_e^h; p_e^v)$ then the generalized coordinates of this system turns out to be $q_e = (q; p_e)$. Again, the method of Lagrange results in dynamics as shown in (2.15), where F_{ext} represents external impulsive forces acting between the ground and the tip of the swing leg when the collision occurs.

$$D_e(q)\ddot{q} + C_e(q, \dot{q})\dot{q} + G_e(q) = Bu + \delta F_{ext}. \tag{2.15}$$

From the conservation of momentum, change in the momentum throughout the impact event, which is momentarily, is equal to the external forces (2.16) where $F_{ext} = \int_{t^-}^{t^+} \delta F_{ext}(\tau) d\tau$ where t^- and t^+ represent time just before and just after the impact, respectively.

$$D_e(q_e^+)\dot{q}_e^+ - D_e(q_e^-)\dot{q}_e^- = F_{ext} \tag{2.16}$$

If the impulsive forces at the tip of the swing leg are represented as $F_2 = (F_2^h; F_2^v)$, F_{ext} turns out to be,

$$F_{ext} = E_2(q_e^-)' F_2 \tag{2.17}$$

where, $E_2(q_e) = \dfrac{\partial}{\partial q_e} p_2(q_e)$ and $p_2(q_e)$ represents position of the tip of the swing leg with respect to the inertial frame. The conservation of momentum equation (2.16) contains $n + 2$ equations along with $n + 4$ unknowns which are $\dot{q}_e^+$, F_2^h and F_2^v. The required additional two equations comes from no slip and rebound assumption,

$$E_2(q_e^-)\dot{q}_e^+ = 0 \tag{2.18}$$

which states that velocity of the tip of the swing leg just after the impact is zero. Combination of the equations yields to

$$\begin{bmatrix} D_e(q_e^-) & -E_2(q_e^-)' \\ E_2(q_e^-) & 0_{2\times 2} \end{bmatrix} \begin{bmatrix} \dot{q}_e^+ \\ F_2 \end{bmatrix} = \begin{bmatrix} D_e(q_e^-)\dot{q}_e^- \\ 0_{2\times 1} \end{bmatrix}. \tag{2.19}$$

Since the matrix D_e is positive definite and E_2 is full rank, left hand side of (2.19) is invertible. If the location of the Cartesian coordinate p_e added to the robot is chosen to be the tip of the stance leg then,

$$\begin{bmatrix} D_e(q_e^-) & -E_2(q_e^-)' \\ E_2(q_e^-) & 0_{2\times 2} \end{bmatrix} \begin{bmatrix} \dot{q}_e^+ \\ F_2 \end{bmatrix} = \begin{bmatrix} D_e(q_e^-) \begin{bmatrix} I_{n\times n} \\ 0_{2\times n} \end{bmatrix} \\ 0_{2\times n} \end{bmatrix} \dot{q}^-. \tag{2.20}$$

As a result the impact map for velocities, $\Delta_{\dot{q}}(x^-)$, turns out to be,

$$\begin{aligned}\Delta_{\dot{q}}(x^-)\dot{q}^- &:= \begin{bmatrix} R & 0_{n\times 4}\end{bmatrix}\begin{bmatrix}\dot{q}_e^+ \\ F_2\end{bmatrix} \\ &= \underbrace{\begin{bmatrix} R & 0_{n\times 4}\end{bmatrix}\begin{bmatrix} D_e(q_e^-) & -E_2(q_e^-)' \\ E_2(q_e^-) & 0_{2\times 2}\end{bmatrix}^{-1}\begin{bmatrix} D_e(q_e^-)\begin{bmatrix} I_{n\times n} \\ 0_{2\times n}\end{bmatrix} \\ 0_{2\times n}\end{bmatrix}}_{\Delta_{\dot{q}}(x^-)}\dot{q}^-.\end{aligned} \tag{2.21}$$

where, R is a relabelling matrix. The relabelling matrix (2.22) is used to perform coordinate changes when the impact occurs, i.e., the swing leg becomes the stance leg and vice versa. The resultant impact dynamics that happen when the tip of the swing leg collides with the ground (2.2), causes jumps in velocities and triggers coordination changes, i.e., the swing leg becomes stance leg, and the stance leg becomes the swing leg is shown in (2.23), where Δ_q represents impact map for positions. Using the available information from just before the impact, that is x^-, the impact dynamics is used to calculate the new velocity $\dot{q}^+$ of the system just after the impact and to perform the relabelling operation.

$$R = \Delta_q = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.22}$$

$$x^+ = \Delta(x^-) = \begin{bmatrix} \Delta_q q^- \\ \Delta_{\dot{q}}\dot{q}^- \end{bmatrix} \tag{2.23}$$

CHAPTER 3

TRAJECTORY GENERATION AND CONTROL OF BIPEDAL LOCOMOTION USING HYBRID ZERO DYNAMICS

Zero dynamics control of a system that is modeled by ordinary differential equations covers driving the outputs, which are equal in number to the inputs, to zero [41]. With a proper desired trajectory selection, forcing the output to zero can be used to encode a geometric task to the robot such that driving the output to be zero is equivalent to achieving the desired behavior. Hybrid Zero Dynamics (HZD), on the other hand, is an extension developed to implement zero dynamics to hybrid systems, i.e., systems that include impact dynamics, such that the zero dynamics notion becomes compatible with the complete model of the bipeds [12, 39]. This notion is used to design a walking gait via solving a constrained parameter optimization problem, i.e., desired trajectory generation problem, and used to follow that trajectory. Optimization is used to select output parameters such that zeroing the output is equivalent to walking motion obeying a unique set of kinematic and dynamic constraints. This chapter includes a summary of the HZD method implementation. A more complete mathematical description with related proofs and theorems can be found in [39, Chapter 5-6]. Additionally, this chapter includes an initial condition selection approach that we employ for the optimization problem and a primitive controller parameter selection approach. In the end, some simulation results are provided.

3.1 Swing Phase Zero Dynamics

The swing phase defines state evolution between two consecutive impact events, i.e., the stance leg is in contact with the ground, and the swing leg is moving forward in

the air. Since there is no impact, the system model is expressed as in (2.13). For an output function,

$$y = h(q) := h_0(q) - h_d \circ \theta(q) \tag{3.1}$$

where, $h_0(q)$ represents $m = n - 1$ number of independent quantities that are to be controlled and $h_d \circ \theta(q)$ represents desired evolution of these quantities as a function of a scalar quantity $\theta(q)$, which is called internal clock (see Fig. 2.1). The internal clock is a monotonic one-to-one function that slaves the desired states, and a combination of the configuration variables constitutes it, $\theta(q) = cq$, where $c \in \mathbb{R}^{1\times n}$. The internal clock selection is shown in (3.2).

$$\theta(q) = cq = \begin{bmatrix} -1 & 0 & -1/2 & 0 & -1 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \end{bmatrix} \tag{3.2}$$

Similarly, the controlled quantity selection is shown in (3.3), where $H_0 \in \mathbb{R}^{n-1\times n}$.

$$h_0(q) = H_0 q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \end{bmatrix} \tag{3.3}$$

It is important to note that, driving (3.1) to zero, will force the controlled quantities to track desired state evolution, i.e., $h_0(q) - h_d \circ \theta(q) = 0$. This virtual constraint will be used to mimic walking behavior. Due to the underactuated nature of the system, the unactuated degree of freedom, q_5, will be controlled indirectly. Considering the system dynamics, the controlled quantities should follow a trajectory such that the unactuated quantity will perform the desired behavior. In order to zero the output, one should observe the second derivative of the output (3.4) where acceleration terms are replaced with the corresponding system dynamics terms. $L_g L_f h$ is called decoupling matrix and as long as the decoupling matrix is invertible, substitution of (3.5) into

(3.4) results with $\ddot{y} = 0$.

$$\begin{aligned}\frac{d^2y}{dt^2} =& \underbrace{\begin{bmatrix}\frac{\partial}{\partial q}\left(\frac{\partial h}{\partial q}\dot{q}\right) & \frac{\partial h}{\partial q}\end{bmatrix}\begin{bmatrix}\dot{q} \\ D^{-1}\left[-C\dot{q} - G\right]\end{bmatrix}}_{L_f^2 h} + \underbrace{\frac{\partial h}{\partial q}D^{-1}B}_{L_g L_f h}\, u \\ =& L_f^2 h(q,\dot{q}) + L_g L_f h(q) u.\end{aligned} \tag{3.4}$$

$$u^*(x) = -(L_g L_f h(x))^{-1} L_f^2 h(x) \tag{3.5}$$

As a result, since $\ddot{y} = 0$, for an open set $\tilde{\mathcal{Q}} \subset \mathcal{Q}$ such that for each $q \in \tilde{\mathcal{Q}}$ the decoupling matrix $L_g L_f h(q)$ is invertible, the system dynamics can be restricted into an invariant manifold, i.e., a lower dimensional surface in which if the system is initialized on the surface, the evolution remains on the surface. The two-dimensional embedded submanifold of $T\mathcal{Q}$ takes the following form:

$$\mathcal{Z} = \{x \in T\tilde{\mathcal{Q}} \mid h(x) = 0, L_f h(x) = 0\}. \tag{3.6}$$

The feedback control u^* renders $\mathcal{Z}$ invariant under the swing phase dynamics, and for every $z \in \mathcal{Z}$,

$$f_{zero}(z) = f(z) + g(z)u^*(z) \in T_z\mathcal{Z}. \tag{3.7}$$

The set $\mathcal{Z}$ is called the zero dynamics manifold and $\dot{z} = f_{zero}(z)$ is called the zero dynamics. For a valid coordinate transformation $\Phi := [h; \theta(q)]$ on $\tilde{\mathcal{Q}}$,

$$\begin{aligned}\eta_1 = h(q), \;\; \eta_2 = L_f h(q,\dot{q}), \\ \xi_1 = \theta(q), \;\; \xi 2 = L_f \theta(q,\dot{q}),\end{aligned} \tag{3.8}$$

represents coordinate transformation on $T\tilde{\mathcal{Q}}$. In these coordinates, for,

$$\begin{aligned}q &= \Phi^{-1}(\eta_1, \xi_1) \\ \dot{q} &= \left(\frac{\partial \Phi}{\partial q}\right)^{-1}\begin{bmatrix}\eta_2 \\ \xi_2\end{bmatrix}\end{aligned} \tag{3.9}$$

the swing phase dynamics $\dot{x} = f(x) + g(x)u$, and the output $y = h(q)$ takes the following form:

$$\begin{aligned}\dot{\eta}_1 = \eta_2, \;\; \dot{\eta}_2 = L_f^2 h + L_g L_f h u, \\ \dot{\xi}_1 = \xi_2, \;\; \dot{\xi}_2 = L_f^2 \theta + L_g L_f \theta u, \\ y = \eta_1.\end{aligned} \tag{3.10}$$

For input u^*, on the zero dynamics manifold $y \equiv 0$, such that $\eta_1 = h = 0$ and $\eta_2 = L_f h = 0$. As a result the zero dynamics becomes:

$$\begin{aligned} \dot{\xi}_1 &= \xi_2, \\ \dot{\xi}_2 &= L_f^2 \theta + L_g L_f \theta u^*. \end{aligned} \tag{3.11}$$

Equation (3.11) indicates that the zero dynamics is a second-order system. Even though dimension of the system is reduced, this form of the equation is very difficult to compute due to the required matrix inversions. With this motivation, Theorem 5.1 of [39] indicates that for a modified set of coordinates, on $\mathcal{Z}$, the computation of the zero dynamics is easier. The theorem states that relating zero dynamics states with θ and generalized momentum conjugate to θ on $\mathcal{Z}$, $(\xi_1; \xi_2) = (\theta(q); \overline{\sigma}_n)$, is a valid set of coordinates. For

$$H := \begin{bmatrix} H_0 \\ c \end{bmatrix} \tag{3.12}$$

the new set of coordinates are shown in (3.13) and (3.14), where q_b indicates body coordinates and θ indicates the internal clock, i.e., an absolute angle.

$$\tilde{q} := Hq = \begin{bmatrix} q_b \\ \theta \end{bmatrix} \tag{3.13}$$

$$\dot{\tilde{q}} := H\dot{q} = \begin{bmatrix} \dot{q}_b \\ \dot{\theta} \end{bmatrix} \tag{3.14}$$

In these new set of coordinates, the potential energy, the inertia matrix and the output function becomes as in (3.15), (3.16) and (3.17), respectively.

$$\tilde{V}(\tilde{q}) = V(q)|_{q=H^{-1}\tilde{q}} \tag{3.15}$$

$$\tilde{D}(\tilde{q}) = (H^{-1})' D(q) H^{-1}|_{q=H^{-1}\tilde{q}} \tag{3.16}$$

$$y = h(\tilde{q}) := q_b - h_d(\theta) \tag{3.17}$$

In the coordinates $(q_b; \theta; \dot{q}_b; \dot{\theta})$, the zero dynamics manifold can be written as

$$\mathcal{Z} := \left\{ (q_b; \theta; \dot{q}_b; \dot{\theta}) \;\middle|\; q_b = h_d(\theta), \dot{q}_b = \frac{\partial h_d(\theta)}{\partial \theta} \dot{\theta} \right\}. \tag{3.18}$$

For the virtual inertia

$$I(\theta) := \left[\tilde{d}_{n,n}(q_b) + [\tilde{d}_{n,1}(q_b), \ldots, \tilde{d}_{n,n-1}(q_b)]\frac{\partial h_d(\theta)}{\partial \theta}\right]\Bigg|_{q_b=h_d(\theta)}, \tag{3.19}$$

the generalized momentum conjugate to θ on $\mathcal{Z}$ becomes

$$\overline{\sigma}_n = I(\theta)\dot{\theta}. \tag{3.20}$$

On the zero dynamics manifold, since there is no external input to the system, the rate of change of angular momentum is directly related with potential energy,

$$\dot{\overline{\sigma}}_n = -\frac{\partial \tilde{V}}{\partial \theta}(q_b, \theta)\Big|_{q_b=h_d(\theta)}. \tag{3.21}$$

Explicit representation of $\tilde{V}$ is shown in (3.22), as a result, the rate of change of angular momentum becomes as in (3.23).

$$\tilde{V}(q_b, \theta) = m_{tot} g_0 p^v_{cm}(q_b, \theta) \tag{3.22}$$

$$\dot{\overline{\sigma}}_n = m_{tot} g_0 p^h_{cm}(q_b, \theta)\Big|_{q_b=h_d(\theta)} \tag{3.23}$$

For $(\xi_1; \xi_2) = (\theta(q); \overline{\sigma}_n)$, zero dynamics of the system becomes as

$$\begin{aligned} \dot{\xi}_1 &= \kappa_1(\xi_1)\xi_2 \\ \dot{\xi}_2 &= \kappa_2(\xi_1) \end{aligned} \tag{3.24}$$

where,

$$\begin{aligned} \kappa_1(\xi_1) &= \frac{1}{I(\xi_1)} \\ \kappa_2(\xi_1) &= m_{tot} g_0 p^h_{cm}(h_d(\xi_1), \xi_1). \end{aligned} \tag{3.25}$$

Lastly, using the principle of angular momentum transfer, change in the angular momentum conjugate after the each step can be calculated as

$$\xi_2^+ = \xi_2^- + L_s m_{tot} \dot{p}^{v-}_{cm}, \tag{3.26}$$

where L_s is the step length. As a consequence, after each step, the angular momentum conjugate is scaled by δ_{zero}, such that,

$$\xi_2^+ = \delta_{zero}\xi_2^- \;\text{ where }\; \delta_{zero} = 1 + \frac{L_s m_{tot} \dot{p}^{v-}_{cm}}{\xi_2^-}. \tag{3.27}$$

3.2 Parameterization of h_d by Bézier Polynomials

So far, while discussing the output function (3.1), the existence of a desired trajectory is repeatedly mentioned, but there were no details about its structure. It is mentioned that a trajectory generation optimization problem will be solved to select the parameters of the desired trajectory function, h_d. In order to encode the desired trajectory into h_d, Bézier polynomial [42] notion is employed. Using a vector of Bézier polynomials, a predefined trajectory can be calculated for each controlled quantity such that following these trajectories ends up with the desired behavior. An one-dimensional degree M Bézier polynomial, $b_i : [0, 1] \to \mathbb{R}$, is defined by $M + 1$ coefficients, α_k^i, for $i \leq i \leq (n - 1)$,

$$b_i(s) = \sum_{k=0}^{M} \alpha_k^i \frac{M!}{k!(M-k)!} s^k (1-s)^{M-k}. \tag{3.28}$$

It is important to note that $s \in [0, 1]$. In order to put the polynomial into $h_d \circ \theta(q)$ form, it is necessary to normalize θ, since, in general, $\theta(q)$ will not take values in the unit interval over a phase of single support. As a consequence, s can be defined as,

$$s(q) = \frac{\theta(q) - \theta^+}{\theta^- - \theta^+} \tag{3.29}$$

where, θ^- is the value of θ at the end of the step and θ^+ is the value at the beginning of the step. Since $\theta(q)$ is one-to-one strictly monotonic function, i.e., achieves its maximum and minimum values at the end points, (3.29) takes values in $[0, 1]$. As a result, $h_d \circ \theta(q)$ takes the following form:

$$h_d \circ \theta(q) = \begin{bmatrix} b_1 \circ s(q) \\ b_2 \circ s(q) \\ \vdots \\ b_{n-1} \circ s(q) \end{bmatrix}. \tag{3.30}$$

Some of the useful properties of the Bézier polynomial are shown in (3.31) and (3.32). Equation (3.31) implies that the first and the last coefficient of the polynomial represents the value of the polynomial at the endpoints, respectively. Similarly, (3.32) implies that the rate of change of the polynomial with respect to s at the beginning and the end is determined by the first two and last two coefficients, respectively. These properties will be helpful while discussing the existence and periodicity of the Hybrid

Zero Dynamics.

$$b_i(0) = \alpha_0^i \text{ and } b_i(1) = \alpha_M^i \tag{3.31}$$

$$(\partial b_i(s)/\partial s)|_{s=0} = M(\alpha_1^i - \alpha_0^i) \text{ and } (\partial b_i(s)/\partial s)|_{s=1} = M(\alpha_M^i - \alpha_{M-1}^i) \tag{3.32}$$

3.3 Hybrid Zero Dynamics

Hybrid Zero Dynamics includes the impact dynamics into the zero dynamics notion. If the trajectory contains some impact events, the existence of the HZD is determined by whether the system dynamics after the impact stays on the invariant surface or not, i.e., $\Delta(\mathcal{S} \cap \mathcal{Z}) \subset \mathcal{Z}$ where $\mathcal{S} \cap \mathcal{Z}$ represents the intersection of the invariant surface and the impact surface. If $\Delta(\mathcal{S} \cap \mathcal{Z}) \subset \mathcal{Z}$ then it is clearly possible to obtain a zero dynamics manifold for the complete dynamics of the system.

For a periodic trajectory, achieving $\Delta(\mathcal{S} \cap \mathcal{Z}) \subset \mathcal{Z}$ can be done via showing $h \circ \Delta(\mathcal{S} \cap \mathcal{Z}) = 0$ and $L_f h \circ \Delta(\mathcal{S} \cap \mathcal{Z}) = 0$, respectively. Since the matrix,

$$H := \begin{bmatrix} H_0 \\ c \end{bmatrix} \tag{3.33}$$

is full rank, it is invertible. Equations (3.34) and (3.35) are direct conclusions of (3.31). On the invariant manifold, the system states can be related with the desired state evolutions (3.36), (3.37).

$$h_d(\theta^+) = \alpha_0 \tag{3.34}$$

$$h_d(\theta^-) = \alpha_M \tag{3.35}$$

$$q^+ = H^{-1} \begin{bmatrix} \alpha_0 \\ \theta^+ \end{bmatrix} \tag{3.36}$$

$$q^- = H^{-1} \begin{bmatrix} \alpha_M \\ \theta^- \end{bmatrix} \tag{3.37}$$

As a result, a periodic trajectory with the following constraint,

$$\begin{bmatrix} \alpha_0 \\ \theta^+ \end{bmatrix} = H\Delta_q H^{-1} \begin{bmatrix} \alpha_M \\ \theta^- \end{bmatrix} \tag{3.38}$$

would always guarantee that $h \circ \Delta(\mathcal{S} \cap \mathcal{Z}) = 0$. Equation (3.38) relates the end and the beginning of the desired state evolution according to the impact dynamics such that if the system starts on the invariant manifold, after the impact, the controlled configuration variables will stay on the surface.

In order to achieve $L_f h \circ \Delta(\mathcal{S} \cap \mathcal{Z}) = 0$, a similar methodology can be employed. Similarly, (3.40) and (3.41) are direct conclusions of (3.32) and (3.39).

$$\dot{q} = H^{-1} \begin{bmatrix} \frac{\partial h_d}{\partial \theta} \\ 1 \end{bmatrix} \dot{\theta} \tag{3.39}$$

$$\dot{q}^+ = H^{-1} \begin{bmatrix} \frac{M}{\theta^- - \theta^+}(\alpha_1 - \alpha_0) \\ 1 \end{bmatrix} \dot{\theta}^+ \tag{3.40}$$

$$\dot{q}^- = \underbrace{H^{-1} \begin{bmatrix} \frac{M}{\theta^- - \theta^+}(\alpha_M - \alpha_{M-1}) \\ 1 \end{bmatrix}}_{\omega^-} \dot{\theta}^- \tag{3.41}$$

For a periodic orbit, the end points of the desired trajectory, again, can be related via impact dynamics,

$$\dot{q}^+ = \Delta_{\dot{q}} \dot{q}^-. \tag{3.42}$$

As a consequence,

$$\alpha_1 = H_0 \Delta_{\dot{q}} \omega^- \frac{\theta^- - \theta^+}{M} (c\Delta_{\dot{q}} \omega^-)^{-1} + \alpha_0 \tag{3.43}$$

guarantees that $L_f h \circ \Delta(\mathcal{S} \cap \mathcal{Z}) = 0$ after the impact, as long as $c\Delta_{\dot{q}} \omega^- \neq 0$. As a result, for a periodic trajectory, the existence of the Hybrid Zero Dynamics can be achieved via (3.38) and (3.43) such that when these constraints are applied during the trajectory generation optimization, $\Delta(\mathcal{S} \cap \mathcal{Z}) \subset \mathcal{Z}$ is achieved as long as there exists a solution, i.e., the decoupling matrix is invertible along the trajectory and all other stability, dynamic and kinematic constraints are satisfied.

3.4 Stability and Periodicity Analysis

Assuming $\Delta(\mathcal{S} \cap \mathcal{Z}) \subset \mathcal{Z}$ holds, i.e., the existence of the hybrid zero dynamics holds, with a proper coordinate transformation, stability, and periodicity analysis can be done employing Poincaré return map [39, Chapter 5.4]. Poincaré return map transforms finding periodic orbit for hybrid zero dynamics into existence analysis of a fixed point. The periodic orbit of hybrid zero dynamics also inherits the local stability properties of this fixed point. For a Poincaré section $\mathcal{S} \cap \mathcal{Z}$, the return map turns out to be $P : \mathcal{S} \cap \mathcal{Z} \to \mathcal{S} \cap \mathcal{Z}$. The hybrid zero dynamics coordinates defined in (3.24) is equivalent to

$$\frac{d\xi_2}{d\xi_1} = \frac{\kappa_2(\xi_1)}{\kappa_1(\xi_1)\xi_2}. \tag{3.44}$$

Since the internal clock is defined to be a strictly monotonic function, $\zeta_2 = \frac{1}{2}(\xi_2)^2$ is a valid coordinate change. In these coordinates, (3.44) turns out to be

$$\frac{d\zeta_2}{d\xi_1} = \frac{\kappa_2(\xi_1)}{\kappa_1(\xi_1)}. \tag{3.45}$$

Since the swing phase zero dynamics is Lagrangian, total energy of the orbit is constant over a step. As a result, for $\theta^+ \leq \xi_1 \leq \theta^-$, a potential energy notion can be defined such that

$$V_{zero}(\xi_1) := -\int_{\theta^+}^{\xi_1} \frac{\kappa_2(\xi)}{\kappa_1(\xi)} d\xi. \tag{3.46}$$

As a result of V_{zero}, the equivalent kinetic energy definition becomes

$$K_{zero} = \frac{1}{2}\left(\frac{\dot{\xi}_1}{\kappa_1(\xi_1)}\right)^2 = \zeta_2. \tag{3.47}$$

Integrating (3.45) over a step results in

$$\zeta_2^- = \zeta_2^+ - V_{zero}(\theta^-) \tag{3.48}$$

where, $\zeta_2^- := \frac{1}{2}(\xi_2^-)^2$ by definition and $\zeta_2^+ := \delta_{zero}^2 \zeta_2^-$ as a result of (3.27). Since the total energy is constant over a step, (3.48) indicates that the kinetic energy at the end of the step is determined by how much kinetic energy is gained or lost due to the potential energy evolution over a step. As a result, the Poincaré return map definition turns out to be

$$P(\zeta_2^-) = \delta_{zero}^2 \zeta_2^- - V_{zero}(\theta^-). \tag{3.49}$$

The Poincaré return map indicates that if $\delta_{zero}^2 \neq 1$, for

$$\zeta_2^* := -\frac{V_{zero}(\theta^-)}{1-\delta_{zero}^2} \tag{3.50}$$

there exists a periodic orbit, which indicates that if

$$0 < \delta_{zero}^2 < 1, \tag{3.51}$$

then ζ_2^* is exponentially stable equilibrium point of $P(\zeta_2(k)) = \zeta_2(k+1)$. The stability condition becomes more clear when one disturbs (3.50), by for example ϵ. Substitution of disturbed (3.50) into (3.49) shows that the disturbance fades away if and only if $\delta_{zero}^2 < 1$.

It is important to note that, to talk about an orbit or a Poincaré section at the end of the step, one should ensure that the walker can complete a whole step. To ensure that, a potential energy barrier definition (3.52) is introduced. The potential energy barrier V_{zero}^{MAX} defines the point where the walker has the highest potential energy. After this point, since the center of mass of the walker passes the stance leg, the kinetic energy starts to increase. As a result, a step may be completed if and only if the initial kinetic energy is high enough to pass the potential energy barrier (3.53).

$$V_{zero}^{MAX} := \max_{\theta^+ \leq \xi_1 \leq \theta^-} V_{zero}(\xi_1) \tag{3.52}$$

$$\zeta_2^* > V_{zero}^{MAX}/\delta_{zero}^2 \tag{3.53}$$

3.4.1 Interpretation

Since the swing phase zero dynamics is Lagrangian, total energy, $K_{zero}+V_{zero}$, during the swing phase is constant, as an inverted pendulum subject only to gravity, where K_{zero} and V_{zero} are given in (3.46) and (3.47), respectively. As a result, total energy of a periodic orbit has a constant value $V_{zero}(\theta^-) + \frac{1}{2}(\overline{\sigma}_n^*)^2$, where $\overline{\sigma}_n^*$ indicates generalized momentum conjugate to θ of a periodic orbit. The energy may be gained and lost only at impacts. A constant change of V_{zero} occurs at impact, from V_{zero}^- at the end of the step to V_{zero}^+ at the beginning of the step. The re-initialization rules induce this energy change, i.e., the stance leg becomes the swing leg, and the swing

leg becomes the stance leg. Similarly, the angular momentum of the system, $\bar{\sigma}_n$, is scaled by δ_{zero} (3.27) such that $\bar{\sigma}_n^+ = \delta_{zero}\bar{\sigma}_n^-$. This change is induced by angular momentum transfer caused again by the re-initialization rules.

In order to ensure continuous displacement in the forward direction, the angular momentum of the system, $\bar{\sigma}_n$, should always be positive or negative depending on the direction selection. At the beginning of the step, since the center of mass of the system is behind the stance leg, gravity decreases the angular momentum of the robot until the center of mass passes the stance leg as indicated in (3.23). If the angular momentum is sufficiently large to overcome the potential energy barrier corresponding to V_{zero}^{MAX} (3.52), the center of mass will move past the support leg end, inducing a reverse exchange of energy and the swing leg impacts with the ground. As a consequence, existence of a fixed point can be checked by (3.53), where $\zeta_2^* = \frac{1}{2}(\overline{\sigma}_n^*)^2$. The existence of a fixed point can be discussed if and only if the walker is able to complete a step, and if (3.53) holds, then it can be concluded that the robot will complete the step.

On the other hand, exponential stability can be shown using angular momentum conjugate. At each step, the angular momentum conjugate is scaled by δ_{zero} (3.27). The same scale is true for the difference between the angular momentum and its value on the periodic orbit, given by $\overline{\sigma}_n - \overline{\sigma}_n^*$. Consequently, if the angular momentum conjugate decreases at each impact, the error decreases along with it, and the system converges to the periodic orbit. This condition is shown in (3.51). As a result, dictating (3.53) and (3.51) inside the trajectory generation optimization problem, if there is a solution, it results in an exponentially stable periodic orbit of hybrid zero dynamics. Note that since $V_{zero}(\theta^+) = 0$ and $V_{zero}^{MAX} \geq 0$, $\delta_{zero} < 1$ is equivalent to $V_{zero}(\theta^-) < 0$.

3.5 Trajectory Generation

So far, some properties and the existence of the hybrid zero dynamics of bipedal walking have been discussed. But, it is not pointed out how to select the parameters of the desired state evolution, i.e., h_d. In this section, an optimization problem used to select

the desired state evolution of the system is discussed such that following this trajectory will end up with the desired walking motion. Additionally, it is discussed how to select a proper initial condition for the optimization problem such that computation time is decreased and the chance to end up with a proper trajectory is increased.

3.5.1 Cost Function

A popular cost function over a single step is shown in (3.54), where T_I, p_2^h and $u^*(t)$ represent step time, step length and constraining input (3.5), respectively. The cost function represents input cost of a step and in this study the cost function with dedicated constraints is minimized via MATLAB' s built-in $fmincon$ function.

$$J_{cost} = \frac{1}{p_2^h(q_0^-)} \int_0^{T_I} ||u^*(t)||_2^2 dt \tag{3.54}$$

3.5.2 Constraints

Selection of constraints is so essential to achieving a proper walking motion. The constraints include existence and stability requirements and physical limits to obtain a meaningful and realistic walking motion.

Friction Cone

Friction cone constraint ensures that the tip of the stance leg does not slide during the motion. It ensures that contact forces are always inside a friction cone, i.e., $|F_1^T/F_1^N| < \mu$, where F_1 represents contact forces at the stance leg in tangential and vertical directions.

The calculation of F_1 requires the full $N + 2 - DOF$ model, which is discussed in Section 2.2 and given in Appendix A.2, as in impact dynamics. Then for a Jacobian matrix J, the contact forces can be obtained as the following:

$$F_1 = (JD_e^{-1}J')^{-1}(JD_e^{-1}(C_e\dot{q}_e + G_e - Bu)). \tag{3.55}$$

For $q_e = (q; p_1)$, the Jacobian matrix turns out to be $J = [0_{2\times n}, I_{2\times 2}]$.

Normal Ground Reaction Force

In order to ensure that the stance leg is always in contact with the ground, the normal ground reaction force experienced by the stance leg end should be,

$$F_1^N > 0 \tag{3.56}$$

throughout the step.

Swing Leg Height

In order to ensure that impact occurs only at the end of the step, one should ensure that:

$$p_2^v(q) = 0 \iff q = q^-. \tag{3.57}$$

Average Walking Rate

Average walking rate can be enforced using step length and time to impact information:

$$\bar{\nu} = \frac{p_2^h(q_0^-)}{T_I}. \tag{3.58}$$

Time to impact is the time passed from the beginning of the step to the end.

Ground Separation

In order to ensure that the contact is momentary, the vertical velocity component of the tip of the swing leg just after the impact should be positive such that:

$$\dot{p}_2^v(q^+) > 0. \tag{3.59}$$

In the other words, vertical velocity component of the tip of the swing leg at the very beginning of the step should be positive such that leg scuffing does not occur.

Step Length

The desired step length can be enforced such that:

$$p_2^h(q_0^-) = L_s \tag{3.60}$$

where, $p_2^h(q_0^-)$ represents horizontal position of the tip of the swing leg at the end of the step and L_s represents the desired step length.

Existence of the fixed point

This constraint ensures that the angular momentum of the system is high enough to overcome the potential energy barrier (3.53).

$$\zeta_2^* > V_{zero}^{MAX}/\delta_{zero}^2 \tag{3.61}$$

Stability of the fixed point

This constraint ensures that the fixed point is exponentially stable (3.51).

$$0 < \delta_{zero}^2 < 1. \tag{3.62}$$

Hyperextension

Hyperextension constraints are defined to prevent knee bending in the reverse direction. Other than a requirement, this constraint is a preference to obtain a human-like walking (see Fig. 2.1).

$$q_3 < 0 \tag{3.63}$$

$$q_4 < 0 \tag{3.64}$$

Input Limits

Since each real actuator element has limits, limiting the input of actuators results in realistic motions. This constraint can be used to restrict input torque, force, and

power. In this study, input torque along the trajectory is limited, such that:

$$u < u_{max}. \tag{3.65}$$

3.5.3 The Optimization Problem

The resultant optimization problem is constituted by the combination of the cost function (3.54) and the above-defined nonlinear inequality and equality constraints. The problem is solved to minimize the cost of the system with proper desired trajectory selection. Since the trajectory is fitted into Bézier polynomials, each polynomial is represented by $M+1$ coefficients, α_k^i, for $1 \leq i \leq (n-1)$. The HZD existence conditions (3.38) and (3.43) states that, some of the coefficients are restricted. As a result, for $M = 6$, there are in total $(M+1)(n-1) = 28$ parameters in which 8 are restricted. In total 20 parameters are left to be found out by solving the optimization problem,

$$\begin{gathered} \underset{\alpha^1,\ldots,\alpha^{n-1}}{\arg\min} \left(\frac{1}{p_2^h(q_0^-)} \int_0^{T_I} ||u^*(t)||_2^2 dt \right) \\ \text{such that} \\ \text{(closed-loop dynamics)} \\ \text{(HZD conditions)} \\ \text{(kinematic constraints)} \\ \text{(dynamic constraints)} \end{gathered} \tag{3.66}$$

Note that the desired trajectory contains information only for the controlled quantities. In order to select the initial condition of unactuated quantities, which are q_5 and $\dot{q}_5$, using optimization, these terms can be included in the optimization problem. As a result, the optimization problem makes the initial condition selection of all parameters, i.e., x^-. In this case, the number of parameters required to be found raises from 20 to 22.

3.5.4 Initial Condition Selection for the Optimization Problem

Initial condition selection for the optimization problem (3.66) affects the solution directly. The optimization problem solver may terminate without finding a useful

parameter depending on the initial condition selection. Additionally, even though the solver finds a solution, it may take too much time to solve due to the initial condition selection. Considering the natural posture of the walker (see Fig. 2.1) and kinematics constraints, a logical initial condition selection can be made.

At the beginning of a step, the tip of the swing and stance legs is expected to be on the ground. Additionally, it is expected that the distance between the tips is equal to the step length. To start with, initial condition of q_3^{0-}, q_4^{0-} and q_5^{0-} are selected as $(q_3^{0-}; q_4^{0-}; q_5^{0-}) = -(5; 5; 5)$ degrees. Solving for q_1^{0-} and q_2^{0-} such that $p_1^- = (0; 0)$ and $p_2^- = (0.4; 0)$, where p_1 and p_2 represents position of the tip of the stance and swing legs, respectively, results with $(q_1^{0-}; q_2^{0-}) = (3.0196; 3.5254)$ radians. Additionally, the initial velocity of the walker should be selected such that the robot moves (falls) forward. The resultant initial state selection is formed as the following:

$$x_0^- = \begin{bmatrix} q_1^{0-} \\ q_2^{0-} \\ q_3^{0-} \\ q_4^{0-} \\ q_5^{0-} \\ \dot{q}_1^{0-} \\ \dot{q}_2^{0-} \\ \dot{q}_3^{0-} \\ \dot{q}_4^{0-} \\ \dot{q}_5^{0-} \end{bmatrix} \begin{bmatrix} 3.0196 \\ 3.5254 \\ -0.0873 \\ -0.0873 \\ -0.0873 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix} \tag{3.67}$$

where position and velocity variables are expressed in rad and rad/s, respectively. Also from (2.23), $x^+ = \Delta(x^-)$. The resultant initial posture of the robot is shown in Fig. 3.1. Relating the initial state selection of the walker with the first two and last two parameters of the Bézier polynomial are shown in (3.36), (3.37), (3.40) and (3.41). The rest of the parameters can be selected as in (3.68). The first row indicates a smooth displacement of the stance leg from q_1^+ to q_1^-. The second row is selected such that the swing leg moves to forward quickly. The third row represents desired state evolution for stance leg knee, q_3. The stance leg knee is set to expand during the step. Lastly, the last row indicates the desired state evolution for the swing leg knee,

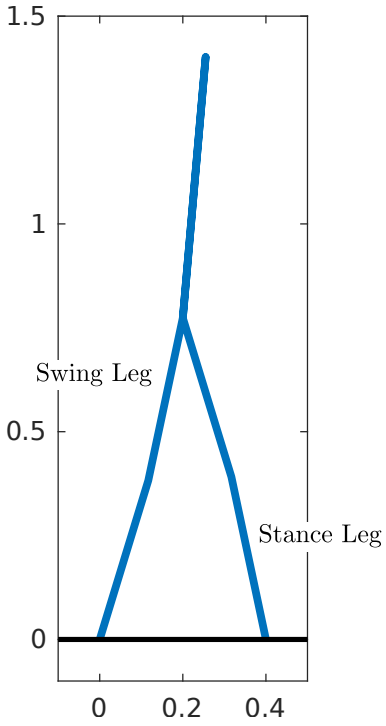

Figure 3.1: Posture of the walker at the initial condition.

q_4. It is decreased quickly to prevent foot scuffing.

$$\begin{bmatrix} \alpha_1^2 & \alpha_1^3 & \alpha_1^4 \\ \alpha_2^2 & \alpha_2^3 & \alpha_2^4 \\ \alpha_3^2 & \alpha_3^3 & \alpha_3^4 \\ \alpha_4^2 & \alpha_4^3 & \alpha_4^4 \end{bmatrix} = \begin{bmatrix} 3.45 & 3.30 & 3.15 \\ 3.5 & 3.5 & 3.5 \\ 0 & 0 & 0 \\ -0.8 & -0.8 & -0.8 \end{bmatrix} \tag{3.68}$$

3.6 Generating Exponentially Stable Orbit

So far, all mathematical results represented in this section are valid in the zero dynamics manifold. The objective is to show that exponentially stable periodic orbits of hybrid zero dynamics are exponentially stabilizable periodic orbits of the full-order system. As long as initial condition is on the zero dynamics manifold, i.e., $y = 0$ and $\dot{y} = 0$, and the input is chosen to be u^* (3.5), the state evolution stays on the zero dynamics manifold since

$$\ddot{y} = 0. \tag{3.69}$$

In order to force the system to converge back to the zero dynamics manifold, whenever it diverges due to disturbance and modeling errors, the input form can be modified as

$$u(x) = (L_g L_f h(x))^{-1}(v - L_f^2 h(x)), \tag{3.70}$$

such that,

$$\ddot{y} = v. \tag{3.71}$$

Selecting v to be,

$$v = -K_D \dot{y} - K_P y, \tag{3.72}$$

where K_D and K_P are $m \times m$ (where $m = n - 1$) positive definite gain matrices, results in

$$\ddot{y} = -K_D \dot{y} - K_P y. \tag{3.73}$$

Equation (3.73) indicates that, the resultant output dynamics is exponentially stable.

3.6.1 Parameter Selection

In (3.73) it is shown that the computed torque with the PD control method results in exponentially stable output dynamics. Parameter selection for this output dynamics determines the general behavior of the system. Selection should be made such that the walking is stable and joints do not oscillate. Additionally, it is also essential to make the walker robust against external disturbances such as modeling errors, external forces, input noises, and initial condition disturbances in velocity and position components. The parameter selection is made by observing the eigenvalues and eigenvectors of Poincaré return map estimations, which are calculated numerically. Calculation details of this map are covered extensively in Chapter 6.2. This section discusses the methodology, ideas, and concerns behind the parameter selection.

Thanks to trajectory generation optimization, on the invariant manifold, computed torque control follows the predetermined trajectory well. But, it is too fragile against any disturbance. That is why, as long as there is no oscillation due to the parameter selection, almost for any positive definite K_P and K_D selection, the system is stable. The parameter selection is mainly made to increase the robustness of the walker against various uncertainties, modeling errors, and disturbances. The PD part of the

computed torque with PD control (3.70) is used to force the states to converge back into the invariant manifold as soon as it diverges from there. Eigenvalues and eigenvectors of numerically calculated Poincaré return map estimations are employed for parameter tuning. One important drawback of numerical computation is that they are not much insightful. It is very difficult, if not impossible, to establish a cause-and-effect relationship between a particular parameter and the resultant numerical computation. In this case, it is even harder because the system's dynamics are highly nonlinear, and the dimension of the model is too high. In order to resolve this issue, in this study, a computationally intensive primitive search algorithm is employed. Starting from a base K_P and K_D selection, the search algorithm checks the effects of each parameter on the eigenvalues of the Jacobian matrix, i.e., the numerical derivative of the Poincaré return map with respect to the state variables (see Chapter 6.2), then it selects a proper parameter combination. Since the calculation requires a significant computation source, the search is done independently for each parameter. If K_P and K_D are said to be equal to $diag(K_{P1}, K_{P2}, K_{P3}, K_{P4})$, and $diag(K_{D1}, K_{D2}, K_{D3}, K_{D4})$, respectively, then implementation of the search algorithm can be expressed as:

1. Set $K_P = K_D = diag(20, 20, 20, 20)$ and $i = 0$.
2. Set $i = i + 1$.
3. Calculate the eigenvalues of the closed-loop system for each of $K_{Pi} - 10$, K_{Pi}, and $K_{Pi} + 10$.
4. Select the one with the smallest eigenvalue summation $\sum_{i=1}^{2n} (1 + |\lambda_i|)^2$.
5. Repeat step 2, 3 and 4 for each element of K_P and K_D, i.e., until $i = 2n$, but at each calculation set the other parameters to their initial values. In the first iteration, for example, while calculating for $K_{P2} - 10$, K_{P2}, $K_{P2} + 10$, use $K_{P1} = 20$.
6. Update the initial condition of K_P and K_D with the selected parameters.
7. Set $i = 0$ and start again from step 2 until the smallest eigenvalue combination is found.

It is important to note that in the search algorithm minimum of $\sum_{i=1}^{2n} (1 + |\lambda_i|)^2$ checked instead of directly aiming for maximum magnitude eigenvalue. There are two main reasons for this selection. Firstly, the orbit of the full-order model is a stabilizable periodic orbit. This means some eigenvalues are fixed independent of

K_P and K_D selection. Aiming only for the highest eigenvalue may not cause any difference between different parameters. For example, in this case, independent of the parameter selections, the highest eigenvalue is 0.75. Secondly, this condition decreases oscillation of the joints and tries to reduce all eigenvalues. As a result of the search, the controller parameters turned out to be $K_P = diag(60, 90, 90, 50)$ and $K_D = diag(10, 20, 20, 10)$. The resultant changes in the eigenvalues are:

$$\underbrace{\begin{bmatrix} 0.75 \\ 0.61 \\ -0.57 + 0.19i \\ -0.57 - 0.19i \\ 0.53 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\lambda_{before}} \longrightarrow \underbrace{\begin{bmatrix} 0.75 \\ -0.05 \\ -0.04 + 0.06i \\ -0.04 - 0.06i \\ -0.03 + 0.03i \\ -0.03 - 0.03i \\ 0.02 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\lambda_{after}}. \tag{3.74}$$

3.7 Simulation Results

A simulation environment is utilized to test the results of the optimization. The simulation is conducted using $MATLAB$'s $ode45$ solver. Compatible with the nature of the computers, the controller is implemented in discrete time, in 1kHz frequency. Step length, average walking rate, and friction coefficient are selected during the optimization to be $0.4m$, $0.8m/s$, and 0.6, respectively. Optimized Bézier polynomial parameters are shown in Table 3.1. The resultant initial state of the system turns out

to be:

$$x^- = \begin{bmatrix} q_1^- \\ q_2^- \\ q_3^- \\ q_4^- \\ q_5^- \\ \dot{q}_1^- \\ \dot{q}_2^- \\ \dot{q}_3^- \\ \dot{q}_4^- \\ \dot{q}_5^- \end{bmatrix} \begin{bmatrix} 2.9195 \\ 3.5369 \\ -0.1483 \\ -0.3626 \\ 0.0147 \\ -0.2077 \\ -0.1565 \\ -0.2521 \\ 0.2285 \\ -0.8486 \end{bmatrix}. \tag{3.75}$$

Table 3.1: Bézier polynomial parameters for fitted desired trajectory

i	α_0	α_1	α_2	α_3	α_4	α_5	α_6
1	3.54	3.66	3.39	3.28	3.02	2.93	2.92
2	2.92	2.94	3.12	3.73	3.62	3.55	3.54
3	−0.36	−0.60	−0.28	−0.18	−0.03	−0.13	−0.15
4	−0.15	−0.18	−0.51	−0.81	−0.48	−0.38	−0.36

The posture of the walker throughout the generated periodic trajectory is sketched in Fig. 3.2. State evolution of the walker is shown in Fig. 3.3 and Fig. 3.4. State evolution of the controlled variables, $H_0 q$ and $H_0 \dot{q}$, are identical with the Bézier polynomial output from θ^+ to θ^-. State evolution shows that the walking is periodic, and the knee angles are always negative. The jumps in configuration variables are caused by the re-initialization rules. Similarly, the jumps in velocity variables are caused by the impact dynamics along with the re-initialization rules. The resultant input of the controller is shown in Fig. 3.5. Since the decoupling matrix is invertible along the trajectory, the inputs are finite. Fig. 3.6 indicates that the internal clock, θ, is a monotonically increasing one-to-one function. It also shows that the angular momentum of the walker is always positive, i.e., $\dot{\theta} > 0$. Lastly, the contact force response of the walker at the tip of the stance leg is shown in Fig. 3.7. It can be seen that F_T/F_N

during the step is always smaller than the maximum friction coefficient, μ, defined inside the optimization problem.

Since the optimization is used only for desired trajectory generation and is performed offline, the system is not input limited in online implementation. Online implementation of the walker covers applying a computed torque (3.70) to the system. One drawback of this method is that, when there is a disturbance, the input torque increases dramatically (see Fig. 3.8). This increase may cause saturation in real-life applications and cause defined dynamical constraints to be exceeded, such as friction cone. Even though the figure shows that the system converges to the desired periodic trajectory, the walker may fail in real-life applications due to motor saturations. For example, when the hip motors, u_1 and u_2, saturate at $75Nm$ and the knee motors, u_3 and u_4, saturate at $50Nm$, under the defined disturbance, the walker cannot even complete a single step and falls forward.

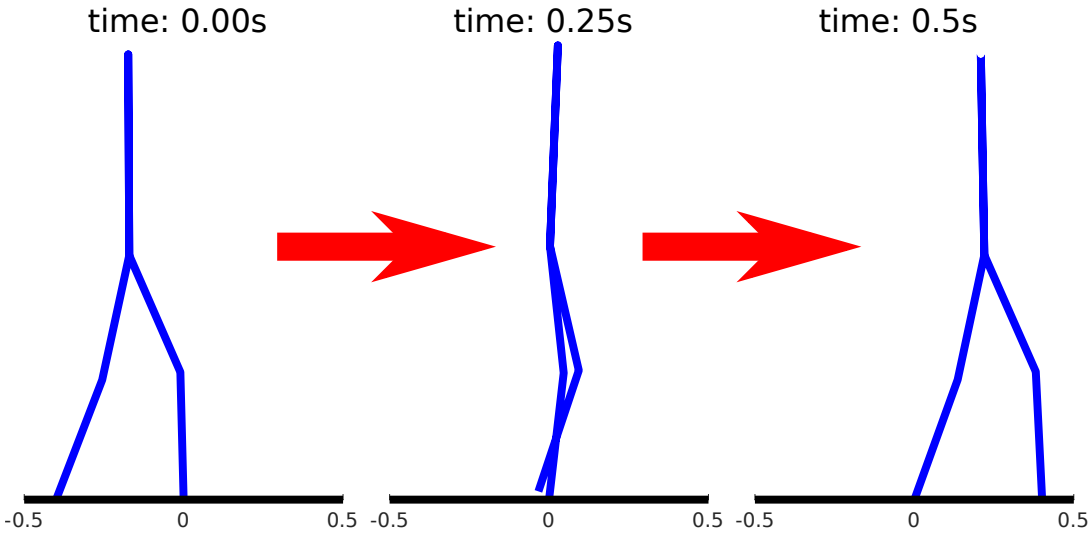

Figure 3.2: Posture of the walker throughout the generated periodic trajectory with 0.4 meters step length and 0.5 seconds step time.

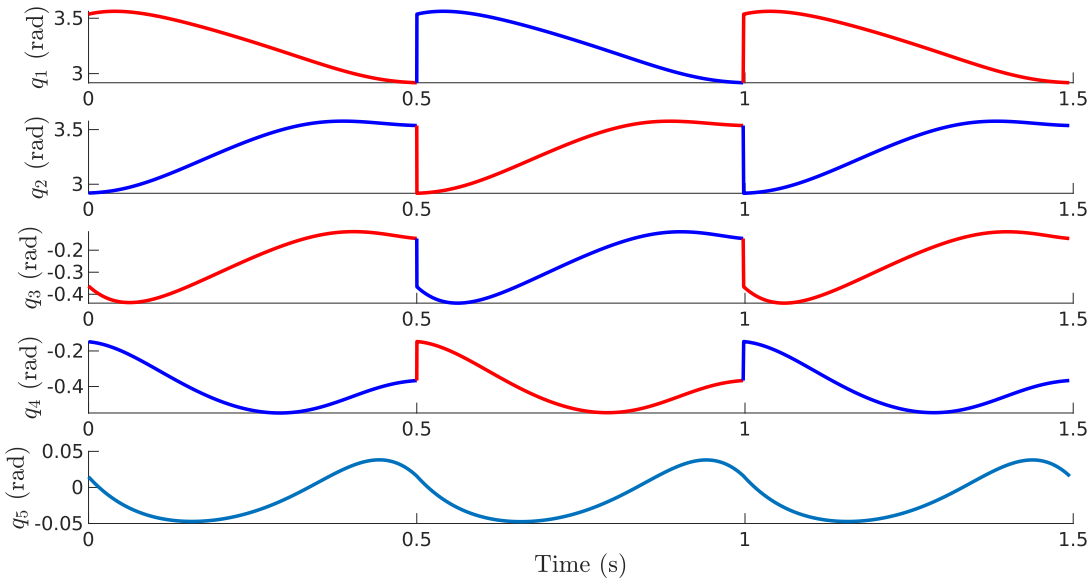

Figure 3.3: Position evolution of the walker under undisturbed conditions. Red and blue indicate leg-1 and leg-2, initialized as stance and swing legs, respectively.

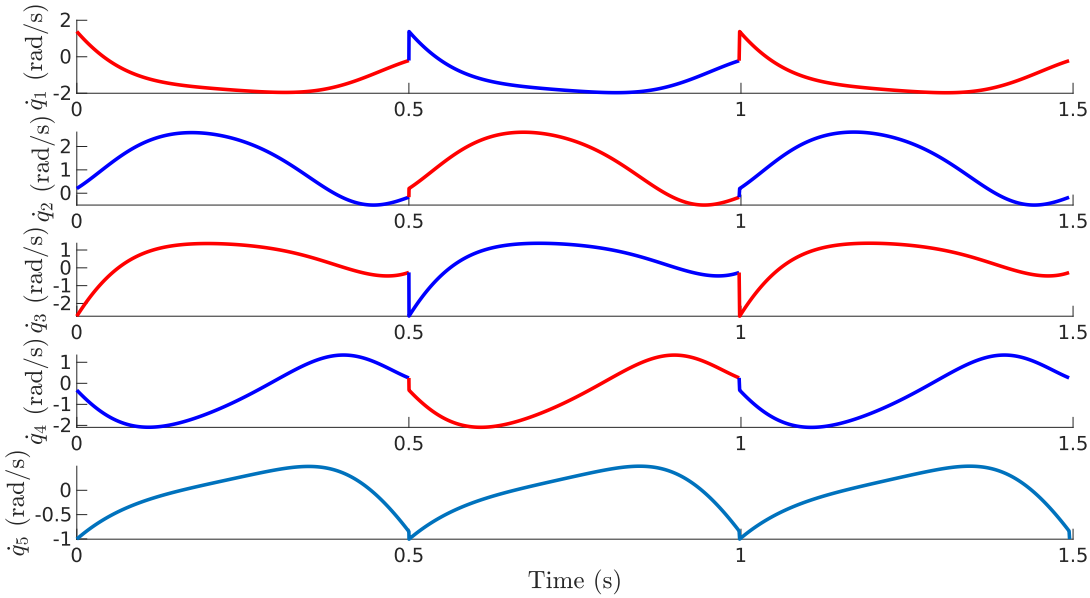

Figure 3.4: Velocity evolution of the walker under undisturbed conditions. Red and blue indicate leg-1 and leg-2, initialized as stance and swing legs, respectively.

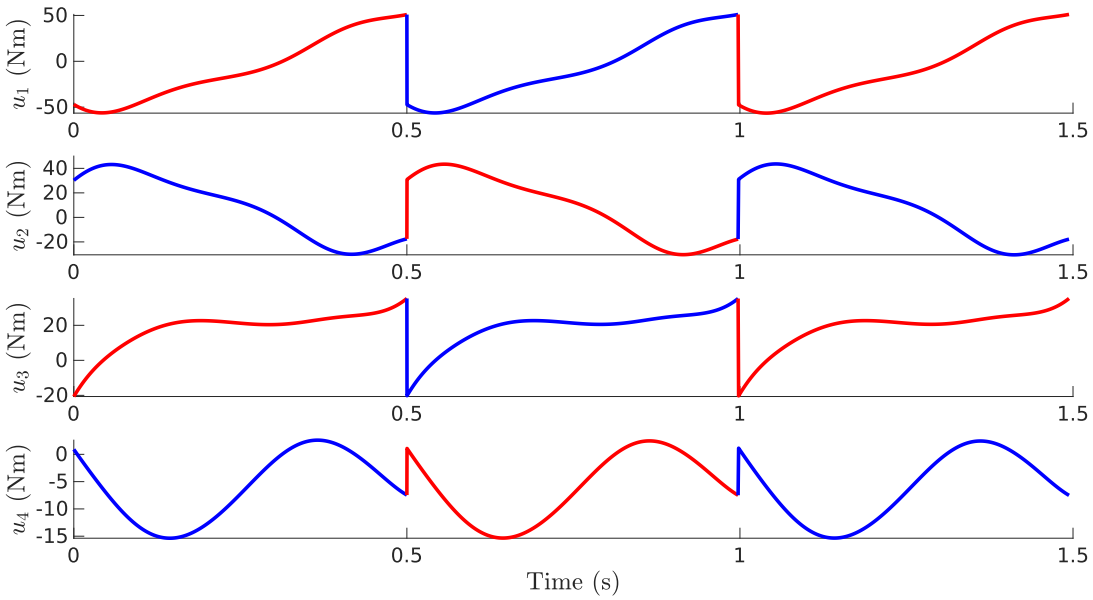

Figure 3.5: Input response of the controller under undisturbed conditions. Red and blue indicate leg-1 and leg-2, initialized as stance and swing legs, respectively.

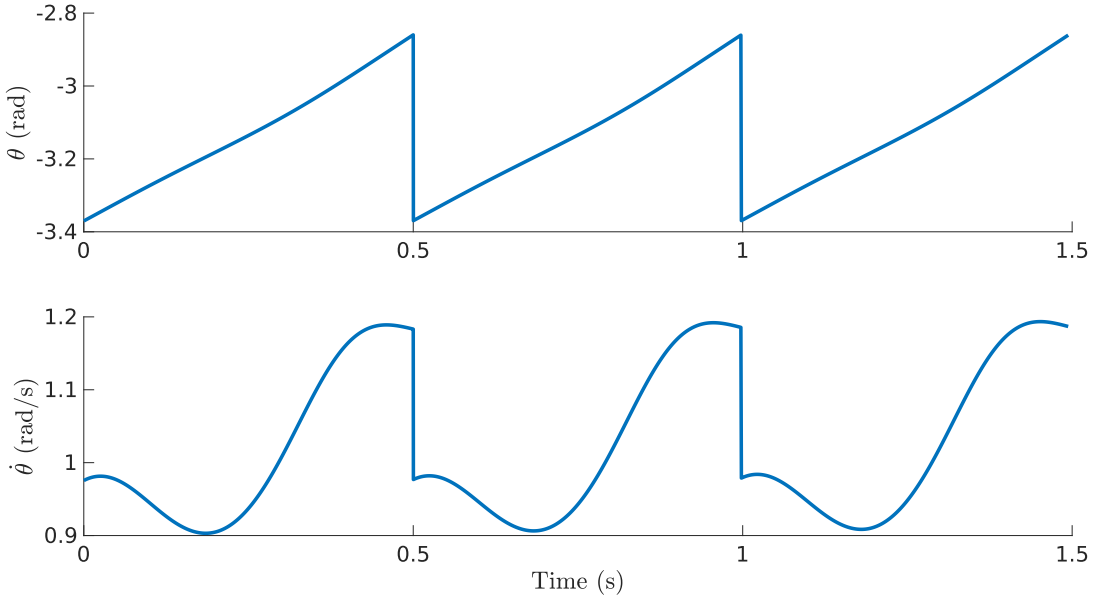

Figure 3.6: Internal clock evolution of the walker under undisturbed conditions.

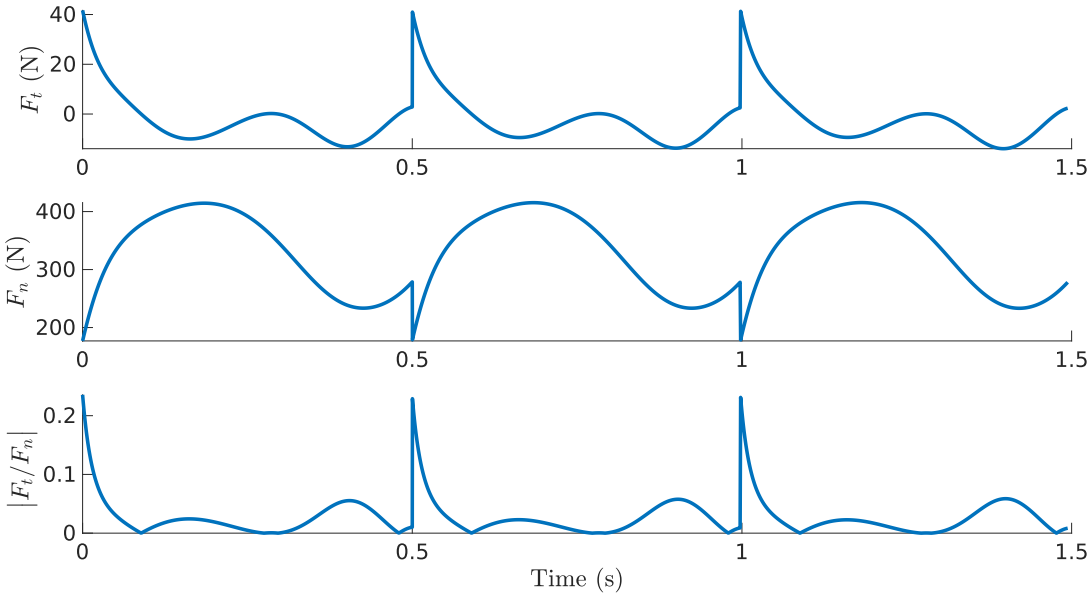

Figure 3.7: Contact force evolution at the tip of the stance leg of the walker under undisturbed conditions.

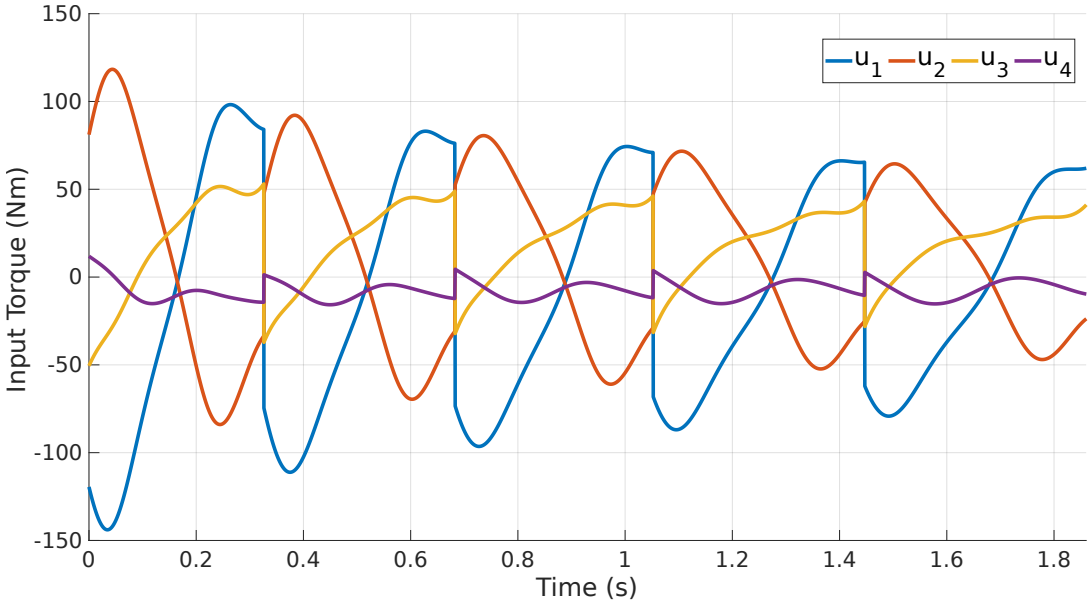

Figure 3.8: Input response of the controller when initial condition of $\dot{q}_5$, velocity of the torso, in (3.75), is disturbed by $-0.4\ rad/s$.

CHAPTER 4

MODEL PREDICTIVE CONTROL OF UNDERACTUATED BIPEDAL WALKING

So far, it has been discussed how to generate a stabilizable periodic orbit using HZD optimization and track that orbit with exponential stability using computed torque with PD control. One drawback of computed torque with PD control is that this control method does not allow input limitation. Unless one tests the trajectory against every possible source of disturbances and uncertainties, this online implementation method tends to exceed the input limitations of the actual system. Due to the complex nature of underactuated bipedal locomotion, exceeding system input limitations are usually not affordable. Exceeding torque limitations of the system causes discrepancies between the system model and the robot, errors in estimations, and instabilities or slow convergence rates. Additionally, excessive torque input usually causes a violation of the other gait restrictions, such as friction cone. In order to address this issue, this study proposes an adaptive model predictive control method as an online trajectory tracking controller that allows counting input limitations. As an online trajectory tracking controller, it requires a predefined trajectory to follow, aiming to enhance the tracking performance of the system. Since the controller is adaptive, the system dynamics approximations inside the controller change at each step to capture the actual dynamics of the walker around the present states. Adaptiveness is an essential aspect because the dynamic of the walker is highly nonlinear and the parameters are very much coupled. In order to adapt this highly nonlinear dynamics, at each time step, approximations around the present states are updated, s.t., the system model employed inside the controller changes at each time step. The proposed control method isolates commonly applied feedback linearization inputs from trajectory tracking dynamics and reduces modeling errors. Additionally, the proposed control method al-

lows considerable posture changes around the stabilizable desired orbits such that it produces realistic and applicable responses against various disturbances while obeying input limitations. Thanks to its short-horizon requirement, the controller can be implemented in real-time at 1kHz, which is crucial for high-dimensional underactuated nonlinear hybrid systems. Lastly, the controller's ability against various aggressive disturbances, where computed torque with PD control failed to maintain stability under input torque saturation, is shown in the simulation results.

4.1 Proposed Control Method

General state-space representation of a nonlinear system with configuration variables $q \in \mathbb{R}^n$ is expressed as:

$$\dot{x} := f(x) + g(x)u \tag{4.1}$$

where $x(t) = (q(t); \dot{q}(t)) \in \mathbb{R}^{2n}$,

$$f(x) = \begin{bmatrix} \dot{q} \\ D^{-1}(q)[-C(q,\dot{q})\dot{q} - G(q) + B(q)u] \end{bmatrix}$$
$$\text{and} \tag{4.2}$$
$$g(x) = \begin{bmatrix} 0 \\ D^{-1}(q)B(q) \end{bmatrix}.$$

Matrices $D = D^\top \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{n\times n}$, $G \in \mathbb{R}^n$ and $B \in \mathbb{R}^{n\times m}$ represent mass-inertia matrix, coriolis matrix, gravity vector and input matrix for m number of actuation, respectively. Using the first and second order discretization for f and g respectively, for a small enough sampling time T, the system model (4.1) can approximately be represented as:

$$x_{k+1} = x_k + (f_k + g_k\tilde{u}_k)T, \quad k = 0, \ldots, N-1 \ , \tag{4.3}$$

where for an initial condition x_0 and $t = kT$,

$$f_k = f + gu \quad \text{and} \quad g_k = \begin{bmatrix} D^{-1}BT/2 \\ D^{-1}B \end{bmatrix}. \tag{4.4}$$

With the new representation of function f, the original system dynamics becomes an autonomous system. Input u can be used to change the system dynamics, for example, via feedback linearization. If $u \neq 0$, $\tilde{u}$ constitutes a second layer controller. Otherwise, $\tilde{u}$ is the only input to the system. Note that, in order to include the effects of input to the position variables, second-order discretization is employed for function g. On the other hand, first-order discretization is employed for function f for simplicity. Using (4.3), general representation of the system becomes as,

$$x_N = x_0 + \sum_{j=0}^{N-1} f_j T + \sum_{j=0}^{N-1} g_j \tilde{u}_j T. \tag{4.5}$$

For a predefined stabilizable orbit, i.e., for a known desired state x^d, error dynamics is a basic manipulation of (4.5),

$$e_N = x_N - x_N^d = x_0 + \sum_{j=0}^{N-1} f_j T + \sum_{j=0}^{N-1} g_j \tilde{u}_j T - x_N^d. \tag{4.6}$$

As a result of (4.6), general error representation can be shown as:

$$\begin{aligned} \underbrace{\begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix}}_{E} = & \underbrace{\left(\begin{bmatrix} x_0 \\ x_0 \\ x_0 \\ \vdots \\ x_0 \end{bmatrix} + \begin{bmatrix} 0 \\ f_0 \\ f_0 + f_1 \\ \vdots \\ \sum_{j=0}^{N-1} f_j \end{bmatrix} T - \begin{bmatrix} x_0^d \\ x_1^d \\ x_2^d \\ \vdots \\ x_N^d \end{bmatrix} \right)}_{F} \\ & + T \underbrace{\begin{bmatrix} 0 & 0 & \dots & 0 \\ g_0 & 0 & \dots & 0 \\ g_0 & g_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_0 & g_1 & \dots & g_{N-1} \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} \tilde{u}_0 \\ \tilde{u}_1 \\ \tilde{u}_2 \\ \vdots \\ \tilde{u}_{N-1} \end{bmatrix}}_{U} \end{aligned} \tag{4.7}$$

where $E \in \mathbb{R}^{2n(N+1)}$, $F \in \mathbb{R}^{2n(N+1)}$, $G \in \mathbb{R}^{2n(N+1) \times Nm}$, and lastly $U \in \mathbb{R}^{Nm}$. The general error representation (4.7) enables us to write a cost function for error in a quadratic form (4.8) in terms of input. In (4.8), $N \in \mathbb{Z}^+$, $Q = Q^\top \in \mathbb{R}^{2n \times 2n} \geq 0$, $Q_f = Q_f^\top \in \mathbb{R}^{2n \times 2n} \geq 0$, $R = R^\top \in \mathbb{R}^{m \times m} > 0$ represent horizon, error weight, final error weight and input weight, respectively.

$$J = e_N^\top Q_f e_N + \sum_{k=0}^{N-1} (e_k^\top Q e_k + \tilde{u}_k^\top R \tilde{u}_k) \tag{4.8}$$

Substituting (4.7) into the general representation of the cost function (4.9) yields the cost function in terms of input (4.10). Basic manipulation of (4.10) shows that the input $U = -M^{-1}\alpha$ minimizes the error where M is invertible for $\underline{Q} = \underline{Q}^\top \geq 0$ and $\underline{R} = \underline{R}^\top > 0$.

$$J = E^\top \underbrace{\begin{bmatrix} Q & & 0 \\ & \ddots & \\ 0 & & Q_f \end{bmatrix}}_{\underline{Q}} E + U^\top \underbrace{\begin{bmatrix} R & & 0 \\ & \ddots & \\ 0 & & R \end{bmatrix}}_{\underline{R}} U \tag{4.9}$$

$$\begin{aligned} J(U) &= U^\top (\underbrace{G^\top \underline{Q} G + \underline{R}}_{M}) U + 2 \underbrace{F^\top \underline{Q} G}_{\alpha^\top} U + \underbrace{F^\top \underline{Q} F}_{\beta} \\ &= (U + M^{-1}\alpha)^\top M (U + M^{-1}\alpha) + \beta - \alpha^\top M^{-1} \alpha \end{aligned} \tag{4.10}$$

In order to define the input constraints, the minimization problem can be solved via quadratic programming (4.11). In this study, MATLAB's $quadprog$ function is utilized.

$$\min_U \{U^\top M U + 2\alpha^\top U\} \quad \text{s.t.} \begin{cases} AU & \leq b, \\ A_{eq} U & = b_{eq}, \\ lb & \leq U \leq ub \end{cases} \tag{4.11}$$

4.2 System Dynamics

Walking is a combination of two consecutive events called stance and impact (4.12), where x^- and x^+ represent states just before and just after the impact, respectively. Stance dynamics (4.1) is a nonlinear whole-body Lagrangian model of the system. The impact is a momentary event that happens when the swing leg collides with the ground (4.13), causes jumps in velocities, and triggers coordination changes, i.e., the swing leg becomes stance leg, and the stance leg becomes the swing leg. A widely used impact model for kinematic chains with multiple contact points [40] can be implemented to the bipedal walkers with some assumptions [12]. Switching surface (4.13) captures the moment when the tip of the swing leg hits the ground ahead of the

stance leg.

$$\Sigma : \begin{cases} \dot{x} = f(x) + g(x)u & x^- \notin \mathcal{S} \\ x^+ = \Delta(x^-) & x^- \in \mathcal{S} \end{cases} \tag{4.12}$$

$$\mathcal{S} := \{(q, \dot{q}) \in T\mathcal{Q} \mid p_2^v(q) = 0, \;\; p_2^h > 0\} \tag{4.13}$$

4.3 Trajectory Generation Using HZD

Hybrid Zero Dynamics (HZD) is a well-known stabilizable trajectory generation method for underactuated hybrid 5-link bipedal walkers [12, 43]. The method ensures that for a certain subset of the configuration variables $q \in \tilde{\mathcal{Q}} \subset \mathcal{Q}$, there exists a set of input such that the output,

$$y = h(q) := h_0(q) - h_d \circ \theta(q) \tag{4.14}$$

is zero, where $h_0(q)$ specifies $n-1$ independent quantities that are to be controlled and $h_d \circ \theta(q)$ specifies the desired evolution of these quantities as a function of a scalar quantity $\theta(q)$, which is called internal clock. The internal clock is a monotonic one-to-one function that slaves the desired states, and a combination of configuration variables constitutes it, $\theta(q) = cq$, where $c \in \mathbb{R}^{1\times n}$. Using HZD, one can fit a degree M Bézier polynomial, $b_i : [0, 1] \to \mathbb{R}$, for the desired state evolution $h_d \circ \theta(q)$. The desired state evolution mimics walking behavior and satisfies defined constraints during HZD optimization process. The study in [39, Chapter 6] provides more throughout explanation on obtaining stabilizable periodic trajectory for such systems.

4.4 Controller Implementation

The existence of a stabilizable orbit guarantees a solution around that trajectory and allows the employment of a model-based controller. The proposed control method requires the error to be defined for each joint, but the error definition (4.14) in HZD covers only $n-1$ independent quantities, which are the controlled joints. To obtain error information for the uncontrolled joint, one can combine the desired state evolution information, the instant actual state information, and the internal clock. Both

system states and desired states share the same internal clock. As a result, the solution of (4.15) for the uncontrolled desired state completes the required error information.

$$\theta = cq = cq^d \quad \text{and} \quad \dot{\theta} = c\dot{q} = c\dot{q}^d \tag{4.15}$$

For each time step, required operations to apply the controller can be listed as:

- Using (4.3) and (4.4), for $\tilde{u} = 0$, simulate ahead up to horizon N and store f_k and g_k matrices,
- Using (4.15), for each simulated time step, complete the missing error information for the unactuated joint.
- Using (4.7), (4.9) and (4.10), generate G and F, then obtain M, $\alpha^\top$ and β,
- Using (4.11), solve for input.

4.5 Parameter Selection

Parameter selection for the proposed adaptive model predictive controller is not intuitive as it was for zero dynamics control in section 3.6.1. Different reasons cause this difference. First of all, in zero dynamics control, it was shown that the PD part of the controller is applied on top of the feedback linearization (3.70). That is why it is evident that, as long as there is an approximate balance between derivative and proportional control parameters, almost any parameter selection would work when there is no disturbance or uncertainty. In this particular case, the control input is applied directly to the nonlinear dynamics of the walker. That is why making an intuitive guess is harder. Additionally, it was shown that for the PD part of the zero dynamics control, there were eight parameters to select (3.72). For the adaptive model predictive control, on the other hand, the required number of selections is 14 (or 24 if $Q_f \neq Q$), which are error weights and input weights (4.8). Even though the same search algorithm introduced in chapter 3.6.1 could be used, it would take too much time. Additionally, parameter increase rate selection is not that obvious in this particular control method in the search algorithm. Even though the search algorithm could be run with different incremental rates, again and again, this is not preferred.

Since the parameter selection in the PD part of the zero dynamics control (3.70) gives a reasonable ratio between joints, almost the same ratio is used. For $R = I_{n-1\times n-1}$, the error weights are selected to be:

$$Q = Q_f = 10^5 \times diag(6 \cdot 10^3, 2 \cdot 10^4, 9 \cdot 10^3, 5 \cdot 10^3, 6 \cdot 10^3, 1, 2, 2, 1, 1) \quad (4.16)$$

It is important to note that ratios between gains related to position errors are almost the same as those in proportional control. Similarly, ratios between the gains related to velocity errors are the same as those in derivative control. For example, the gain related to the error in $\dot{q}_2$ is double of the gain related to the error in $\dot{q}_1$. The only difference is in the gain related to the error in q_2. It is slightly increased to enhance the forward disturbance rejection of the walker. In case of a forward disturbance, the swing leg should be moved forward quickly to prevent the walker from falling forward. Additionally, different from the gains introduced in the PD part of the zero dynamics control, there is an extra 10^3 multiplier between the gains related to the position and velocity errors. The extra $T/2$ multiplier in the discretization (4.4) explains the order differences between weights of the position and the velocity errors in Q. Lastly, it is mentioned that this control method requires extra weights related to the errors in position and velocity of the torso, i.e., the unactuated joint. Since the torso affects the posture of the walker as the stance leg does, the weights of the errors related to the torso are selected to be the same as the weights of the errors associated with the stance leg.

Horizon N selection is another important decision. First of all, it is mentioned that the dynamic model of the walker is updated at each time step. This update is made via simulating the discretized system dynamics (4.3) without any input ($\tilde{u} = u = 0$), i.e., effect of input is neglected while obtaining system dynamics at each time step (see Chapter 4.4). Since the effect of input is neglected, selecting the horizon too high causes considerable deviation between estimated and actual dynamics. Additionally, it increases optimization time. On the other hand, selecting it to be too low decreases the controller's performance. In order to find a good balance, N is selected to be five. This selection provides good disturbance rejection performance, and it allows control of the system in 1kHz frequency. For the same parameter selection, selecting N to be ten increases disturbance rejection performance slightly, but solution time increases. It should be noted that the controller's performance for different horizon

selections is hard to comment on due to the system's complexity. Additionally, different parameter selections could work differently for each horizon selection. Parameter selection is a general drawback of this method, which will be discussed extensively in the discussion chapter.

Unlike the approach followed in this study, the horizon of the model predictive control usually contains two variables called the control horizon and prediction horizon. The control horizon indicates the number of time steps to be controlled with changing input. On the other hand, the prediction horizon indicates the number of further time steps to be observed with a constant input. An increase in the control horizon increases the required amount of computation to calculate the system's input. On the other hand, since there is no new input calculation during the prediction horizon, it is used to obtain a more informative cost function [44, Chapter 20]. Usually, introducing a prediction horizon after the control horizon increases the controller's performance without additional significant computation requirements. In this study, a single horizon notion is employed by the selection. This approach is listed as a future study and discussed in Chapter 7.1.

4.6 Simulation Results

In order to test the performance of the proposed control method on a 5-link planar underactuated bipedal walking, the same simulation environment in the previous chapter is utilized. Similar to the discrete nature of the computers, the controller is implemented in discrete time, at 1 kHz. The proposed control method is tested against various disturbances such as large modeling errors in body weight, aggressive initial condition errors, pushing and pulling torso throughout a step, input noises, and their combinations.

Fig. 3.2 and 4.1 show the posture of the walker around the optimized trajectory and resultant input calculation of the proposed controller, respectively. Fig. 4.1 shows that the hip torques (u_1 and u_2) are limited to $\pm 75Nm$ in magnitude, whereas the knee torques (u_3 and u_4) are limited to $\pm 50Nm$. When there is no disturbance, input torques are within the limits because trajectory generation optimization already

considers these limits. This undisturbed system response will be used as a comparison basis for other simulation scenarios.

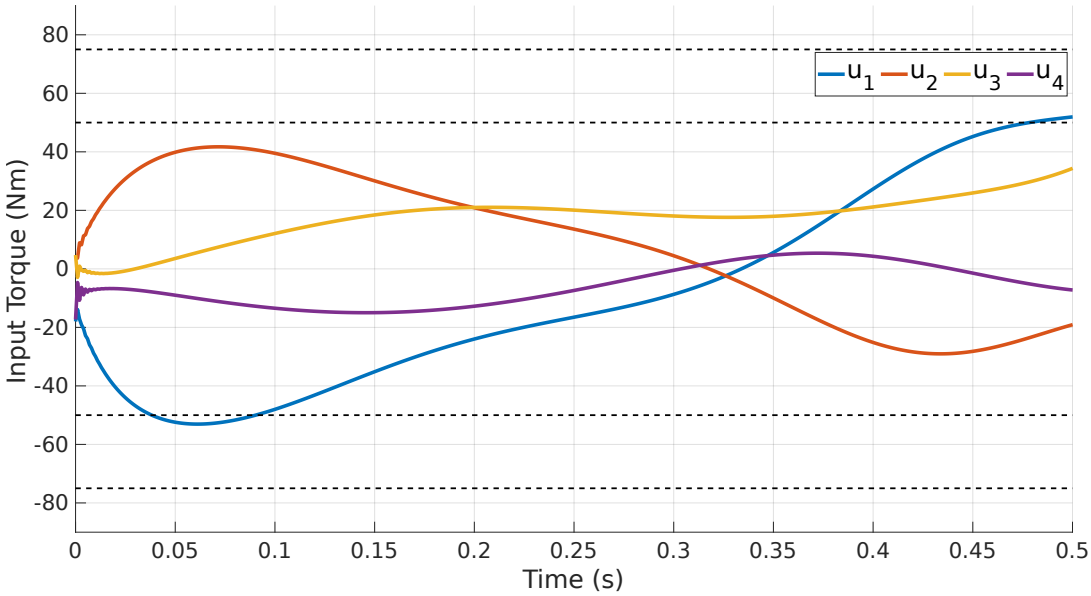

Figure 4.1: Input response of the proposed control method to the optimized trajectory without any disturbance. Dashed lines at $\pm 50Nm$ and $\pm 75Nm$ indicates input limits of the knee and the hip joints, respectively.

Trajectory tracking performance of the controller is tested under an aggressive initial condition error that is introduced in the position of the torso, q_5. Any disturbance introduced in the torso can be considered aggressive since it is the unactuated, the heaviest, and the longest joint. Additionally, the position and velocity of the stance and swing legs are defined relative to the torso (see Fig. 2.1). As a result, any position error in this joint displaces legs from their desired positions, too. That is why the angle of the torso is disturbed by $+0.1$ radian (≈ 6 degrees). Fig. 4.2 shows that the controller changes the posture of the walker such that the center of mass of the system is shifted toward the right in order to compensate for this aggressive initial condition error. This posture change helps the system fall forward using its weight and reduces input requirements. Fig. 4.3 depicts the limited input torque response of the controller and concludes that the input tends to exceed the limits under disturbance.

Generally, model-based controllers are vulnerable to modeling errors. In order to show the robustness of the controller against modeling errors, we simulated the sys-

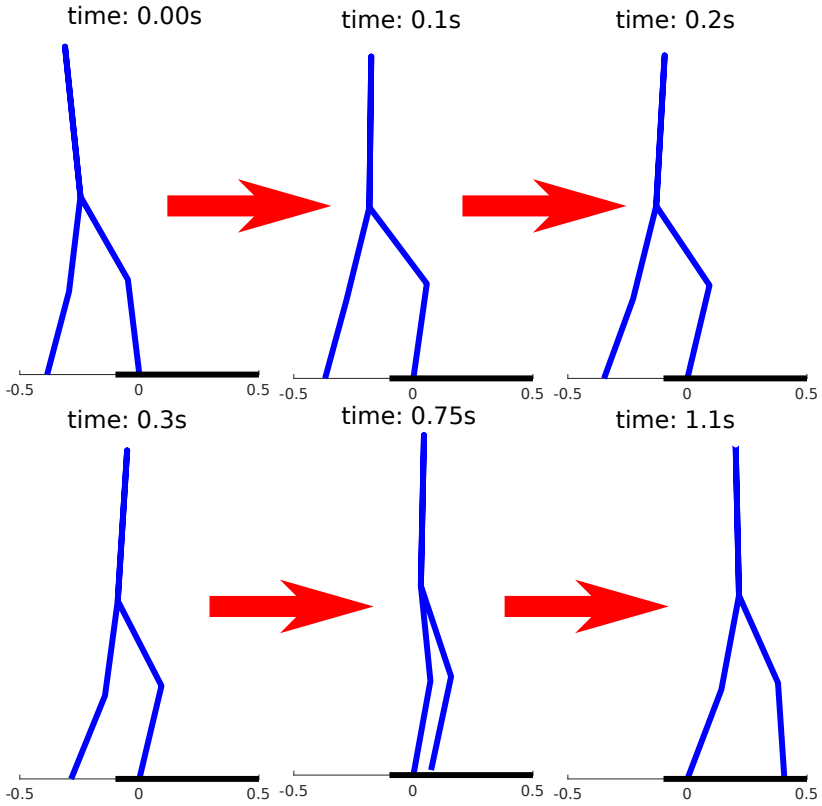

Figure 4.2: Posture of the walker throughout a step with an initial condition error introduced in the position of the torso.

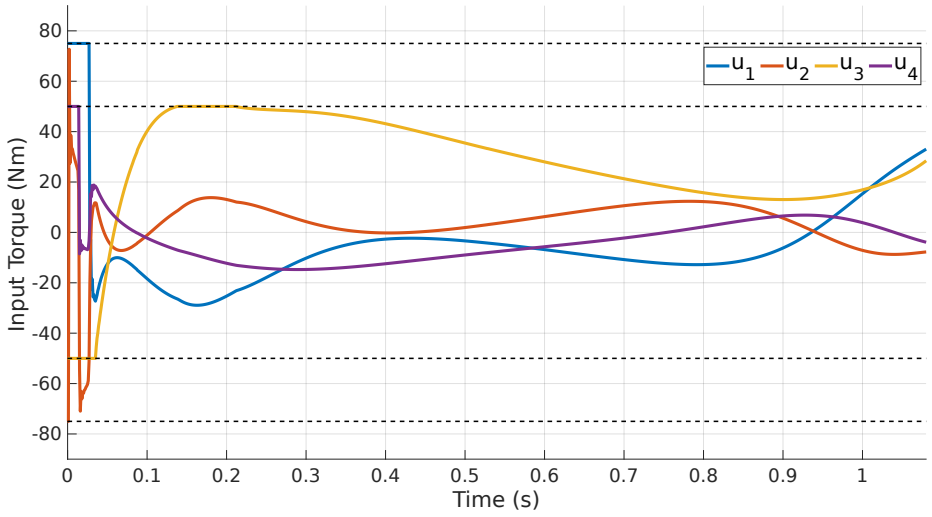

Figure 4.3: Input response of the controller against 0.1 radian ($\approx$ 6 degrees) initial condition error in the position of the torso.

tem with a modeling error along with an initial condition error in the torso by multiplying its weight by two and adding 0.07 radian (≈ 4 degrees) to its initial condition, q_5, inside the stance and impact dynamics of the walker. Fig. 4.4 shows the resulting response against these disturbances. The posture evolution of the walker followed a similar pattern with the one shown in Fig. 4.2.

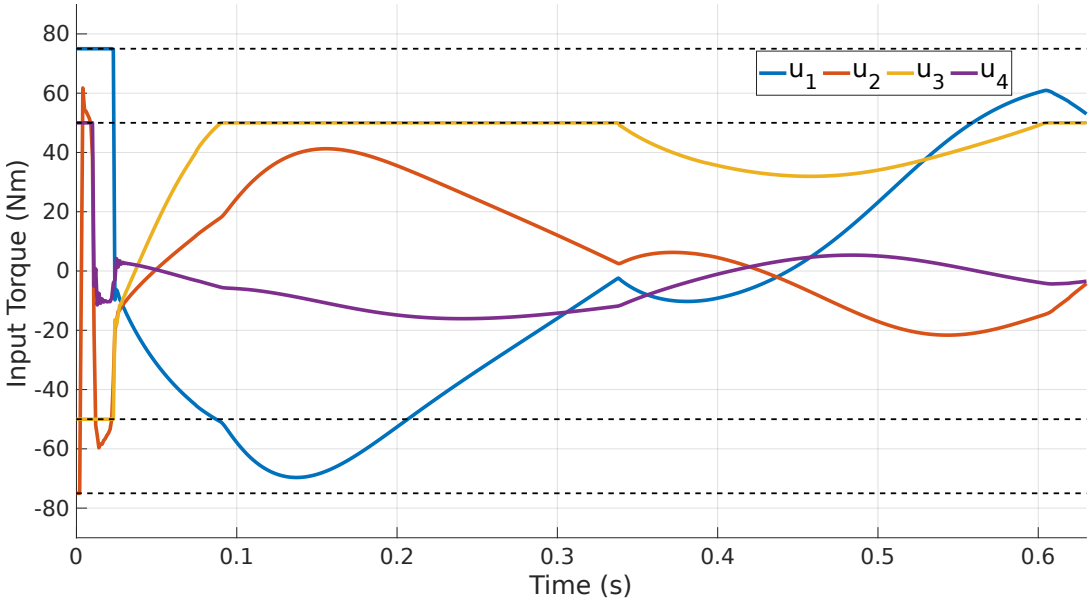

Figure 4.4: Input response of the controller against 0.07 radian (≈ 4 degrees) initial condition error in the position of the torso along with a modelling error introduced in the torso weight.

Finally, the performance of the proposed controller is compared with the local PD controller around restriction dynamics. Fig. 4.5 shows the two-step response of the controllers against a forward force of $40N$ to the torso during the first step. The figure shows that the local PD controller applies much more input than the proposed control method, even with the fine-tuned parameters. The system with the local PD controller failed to maintain stability when there was saturation. Fifteen-step error responses of the systems to the same disturbance are shown in Fig. 4.6 and 4.7. From the figures, it can be seen that outputs of both closed-loop systems are converged to around zero after a few steps. The proposed controller managed it with much less input torque usage in magnitude. Since the adaptive model predictive controller is implemented on top of the full-order system model and in discrete time with some approximations,

a slight steady-state error remains. On the other hand, since the zero-dynamics with PD control is also applied in discrete time, the slight steady-state error remains. The velocity error evolution shows that since the joint velocities are too high due to the aggressive forward disturbance, the impact causes high magnitude jumps. Additionally, due to the different characteristics of the controllers, error evolution shows different behavior. For example, the disturbance causes high error in the swing knee position at the beginning of the closed-loop system with zero dynamics control. On the other hand, the disturbance causes a high magnitude error in stance leg position at the beginning of the closed-loop system with adaptive MPC. It is mainly because the torque is limited in adaptive MPC controller but not in zero dynamics control. And since the stance leg requires high input torque against this disturbance, the system with limited torque exhibits high deviation in the position of the stance leg at both in hip and knee at the beginning of the disturbance. Since the simulated walker model is deterministic, the system with zero dynamics with PD control has a smaller steady-state error. The calculated input torque is exact, and the PD control helps the output converge to around zero. On the other hand, the adaptive MPC is implemented on the full order system and calculates the input via solving an optimization problem. Due to the parameter selections, it showed a slight more steady-state error in magnitude. It is a drawback of this controller compared to the zero dynamics control. The parameter selection has to be made on full-order nonlinear system dynamics. As a result, parameter selection is harder. But, in the following chapters, it is shown that when there is an error in the system model, since the exact torque calculation in zero dynamics control directly depends on the system model, the performance of the closed-loop system with zero dynamics control decreases dramatically.

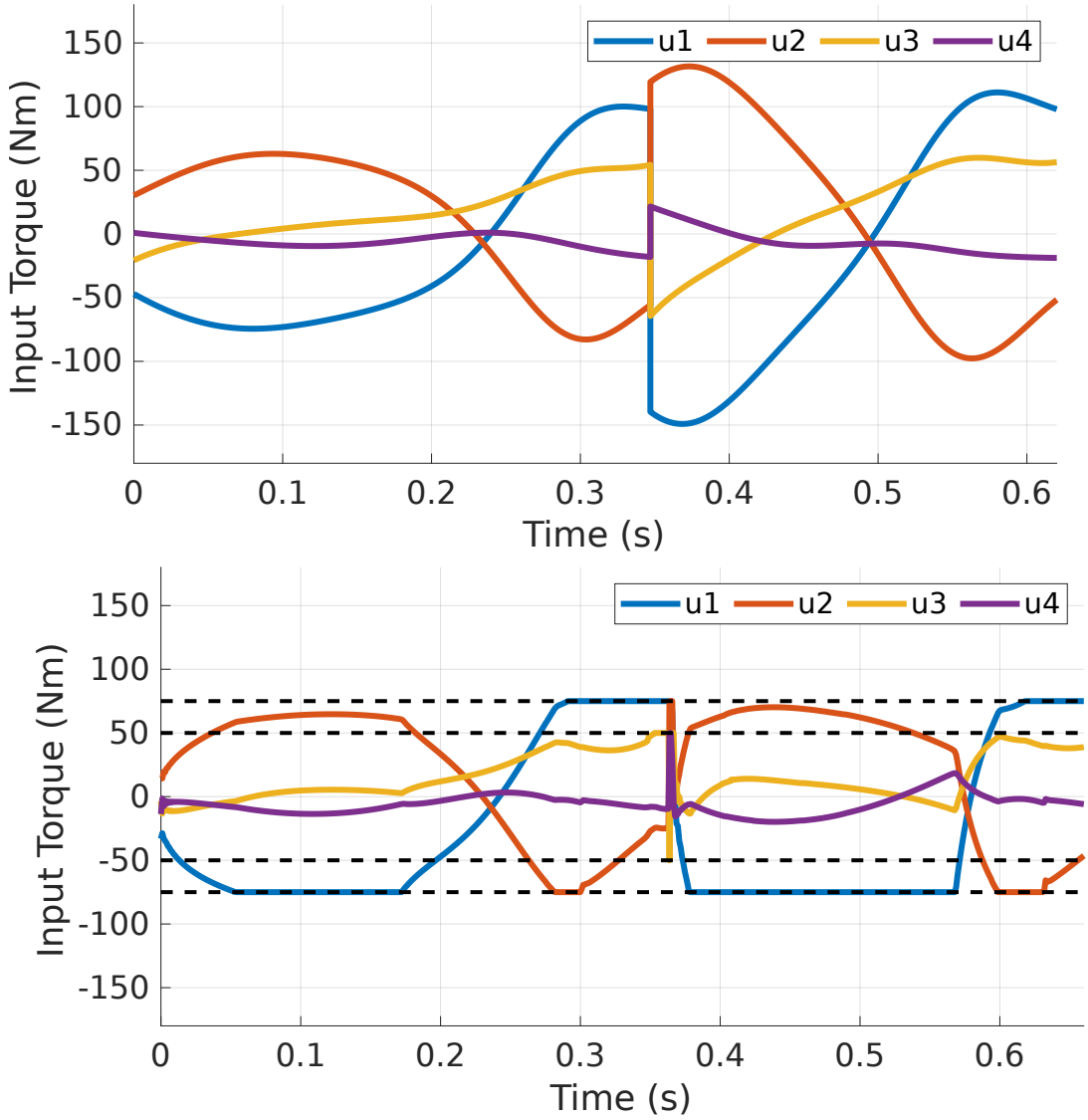

Figure 4.5: Two-step response of the PD controller (top) and the proposed controller (bottom) against a forward force of 40N to the torso throughout the first step of the robot.

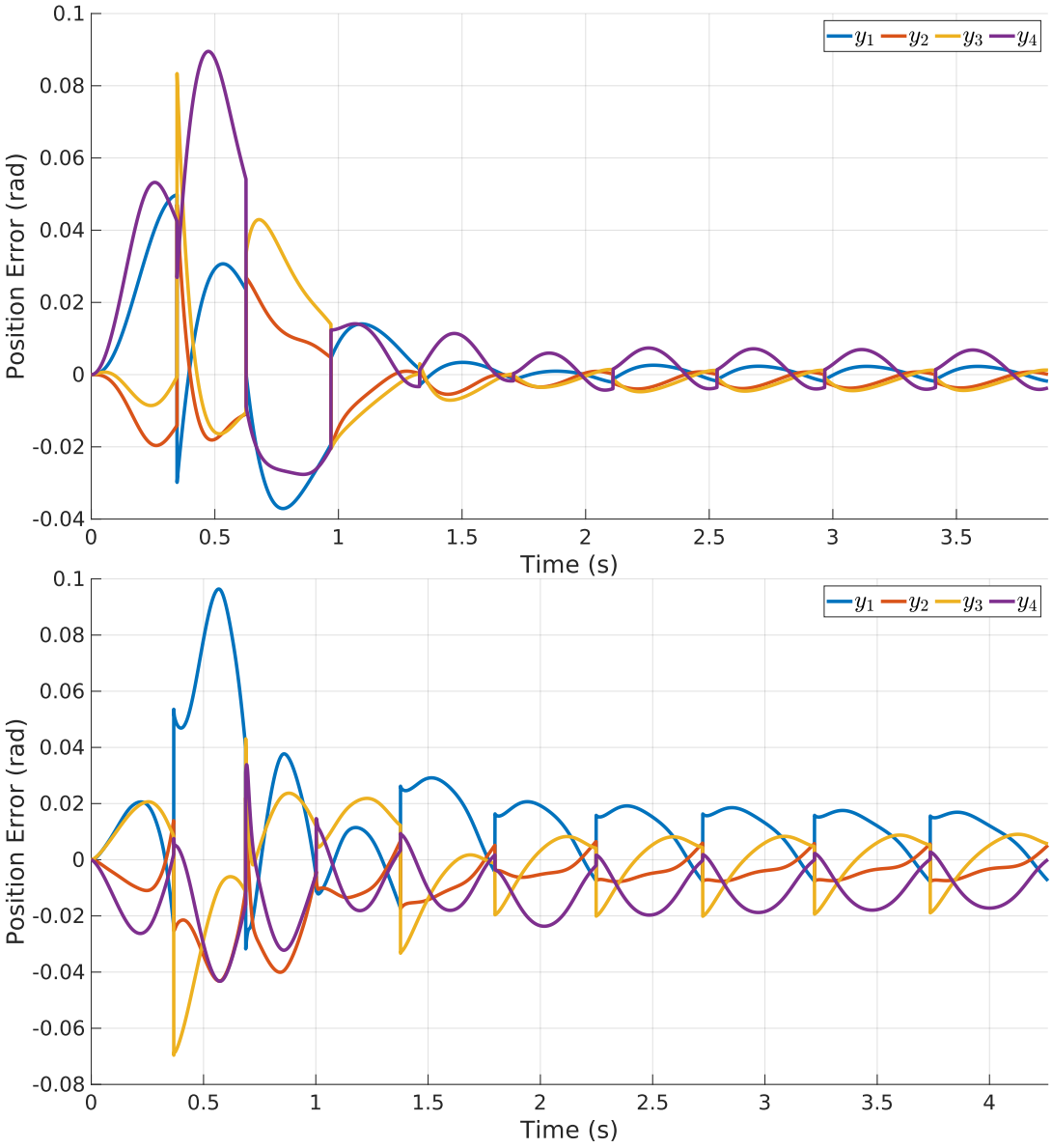

Figure 4.6: Ten-step position error response of the PD controller (top) and the proposed controller (bottom) against a forward force of 40N to the torso throughout the first step of the robot.

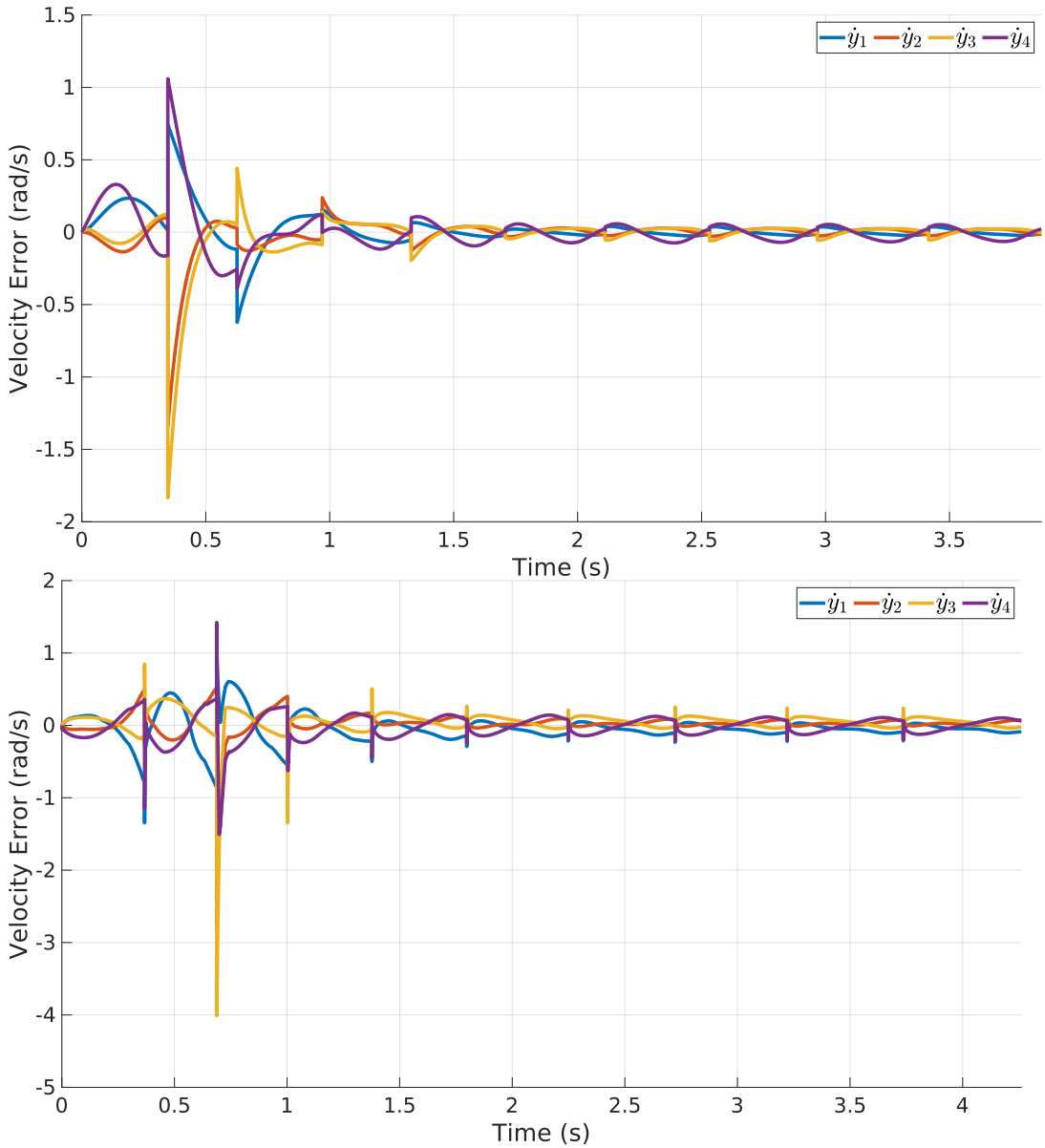

Figure 4.7: Ten-step velocity error response of the PD controller (top) and the proposed controller (bottom) against a forward force of 40N to the torso throughout the first step of the robot.

CHAPTER 5

POWER OPTIMAL CONTROL

In the previous chapter, input limited optimal control of the bipedal walker is studied. The motivation was that in real-life applications, actuators saturate such that after a certain amount of torque, the input torque cannot increase. Even though this is a reasonable motivation, the amount of torque that can be supplied to the system changes by velocity too. For example, the maximum torque, which is usually called peak torque or stall torque, that an actuator can supply is when it is not rotating, and it can only be supplied for a short amount of time [45]. Due to power limitations, as the velocity increases, the amount of torque that can be supplied decreases. In order to cover this issue, one can replace the input cost with a power cost in (4.8), such that

$$J = e_N^\top Q_f e_N + \sum_{k=0}^{N-1} \left(e_k^\top Q e_k + (\tilde{u}_k \cdot \dot{q}_{0,1-4})^\top R(\tilde{u}_k \cdot \dot{q}_{0,1-4})\right) \tag{5.1}$$

where $\dot{q}_{0,1-4}$ represents $(\dot{q}_1; \dot{q}_2; \dot{q}_3; \dot{q}_4)|_{k=0}$. It is important to note that, independent of horizon N, input weight is multiplied by the momentary actual velocity. It is preferred not to disturb the linearity of the cost function such that the same solution sequence introduced in the previous chapter can be employed. It is also important to note that the input weight of the cost function changes at each time step depending on the momentary velocity of the system. Since the input weight changes depending on the velocity of the actuated joints, when velocity is small, it may cause oscillations and abnormalities. When input weight is too small, the input cost decreases too much, and the controller tends to increase the magnitude of the input torque dramatically. In order to prevent this, a lower limit can be introduced. In this study, the minimum value for each member of $(\tilde{u}_k \cdot \dot{q}_{0,1-4})^\top R(\tilde{u}_k \cdot \dot{q}_{0,1-4})$ is set to be one.

In order to present behavioral differences between power and input optimization, error

and final error weight matrices, which are Q and Q_f, respectively, are selected to be identical. Differently, only input weight is scaled by momentary velocities of the corresponding actuators such that the input weight changes at each time step. It should be noted that the parameter selections are not absolute. Even though some well-educated guesses are made and some proper mathematical notations are used, it is clear that the selected parameters do not guarantee the best performance. Further discussions on this can be found in Chapter 7.1.

5.1 Simulation Results

In order to test the performance of the power optimal control, the same simulation environment is employed. Fig. 5.1 depicts that power optimal control reduces the maximum required power input from the actuators when there is no disturbance. In model predictive control, maximum power required at the hip (p_1 and p_2) is observed to be around $83\ Watts$. On the other hand, in power optimal control, the maximum power required at the hip is around $67\ Watts$. Hence, using power optimal control, the sizing of the hip actuators can be reduced. No dramatic change is observed in knee actuators (q_3 and q_4).

The difference between two cost functions (4.8) and (5.1) is more apparent under forward speed disturbances. Fig. 5.2 and Fig. 5.3 show two-step power and input response of the controllers against $20N$ forward disturbance throughout the first step, respectively. The figures show that the maximum required power at the hip is reduced from $221\ Watts$ to $145 Watts$. Similarly, no dramatic change in knee actuators is observed. Fig. 5.3 indicates that as the velocity increases, the power-optimal control reduces the input torque. Similarly, it tends to use more torque when joint velocity is small. This tendency defines distinctions between the input torque profiles with the input minimized model predictive control. Fifteen-step error responses of the systems to the same same disturbance are shown in Fig. 5.4 and 5.5. Since the forward disturbances usually require high torques at high velocities, forward disturbance rejection performance of the system with power-optimal control decreases. From the error responses, it can be seen that it takes more steps for the system with power-optimal control to converge. It takes eleven steps for the closed-loop system with

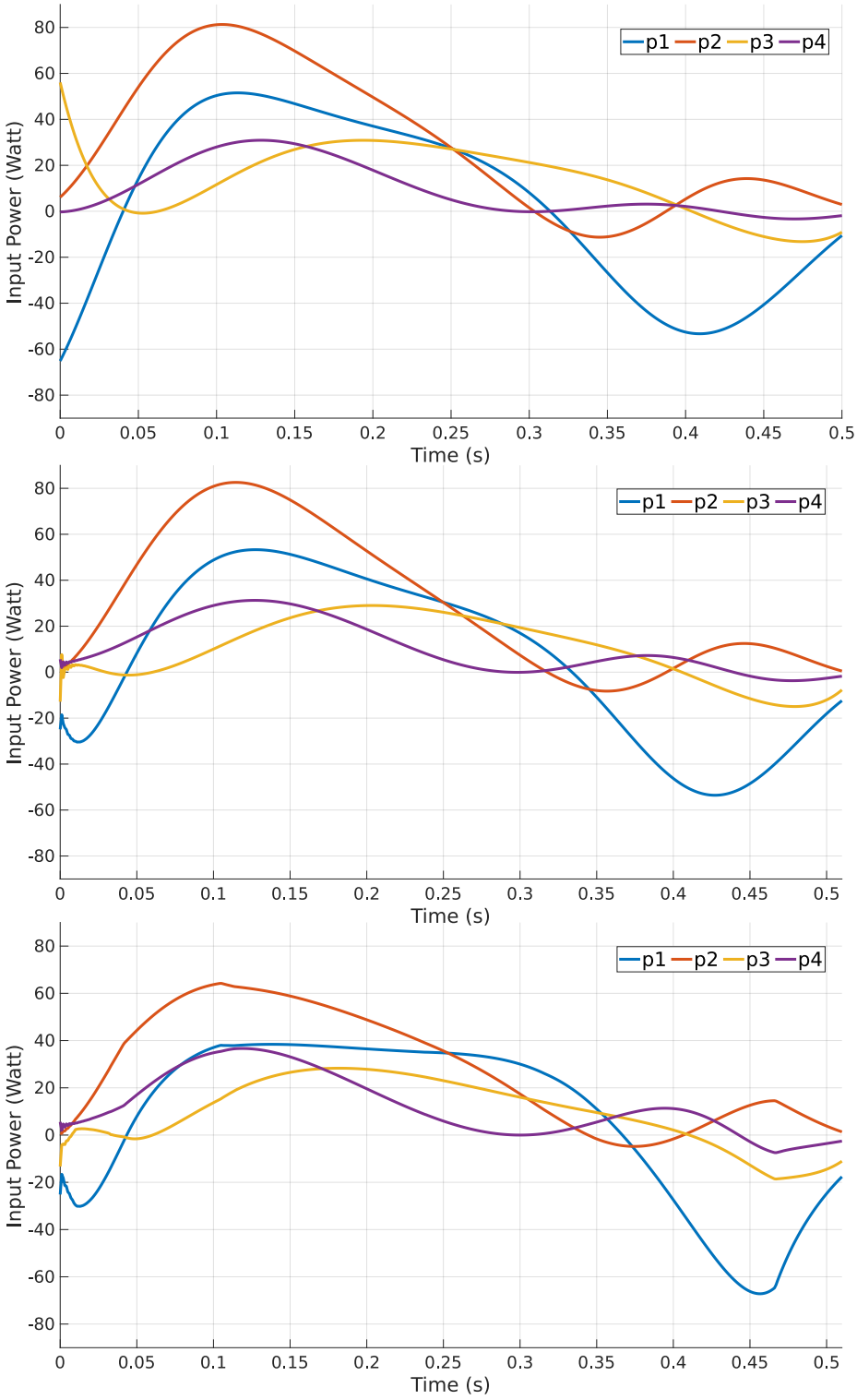

Figure 5.1: Required power profile calculated by three different controllers throughout a step without any disturbance. From top to bottom, the figure shows response of computed torque control (3.70), model predictive control (4.11), and power optimal control (5.1), respectively.

power-optimal control to converge to its steady state. On the other hand, the closed-loop system with adaptive MPC converges to its steady-state in five steps. Since both controllers are implemented on top of the full-order system model and in discrete time with some approximations, a slight steady-state error remains. The only difference in weight selections of the adaptive MPC and power optimal control was the input weight. The input weight of the power optimal control changes at each time step depending on velocity evolution. These input weight differences cause behavioral differences in error characteristics between these two controllers. Finally, since it is an aggressive forward disturbance, high magnitude jumps in velocity error evolution are observed until the controllers slow the walker down.

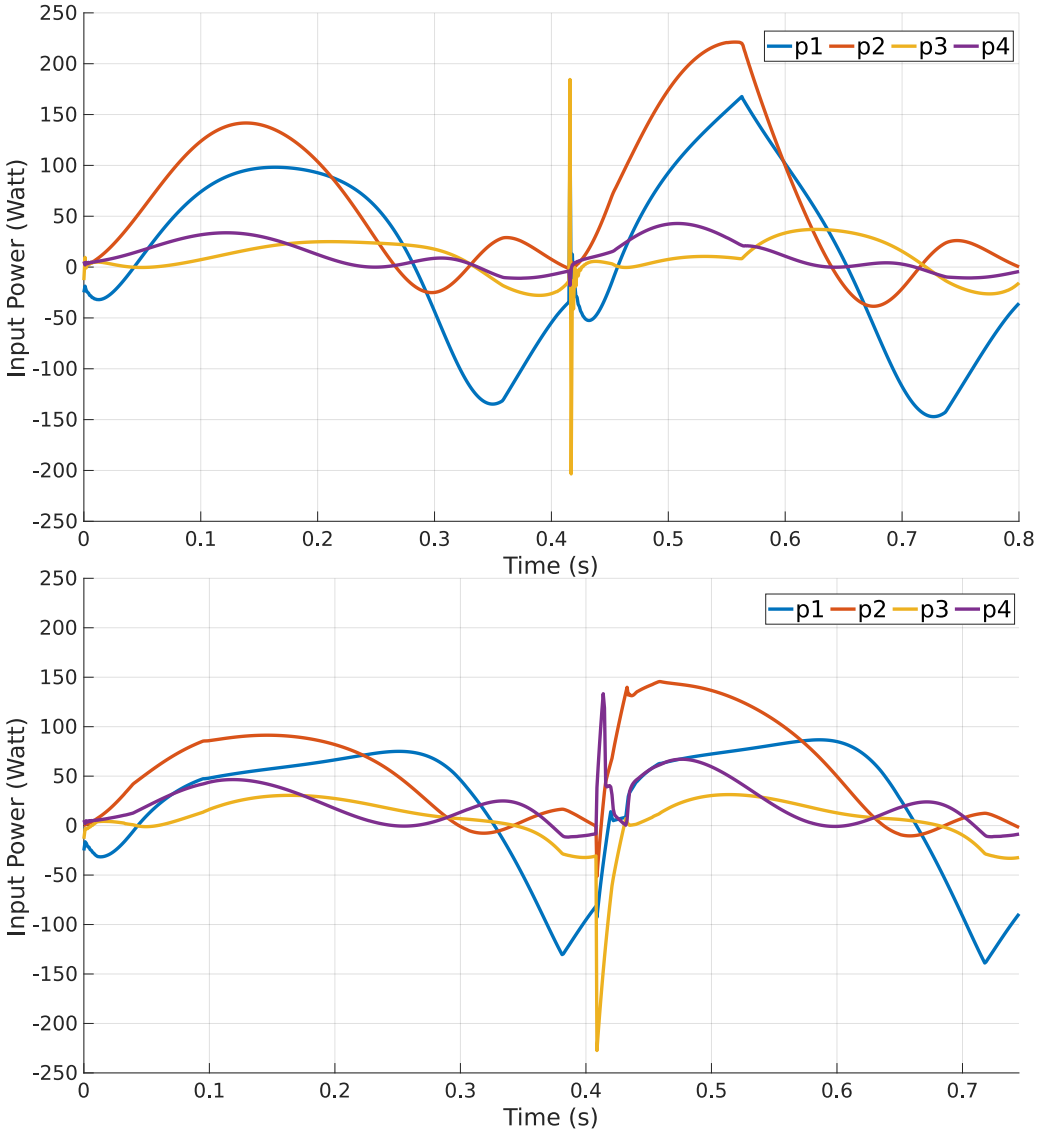

Figure 5.2: Two step power response of the controllers when $20N$ forward disturbance from the CoM of the torso is applied throughout the first step of the walker. From top to bottom, the figure shows response of model predictive control (4.11) and power optimal control (5.1), respectively.

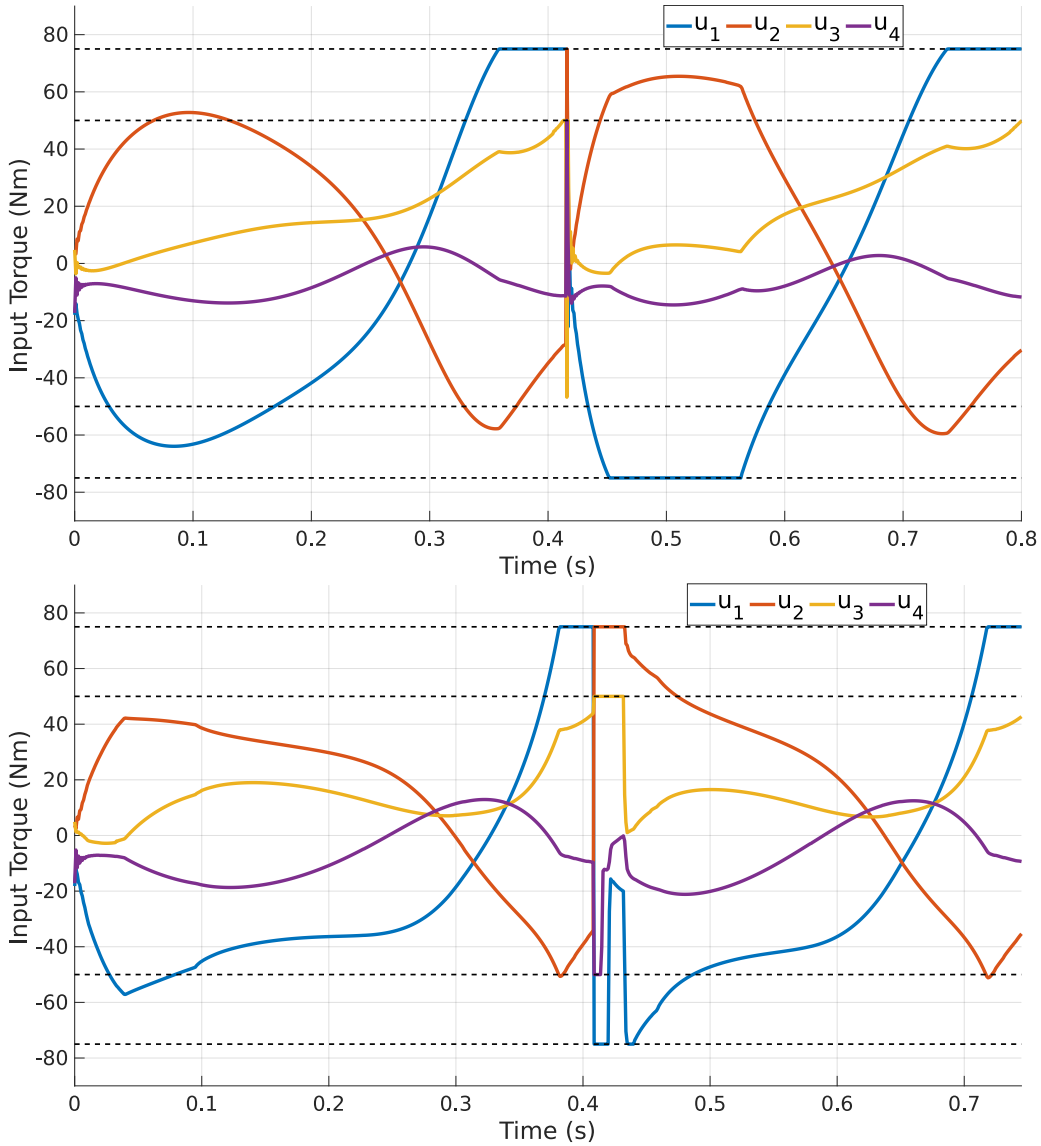

Figure 5.3: Two step input response of the controllers when $20N$ forward disturbance from the CoM of the torso is applied throughout the first step of the walker. From top to bottom, the figure shows response of model predictive control (4.11) and power optimal control (5.1), respectively.

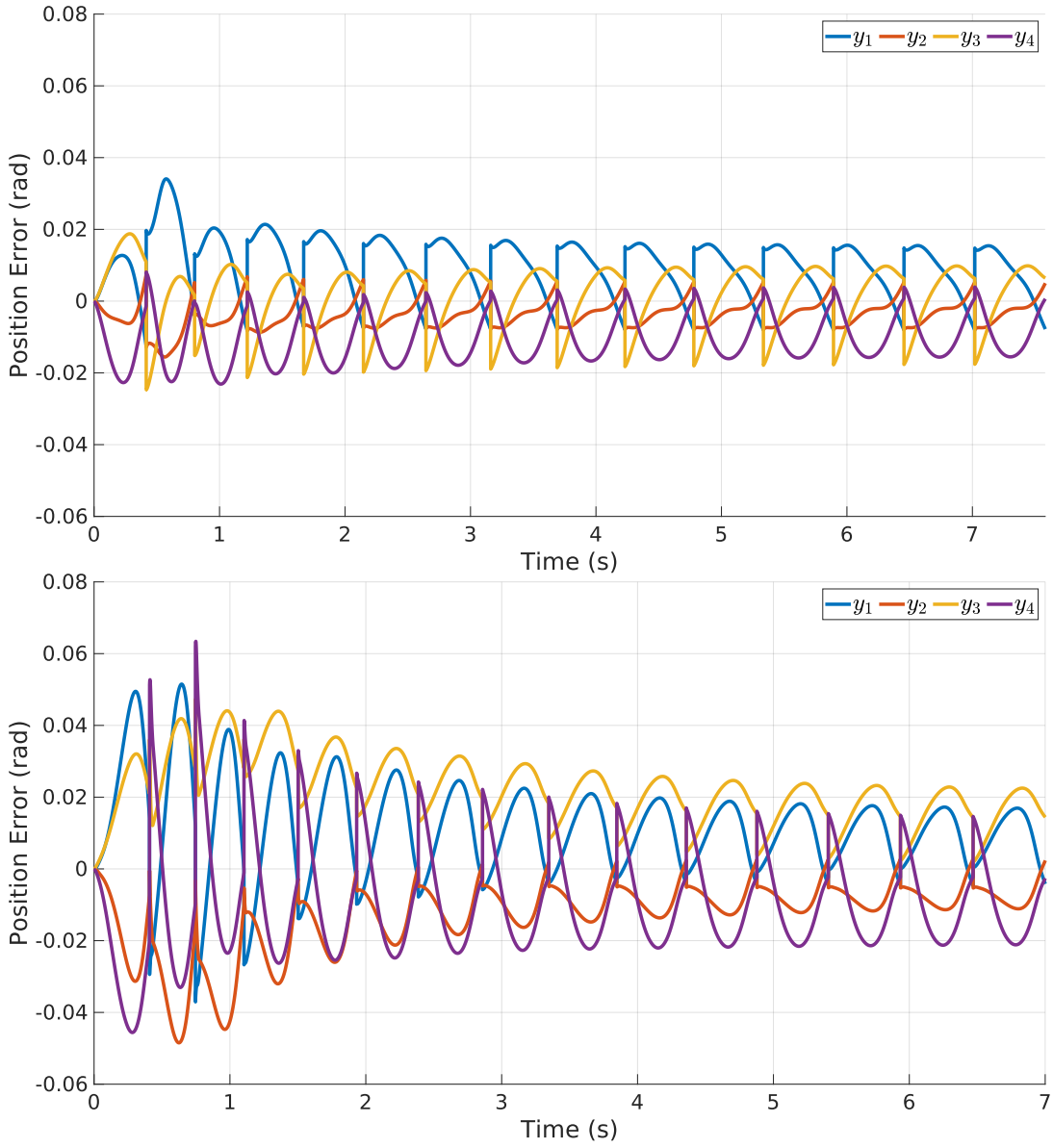

Figure 5.4: Fifteen-step position error response of the model-predictive controller (top) and the power-optimal controller (bottom) against a forward force of 20N to the torso throughout the first step of the robot.

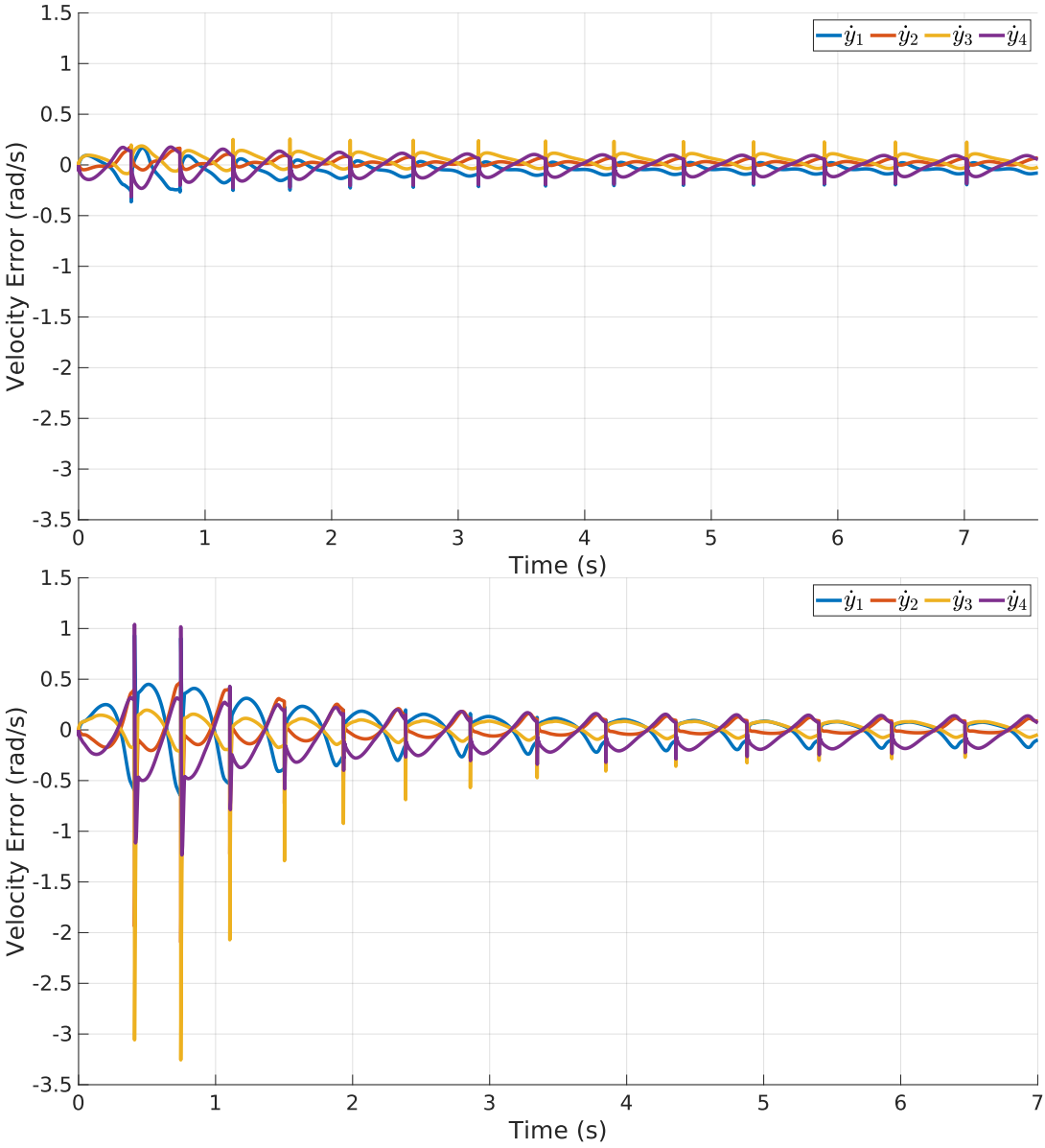

Figure 5.5: Fifteen-step velocity error response of the model-predictive controller (top) and the power-optimal controller (bottom) against a forward force of 20N to the torso throughout the first step of the robot.

CHAPTER 6

PERFORMANCE OF THE CONTROLLERS

The previous chapters investigate the behavioral and characteristics differences between zero dynamics control, adaptive model predictive control, and adaptive power optimal control. This chapter employs some performance metrics to compare the controllers. This chapter compares the basin of attraction of each controller along with their closed-loop Poincaré return map eigenvalue estimations. A basin of attraction surface is generated for initial condition disturbances at the position and velocity of the torso. The disturbances are introduced in the torso because it is the heaviest, the longest, and the unactuated joint. Along with the undisturbed case, in order to discuss the effects of modeling errors on the controller performances, the Poincaré return map estimations are also done with a modeling error introduced in the mass of the torso by multiplying it by two.

6.1 Basin of Attraction Analysis

The basin of attraction of the controllers is calculated for torso angle q_5 and its velocity $\dot{q}_5$. Torso angle and its velocity are selected since it is the heaviest, the longest, and the unactuated joint of the walker. During the calculation, q_5 and $\dot{q}_5$ are disturbed inside the initial state condition of the system (3.75). Since the input is constrained in optimal controllers, during basin of attraction calculation, the input torque that the zero dynamics controller calculates is limited to $75Nm$ at the hips (u_1 and u_2) and $50Nm$ at the knees (u_3 and u_4). Additionally, since the states of the walkers that did not fall until the sixth step converged or came very close to their desired trajectories, the simulation is terminated after six steps. Since the exact convergence is not

guaranteed, the figure can also be named a six-step failure analysis. It shows the collection of disturbance configurations that the controllers managed to proceed at least six steps without falling. Fig. 6.1 shows basin of attraction for all three controllers with respect to changes in initial conditions of q_5 and $\dot{q}_5$. From the figure, the basin of attraction of input saturated zero dynamics control (top) is the smallest. On the other hand, the basin of attraction of the adaptive model predictive control (bottom) is the widest compared to the other two. Since the power-optimal control does not allow high torque input at high velocities, it cannot maintain stability under high magnitude forward velocity disturbances. As a result, the basin of attraction of power optimal control (middle) is wider than the basin attraction of the zero dynamics control but smaller than the basin of attraction of the adaptive model predictive control. It is understandable to observe the smallest basin of attraction for zero dynamics control because the controller does not consider input saturation. As shown in previous discussions, it should be noted that, even though they all seem stable in some regions, each closed-loop system with different controllers handles disturbances differently. Input profiles and posture evolution are distinct for each method. Additionally, it should be noted that unrealistic movements are eliminated during the simulation. For example, if leg scuffing occurs before the swing leg passes through the stance leg, the simulation is terminated, and the behavior is marked as unstable.

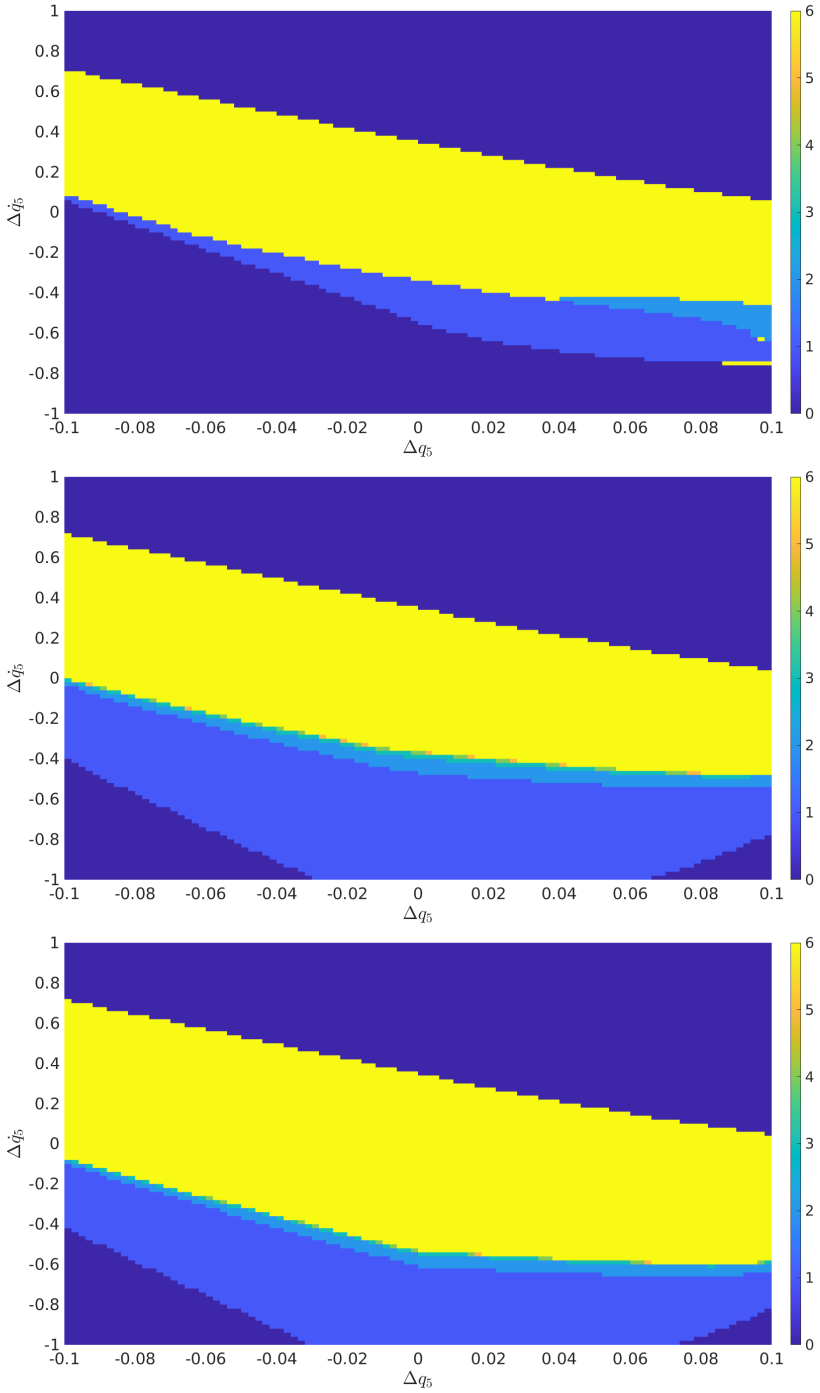

Figure 6.1: Basin of attraction of all three controllers with respect to changes in initial conditions of q_5 and $\dot{q}_5$. From top the bottom the figure shows feedback linarization control with input saturation, power optimal control, and model predictive control. The colorbar indicates number of steps the walker took and the simulation is terminated after six steps.

6.2 Poincaré Return Map Analysis

Gait stability can approximately be determined by analyzing the eigenvalues of Poincaré return map of the system, which is linearized about a fixed point [46, 47, 48, 49]. Poincaré return map transforms the problem of finding periodic orbit into finding a fixed point of the map, which is a discrete-time nonlinear system. If the Poincaré section is selected to be the switching surface $\mathcal{S}$, then the fixed point x^* lies within $\mathcal{S} \cap \mathcal{Z}$ and the resultant Poincaré map definition becomes $P : \mathcal{S} \cap \mathcal{Z} \rightarrow \mathcal{S} \cap \mathcal{Z}$. For $P(x_k) = x_{k+1}$ periodicity is achieved only if $P(x^-) = x^-$. A periodic stable fixed point satisfies that, in its neighbourhood, all eigenvalues of the Jacobian matrix

$$\mathbf{DP} = \frac{\partial f_{cl}}{\partial x_k}(x^*) \tag{6.1}$$

lie within a unit circle ($|\lambda| < 1$) [50], where f_{cl} indicates closed-loop system dynamics. A sketch for the return map is shown in Fig. 6.2. The figure shows that for an initial condition x^-, the states undergo an impulsive impact. Then the closed-loop stance dynamics take place until the next intersection with the switching surface occurs.

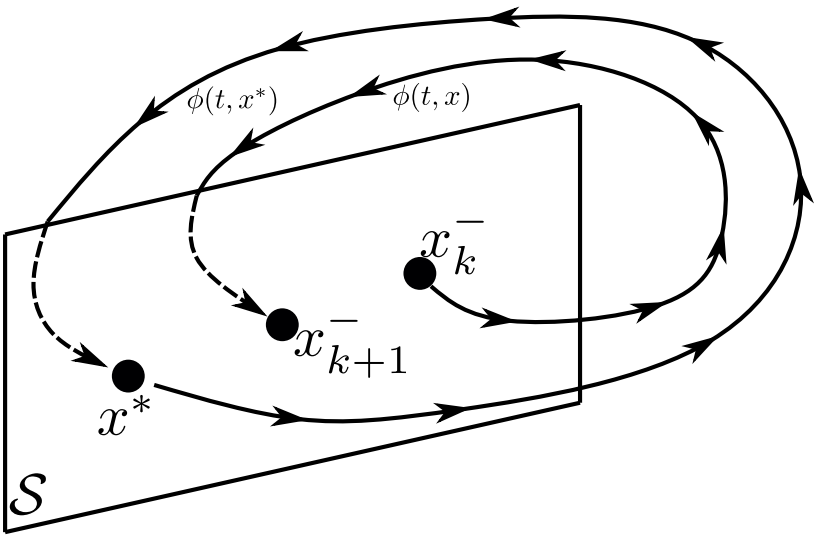

Figure 6.2: Stride-to-stride return (Poincaré) map of the system where switching surface $\mathcal{S}$ is selected as Poincaré section. The dynamics starts from x^-, undergoes an impact $x^+ = \Delta(x^-)$, and evolves into the next x^- in accordance with closed-loop stance dynamics.

Since there is no closed-form analytical solution for such nonlinear ordinary differential equations, generating the Jacobian matrix (6.1) requires an intensive numerical calculation. Employing a numerical differentiation method [51, Chapter 6], one can calculate each individual term of the Jacobian matrix using the centered formula of order $O(h^4)$

$$f'(x) \approx \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h} \tag{6.2}$$

where h indicates step size for the numerical differentiation. Selection of h is important because selecting it too high or too small results in wrong estimation [51, Chapter 6]. Choosing it to be too small causes the result of the differentiation to be zero or almost zero since the functions in the numerator of (6.2) lose their significance with respect to each other, i.e., they all results in the same number. On the other hand, selecting h to be too high prevents the two points from being close enough to each other to extract information from that particular location. In order to overcome this selection issue, starting from a relatively high step size selection, one can reduce the step size until the result of differentiation starts to converge to zero. As a result, in this study, h is selected to be $10^{-2-(N/10)}$ and the differentiation is repeated with increasing N until the condition

$$|D_{N+1} - D_N| \geq |D_N - D_{N-1}| \quad \text{for} \quad N = 1, 2, \dots \tag{6.3}$$

holds. The condition ensures that the step size is reduced until the differentiation starts to converge to zero. The numerical differentiation has to be repeated for each member of the Jacobian matrix, which contains $2n \times 2n$ individual members. Considering (6.2) and (6.3), for a 10×10 Jacobian matrix, the closed-loop dynamics have to be solved at least 1200 times throughout a full walking step.

Since the gait stability can be determined by analyzing the eigenvalues of the system's single-step return map, which is linearized about the fixed point, this tool can be used to comment on the performances of the controllers introduced in previous chapters. The maximum magnitude eigenvalues of the undisturbed closed-loop systems with zero dynamics control, adaptive model predictive control, and adaptive power optimal control are given in the first column of Table 6.1, respectively. Since no disturbance and uncertainty is introduced, similar maximum magnitude eigenvalues are obtained. Input limits are already taken into account during the trajectory generation optimiza-

tion, and in the undisturbed case, everything is within the constrained limits, and no input saturation occurs.

In order to check the robustness of the controllers against modeling errors, the same calculation is repeated with some modeling error introduced in the system dynamics. There always are some unavoidable deviations between the dynamical model and the actual system in real-life applications. In order to mimic these differences, an aggressive modeling error is introduced, and the mass of the torso is multiplied by two in the simulator. The disturbance is introduced in the torso because it is the heaviest, the longest, and the unactuated joint. The maximum magnitude eigenvalues of the disturbed closed-loop systems with zero dynamics control, adaptive model predictive control, and adaptive power optimal control are given in the second column of Table 6.1, respectively. Since the maximum magnitude eigenvalue of the closed-loop system with zero dynamics control increased dramatically. This controller mainly relies on direct torque computation from the system dynamics, and any error introduced in the system model reduces the stability of the closed-loop system. Since the eigenvalue of the one-step response of the walker is too close to unity, the closed-loop system is simulated for multiple steps. It is observed that the controller cannot maintain stability, and the walker falls forward after two walking steps. The result is the same even after the input saturation is disabled. On the other hand, since the model predictive control is based on input selection that minimizes the cost function (4.8) and (5.1), it handles modeling errors better, and among the others, it results in the smallest maximum magnitude eigenvalue. Input selection in the model predictive control method does not directly rely on the system dynamics, as is the case in the computed torque control. The maximum magnitude eigenvalue of the closed-loop system with power optimal control is higher than the one with model predictive control. Since the mass of the torso is multiplied by two, the walker tends to fall forward. In order not to fall forward, the velocity of the swing leg should be increased such that the impact occurs before the walker falls. In the power optimal control, the controller does not apply high torques at high velocities since the input power is minimized. This behavior decreases the high-speed disturbance rejection performance of the system, as also shown in the basin of attraction calculation (see Fig. 6.1).

Table 6.1: Maximum magnitude eigenvalues of the disturbed and undisturbed closed-loop systems.

	Undisturbed	Doubled Torso Mass
Zero Dynamics Control	0.75	0.94
Adaptive MPC	0.71	0.66
Power Optimal Control	0.78	0.84

CHAPTER 7

DISCUSSION

This study discusses dynamic modeling and control of five-link underactuated planar bipedal walking. It is shown that using the hybrid zero dynamics notion, considering the desired characteristics, a walking trajectory can be generated for bipedal walkers. Additionally, it is also shown that using the zero dynamics notion, a feedback linearization input with PD control can be developed such that the generated desired trajectory is followed. One drawback of this method is that since the trajectory generation optimization problem contains too many nonlinear equality and inequality constraints and since the dimension of the system is too high, the trajectory generation optimization problem has to be solved offline. Then the generated trajectory is followed using a zero dynamics control, i.e., a computed torque with PD control. This input is an unconstrained computed torque, which means when there is a disturbance or uncertainty, nothing prevents input from exceeding its limits. Then it is shown that an adaptive model predictive control method can be implemented such that online trajectory tracking can be done with an input limited optimal control. Additionally, it is discussed that, since power input is also a consideration, the cost function of the controller can be manipulated such that it minimizes power usage while limiting the input torque of the system.

An important aspect of HZD optimization for trajectory generation is that the solution does not guarantee existence and optimality. Since the cost function is highly nonlinear, the method allows finding a solution if there exists that satisfies constraints. And offline calculation necessity prevents the walker from adapting itself for different walking surfaces such as inclination, stair climbing, roughness, etc. As mentioned, in order to adjust the walker for different walking surfaces, literature that implements

HZD employs a trajectory library such that with a proper switching policy, walking gait can be switched depending on surface conditions. This approach also requires foresight such that the designer should predict all types of conditions, generate proper walking gaits corresponding to those conditions and store them in the library.

The performances of the controllers are compared within their corresponding chapters and in chapter 6. It is shown that the adaptive model predictive control is superior to the zero dynamics control in many aspects. First of all, it accounts for input saturation and regulates the walker accordingly. Since the model predictive control does not calculate torque directly relying on the system dynamics, as in the zero dynamics control, the control method allows a certain amount of posture changes around the predefined trajectory. This posture change enables the walker to be more robust against disturbances than the input saturated zero dynamics control.

Even though some performance comparisons are made between different controllers, it should be noted that parameter selections are not absolute. Eigenvalues and eigenvectors of numerically calculated Poincaré return map estimations are employed for parameter tuning. A critical drawback of numerical computations is that they are not insightful. It is very difficult, if not impossible, to establish a cause-and-effect relationship between a particular parameter and the resultant numerical computation, and the information is local. In this case, it is even more challenging because the system's dynamics are highly nonlinear, and the dimension of the model is too high. Considering these, even though some well-educated guesses are made, and some proper mathematical notions are used, it is clear that the selected parameters do not guarantee the best performance. Additionally, since the stability information is local, it is hard to estimate the system's behavior under disturbances and uncertainties. The system's response to various effects is observed throughout the study to compensate for this issue. In the different regions, the controllers may show different performances.

Even though it is shown that the proposed adaptive model predictive control is superior to the zero dynamics control in many aspects, it has its own drawbacks. It is discussed that parameter selection for the proposed adaptive model predictive control is not trivial, as it was almost trivial for zero dynamics control. Additionally, it requires much more parameter selection than the zero dynamics control. Further, the

implementation of zero dynamics control is more straightforward since the proposed adaptive model predictive control requires much more preparation to obtain the corresponding cost function. After the cost function generation, an optimization problem solver takes place and performs additional computations to find cost-minimizing inputs.

7.1 Future Studies

Unlike the approach followed in this study, the horizon of the model predictive control usually contains two variables called the control horizon and prediction horizon. The control horizon indicates the number of time steps that are to be controlled with changing input. The prediction horizon indicates the number of further time steps that are to be observed with constant input. An increase in the control horizon increases the required amount of computation to calculate the system's input. On the other hand, since there is no new input calculation during the prediction horizon, it is used to obtain a more informative cost function. In order to increase the performance of the controller, a good balance should be found between the computational load and good system estimation. A more throughout explanation along with parameter selection discussions is captured in [44, Chapter 20]. Even though the system model of the walker introduced in this study is highly nonlinear and estimation and parameter selection are not that trivial, separating the control and prediction horizon in the controllers introduced in this study could increase the walker's performance. Observing the performance of the closed-loop systems with separate control and prediction horizon is a future study.

One of the main differences between legged robots and other mechatronics machines is that legged robots are not fixed to a place and interact with changing environments, for example, different surface types. Additionally, since long open kinematic chains constitute them and they contain many joints, the dynamics of bipedal walkers are too complicated. These issues motivate the importance of experimental realization of the proposed control methods for such robotic walkers. Implementing the proposed control method on an experimental testbed is an open problem in future studies. If it is not possible, implementing these methods on a well-accepted physical simulator,

such as $gazebo$, is also a valuable prospective study.

In this study, during the model predictive control implementation, impact dynamics are not taken into account. Even though the horizon is too short and only the first input computation is used at each time step, detecting impact and switching system dynamics during cost function generation would enhance the system's closed-loop performance. It should be noted that due to the system's nonlinearity, impact detection while constructing cost function is not that trivial.

REFERENCES

[1] C. G. Atkeson, P. Benzun, N. Banerjee, D. Berenson, C. P. Bove, X. Cui, M. DeDonato, R. Du, S. Feng, P. Franklin, *et al.*, "What happened at the darpa robotics challenge finals," in *The DARPA robotics challenge finals: Humanoid robots to the rescue*, pp. 667–684, Springer, 2018.

[2] E. Krotkov, D. Hackett, L. Jackel, M. Perschbacher, J. Pippine, J. Strauss, G. Pratt, and C. Orlowski, "The darpa robotics challenge finals: Results and perspectives," *Journal of Field Robotics*, vol. 34, no. 2, pp. 229–240, 2017.

[3] M. Johnson, B. Shrewsbury, S. Bertrand, T. Wu, D. Duran, M. Floyd, P. Abeles, D. Stephen, N. Mertins, A. Lesman, *et al.*, "Team ihmc's lessons learned from the darpa robotics challenge trials," *Journal of Field Robotics*, vol. 32, no. 2, pp. 192–208, 2015.

[4] C. G. Atkeson, B. P. W. Babu, N. Banerjee, D. Berenson, C. P. Bove, X. Cui, M. DeDonato, R. Du, S. Feng, P. Franklin, M. Gennert, J. P. Graff, P. He, A. Jaeger, J. Kim, K. Knoedler, L. Li, C. Liu, X. Long, T. Padir, F. Polido, G. G. Tighe, and X. Xinjilefu, "No falls, no resets: Reliable humanoid behavior in the darpa robotics challenge," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 623–630, 2015.

[5] J. Lim, I. Lee, I. Shim, H. Jung, H. M. Joe, H. Bae, O. Sim, J. Oh, T. Jung, S. Shin, *et al.*, "Robot system of drc-hubo+ and control strategy of team kaist in darpa robotics challenge finals," *Journal of Field Robotics*, vol. 34, no. 4, pp. 802–829, 2017.

[6] J. Kerdraon, J. G. Previnaire, M. Tucker, P. Coignard, W. Allegre, E. Knappen, and A. Ames, "Evaluation of safety and performance of the self balancing walking system atalante in patients with complete motor spinal cord injury," *Spinal cord series and cases*, vol. 7, no. 1, pp. 1–8, 2021.

[7] M. Tucker, E. Novoseller, C. Kann, Y. Sui, Y. Yue, J. W. Burdick, and A. D. Ames, "Preference-based learning for exoskeleton gait optimization," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2351–2357, IEEE, 2020.

[8] T. Elery, S. Rezazadeh, E. Reznick, L. Gray, and R. D. Gregg, "Effects of a powered knee-ankle prosthesis on amputee hip compensations: A case series," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 12, pp. 2944–2954, 2020.

[9] T. Elery, S. Rezazadeh, C. Nesler, and R. D. Gregg, "Design and validation of a powered knee–ankle prosthesis with high-torque, low-impedance actuators," *IEEE Transactions on Robotics*, vol. 36, no. 6, pp. 1649–1668, 2020.

[10] R. Gehlhar, J.-h. Yang, and A. D. Ames, "Powered prosthesis locomotion on varying terrains: Model-dependent control with real-time force sensing," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5151–5158, 2022.

[11] K. Li, M. Tucker, R. Gehlhar, Y. Yue, and A. D. Ames, "Natural multicontact walking for robotic assistive devices via musculoskeletal models and hybrid zero dynamics," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4283–4290, 2022.

[12] E. Westervelt, J. Grizzle, and D. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 42–56, 2003.

[13] A. D. Ames, "Human-inspired control of bipedal walking robots," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1115–1130, 2014.

[14] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.

[15] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. Westervelt, C. C. De Wit, and J. Grizzle, "Rabbit: A testbed for advanced control theory," *IEEE Control Systems Magazine*, vol. 23, no. 5, pp. 57–79, 2003.

[16] E. R. Westervelt, G. Buche, and J. W. Grizzle, "Experimental validation of a framework for the design of controllers that induce stable walking in planar bipeds," *The International Journal of Robotics Research*, vol. 23, no. 6, pp. 559–582, 2004.

[17] I. Poulakakis, *Stabilizing monopedal robot running: Reduction-by-feedback and compliant hybrid zero dynamics*. University of Michigan, 2009.

[18] B. Morris, E. Westervelt, C. Chevallereau, G. Buche, and J. Grizzle, "Achieving bipedal running with rabbit: Six steps toward infinity," in *Fast Motions in Biomechanics and Robotics*, pp. 277–297, Springer, 2006.

[19] K. Sreenath, *Feedback control of a bipedal walker and runner with compliance*. PhD thesis, University of Michigan, 2011.

[20] K. Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle, "A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on mabel," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1170–1193, 2011.

[21] K. Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle, "Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on mabel," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 324–345, 2013.

[22] J. W. Grizzle, C. Chevallereau, A. D. Ames, and R. W. Sinnet, "3d bipedal robotic walking: models, feedback control, and open problems," *IFAC Proceedings Volumes*, vol. 43, no. 14, pp. 505–532, 2010.

[23] J. W. Grizzle, C. Chevallereau, R. W. Sinnet, and A. D. Ames, "Models, feedback control, and open problems of 3d bipedal robotic walking," *Automatica*, vol. 50, no. 8, pp. 1955–1988, 2014.

[24] A. D. Ames, E. A. Cousineau, and M. J. Powell, "Dynamically stable bipedal robotic walking with nao via human-inspired hybrid zero dynamics," in *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, pp. 135–144, 2012.

[25] M. J. Powell, A. Hereid, and A. D. Ames, "Speed regulation in 3d robotic walking through motion transitions between human-inspired partial hybrid zero dynamics," in *2013 IEEE international conference on robotics and automation*, pp. 4803–4810, IEEE, 2013.

[26] A. Hereid, E. A. Cousineau, C. M. Hubicki, and A. D. Ames, "3d dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1447–1454, IEEE, 2016.

[27] Q. Nguyen, A. Agrawal, W. Martin, H. Geyer, and K. Sreenath, "Dynamic bipedal locomotion over stochastic discrete terrain," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1537–1553, 2018.

[28] R. Hartley, X. Da, and J. W. Grizzle, "Stabilization of 3d underactuated biped robots: Using posture adjustment and gait libraries to reject velocity disturbances," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 1262–1269, 2017.

[29] M. Fevre, H. Lin, and J. P. Schmiedeler, "Stability and gait switching of underactuated biped walkers," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2279–2285, 2019.

[30] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement learning for robust parameterized locomotion control of bipedal robots," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2811–2817, 2021.

[31] M. J. Powell, E. A. Cousineau, and A. D. Ames, "Model predictive control of underactuated bipedal robotic walking," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5121–5126, 2015.

[32] E. Cousineau and A. D. Ames, "Realizing underactuated bipedal walking with torque controllers via the ideal model resolved motion method," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5747–5753, 2015.

[33] X. Xiong, J. Reher, and A. D. Ames, "Global position control on underactuated bipedal robots: Step-to-step dynamics approximation for step planning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2825–2831, 2021.

[34] E. Daneshmand, M. Khadiv, F. Grimminger, and L. Righetti, "Variable horizon mpc with swing foot dynamics for bipedal walking control," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2349–2356, 2021.

[35] G. Gibson, O. Dosunmu-Ogunbi, Y. Gong, and J. Grizzle, "Terrain-aware foot placement for bipedal locomotion combining model predictive control, virtual constraints, and the alip," *arXiv preprint arXiv:2109.14862*, 2021.

[36] M. Krause, J. Englsberger, P.-B. Wieber, and C. Ott, "Stabilization of the capture point dynamics for bipedal walking based on model predictive control," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 165–171, 2012. 10th IFAC Symposium on Robot Control.

[37] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 2017.

[38] M. W. Spong, S. Hutchinson, M. Vidyasagar, *et al.*, *Robot modeling and control*, vol. 3. Wiley New York, 2006.

[39] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2018.

[40] Y. Hurmuzlu and D. B. Marghitu, "Rigid body collisions of planar kinematic chains with multiple contact points," *The International Journal of Robotics Research*, vol. 13, no. 1, pp. 82–92, 1994.

[41] A. Isidori, "Nonlinear control systems. communications and control engineering," *Springer. 3rd edition.*, 1995.

[42] P. Bézier, "Numerical control-mathematics and applications," *Translated by AR Forrest*, 1972.

[43] A. D. Ames and I. Poulakakis, "Hybrid zero dynamics control of legged robots," 2018.

[44] D. E. Seborg, T. F. Edgar, D. A. Mellichamp, and F. J. Doyle III, *Process dynamics and control*. John Wiley & Sons, 2016.

[45] P. Yedamale, "Brushless dc (bldc) motor fundamentals," *Microchip Technology Inc*, vol. 20, no. 1, pp. 3–15, 2003.

[46] M. M. Ankarali and U. Saranli, "Stride-to-stride energy regulation for robust self-stability of a torque-actuated dissipative spring-mass hopper," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 20, no. 3, p. 033121, 2010.

[47] J. Schmitt, "A Simple Stabilizing Control for Sagittal Plane Locomotion," *Journal of Computational and Nonlinear Dynamics*, vol. 1, pp. 348–357, 05 2006.

[48] S. G. Carver, N. J. Cowan, and J. M. Guckenheimer, "Lateral stability of the spring-mass hopper suggests a two-step control strategy for running," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 19, no. 2, p. 026106, 2009.

[49] H. Geyer, A. Seyfarth, and R. Blickhan, "Compliant leg behaviour explains basic dynamics of walking and running," *Proceedings of the Royal Society B: Biological Sciences*, vol. 273, no. 1603, pp. 2861–2867, 2006.

[50] S. H. Strogatz, *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. CRC press, 2018.

[51] J. H. Mathews, K. D. Fink, *et al.*, *Numerical methods using MATLAB*, vol. 4. Pearson prentice hall Upper Saddle River, NJ, 2004.