ROBUST AND ADAPTIVE DYNAMIC WALKING OF BIPEDAL ROBOTS

Submitted in partial fulfillment of the requirements for

the degree of

Doctor of Philosophy

in

Department of Mechanical Engineering

Quan T. Nguyen

B.S., Electrical Engineering, Hanoi University of Science and Technology

Carnegie Mellon University
Pittsburgh, PA

December, 2017

To my wife.

ACKNOWLEDGEMENTS

This thesis could not be completed without the help of many mentors and friends that I met during my research career at Carnegie Mellon University.

First of all, I would like to thank my advisor, Prof. Koushil Sreenath, for all his support and guidance. Without him, this dissertation would never be possible. I can't thank him enough for all his help.

Also, I would like to thank my committee members, Prof. Chris Atkeson, Prof. Mark Bedillion, Prof. Hartmut Geyer and Prof. Aaron Johnson for providing constructive feedback and suggestions for my research as well as in the writing of this thesis.

Furthermore, I would like to thank all the members in my lab: thanks to Guofan for all his help whenever I had any question about CMU and for all his support in research and life, thanks to Ayush for spending a significant time with me to do experiments no matter how many times we failed, thanks to Brian for great course projects, thanks to Katie, Avinash, Prasanth and Roberto for being my great lab-mates and friends.

I further want to thank William Martin for his support on experiment and hardware issues with the ATRIAS robot.

I also want to thank Chris Hertz, the MechE Manager of Academic Programs, who always offers prompt response and help with all the paperwork.

Most importantly, I would like to thank my family for their constant love and support. Especially I want to express my gratitude to my wife, Yen, who always believes in me and understands me. I am so blessed to have them in my life.

Finally, I want to acknowledge the financial support from Carnegie Mellon University and the National Science Foundation (NSF grant IIS-1526515) for my study and research.

ABSTRACT

Legged locomotion has several interesting challenges that need to be addressed, such as the ability of dynamically walk over rough terrain like stairs or stepping stones, as well as the ability to adapt to unexpected changes in the environment and the dynamic model of the robot. This thesis is driven towards solving these challenges and makes contributions on theoretical and experimental aspects to address: dynamic walking, model uncertainty, and rough terrain. On the theoretical front, we introduce and develop a unified robust and adaptive control framework that enables the ability to enforce stability and safety-critical constraints arising from robotic motion tasks under a high level of model uncertainty. We also present a novel method of walking gait optimization and gait library to address the challenge of dynamic robotic walking over stochastically generated stepping stones with significant variations in step length and step height, and where the robot has knowledge about the location of the next discrete foothold only one step ahead. On the experimental front, our proposed methods are successfully validated on ATRIAS, an underactuated, human-scale bipedal robot. In particular, experimental demonstrations illustrate our controller being able to dynamically walk at 0.6 m/s over terrain with step length variation of 23 to 78 cm, as well as simultaneous variation in step length and step height of 35 to 60cm and -22 to 22cm respectively. In addition to that, we also successfully implemented our proposed adaptive controller on the robot, which enables the ability to carry an unknown load up to 68 lb (31 kg) while maintaining very small tracking errors of about 0.01 deg (0.0017 rad) at all joints.

To be more specific, we firstly develop robust control Lyapunov function based quadratic program (CLF-QP) controller and L1 adaptive control to handle model uncertainty for bipedal robots. An application is dynamic walking while carrying an unknown load. The robust CLF-QP controller can guarantee robustness via a quadratic program that can be extended further to achieve robust safety-critical control. The L1 adaptive control can estimate and adapt to the presence of model uncertainty in the system dynamics.

We then present a novel methodology to achieve dynamic walking for underactuated and hybrid dynami-

cal bipedal robots subject to safety-critical constraints. The proposed controller is based on the combination of control Barrier functions (CBFs) and control Lyapunov functions (CLFs) implemented as a state-based online quadratic program to achieve stability under input and state constraints. The main contribution of this work is the control design to enable stable dynamical bipedal walking subject to strict safety constraints that arise due to walking over a terrain with randomly generated discrete footholds.

We next introduce Exponential Control Barrier Functions (ECBFs) as means to enforce high relative-degree safety constraints for nonlinear systems. We also develop a systematic design method that enables creating the Exponential CBFs for nonlinear systems making use of tools from linear control theory. Our method creates a smooth boundary for the safety set via an exponential function, therefore is called Exponential CBFs. Similar to exponential stability and linear control, the exponential boundary of our proposed method helps to have smoother control inputs and guarantee the robustness under model uncertainty. The proposed control design is numerically validated on a relative degree 4 nonlinear system (the two-link pendulum with elastic actuators and experimentally validated on a relative degree 6 linear system (the serial cart-spring system). Thanks to these advantages of Exponential CBFs, we then can apply the method to the problem of 3D dynamic walking with varied step length and step width as well as dynamic walking on time-varying stepping stones. For the work of using CBF for stepping stones, we use only one nominal walking gait. Therefore the range of step length variation is limited ($[25 : 60](cm)$). In order to improve the performance, we incorporate CBF with gait library and increase the step length range significantly ($[10 : 100](cm)$).

While handling physical constraints and step transition via CBFs appears to work well, these constraints often become active at step switching. In order to resolve this issue, we introduce the approach of 2-step periodic walking. This method not only gives better step transitions but also offers a solution for the problem of changing both step length and step height. Experimental validation on the real robot was also successful for the problem of dynamic walking on stepping stones with step lengths varied within $[23 : 78](cm)$ and average walking speed of $0.6(m/s)$.

In order to address the problems of robust control and safety-critical control in a unified control framework, we present a novel method of optimal robust control through a quadratic program that offers tracking stability while subject to input and state-based constraints as well as safety-critical constraints for nonlin-

ear dynamical robotic systems under significant model uncertainty. The proposed method formulates robust control Lyapunov and barrier functions to provide guarantees of stability and safety in the presence of model uncertainty. We evaluate our proposed control design on different applications ranging from a single-link pendulum to dynamic walking of bipedal robot subject to contact force constraints as well as safety-critical precise foot placements on stepping stones, all while subject to significant model uncertainty. We conduct preliminary experimental validation of the proposed controller on a rectilinear spring-cart system under different types of model uncertainty and perturbations. To solve this problem, we also present another solution of adaptive CBF-CLF controller, that enables the ability to adapt to the effect of model uncertainty to maintain both stability and safety. In comparison with the robust CBF-CLF controller, this method not only can handle a higher level of model uncertainty but is also less aggressive if there is no model uncertainty presented in the system.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

There is tremendous interest in employing legged and humanoid robots for dangerous missions in disaster and rescue scenarios. This is evidenced by the grand challenge in robotics, The DARPA Robotics Challenge (DRC). Such time and safety critical missions require the robot to operate swiftly and stably while dealing with high levels of uncertainty and large external disturbances. In addition to inaccurate robot models, model uncertainty can also arise from interaction tasks. For instance, practical robotic applications involving walking robots that lift and carry large unknown loads, pull unknown weights, drag heavy hoses, etc., will have to deal with significant changes to the dynamical model. The limitation of current research, as well as the demand of practical requirement, motivates our research on robust control for hybrid systems in general and bipedal robots in particular.

Furthermore, for a rescue mission, the ability to navigate rough terrains like climbing stairs and stepping stones are very crucial (see Fig. 1.1). This also motivates us to develop a control framework to handle the

(a) Exploration Mission (b) Mars sufface (c) Fire Fighting (d) Disaster Rescue

Figure 1.1: Exploration missions, fire fighting and disaster rescue are motivation examples for developing legged robotics systems with the ability of adapting to uneven terrains and unexpected uncertainty.

problem of dynamic walking on stepping stones in this thesis. While most of current works on footstep placement rely on quasi-static walking, resulting in slow walking speed, short step length and high energy consumption, our proposed work attempts to offer a solution for dynamic walking on stepping stones that instead has fast walking gait, wider range of step length variation and is very energy efficient. Since any disaster scenario may require a long time of search and rescue, energy efficiency plays a very important role to guarantee long enough operation time of the robot under the limitation of its own battery capacity.

For the problem of dynamic walking of bipedal robots, in recent years, the method of Hybrid Zero Dynamics (HZD), [79], has been very successful demonstrated in different robots [15], [72]. However, a primary shortcoming of the method is that it assume perfect knowledge of the dynamic model and deal with periodic walking only. The goal of this thesis is to relax the requirement of perfect knowledge of the model by allowing for bounded uncertainty, and designing an optimal robust controller through a CLF-QP so as to still retain stability. In addition to that, our methods can also offer a solution to address aperiodic motion either via real-time feedback controller or gait optimization. Our development therefore can handle aperiodic walking under model uncertainty. An application is dynamic walking on stepping stones while carrying an unknown load, that requires the robot to be able to adapt to changes in the robot dynamics as well as ground terrain characteristics.

1.2 Thesis Contributions

The main contributions of this thesis with respect to prior work are as follows:

- Robust control via control Lyapunov functions and L1 adaptive control to guarantee robust stability under model uncertainty. We apply these two controllers to the problem of dynamic walking while carrying an unknown load.
- Introducing Exponential control Barrier functions, a novel concept to address high-relative degree constraints in dynamical systems. The proposed control design is numerically validated on a relative degree 4 nonlinear system (the two-link pendulum with elastic actuators) and experimentally validated on a relative degree 6 linear system (the serial cart-spring system)
- Dynamic walking on stepping stones using control Barrier functions based controller. This work is for planar walking with changing step length. We guarantee footstep placement as a safety constraints enforced in real-time via feedback controller, allowing us to address the problem of dynamic walking on time-varying stepping stones as well.
- 3D dynamic walking on stepping stones using control Barrier functions with simultaneous variation

in step length and step width. Numerical validation is conducted on the model of DURUS, a 23 degree-of-freedom humanoid robot.

- Dynamic walking on stepping stones with gait library and control Barrier functions. The combination of CBF and gait library in this work enhances the range of step length variation as well as the feasibility of the quadratic programs based controller.

- Dynamic walking on stepping stones with 2-step periodic gait optimization. This method takes into account step transitions while using a small number of gaits. This work enables the ability to address the change in both step length and step height while having smoother step transitions. We successfully conducted experimental validation on ATRIAS robot walking on stepping stones with the step length variation of $[23 : 78](cm)$ as well as simultaneous variation in step length and step height of $[35 : 60](cm)$ and $[-22 : 22](cm)$ respectively while having dynamic walking with average speed of $0.6(m/s)$.

- Presenting a unified robust optimal controller that can guarantee at the same time robust stability, robust safety and robust constraints. The application on dynamic walking on stepping stones while carrying unknown load was validated. Preliminary experimental result is successfully implemented on a rectilinear spring-cart system under different types of model uncertainty and perturbations.

1.3 Thesis Organization

The rest of the thesis is organized as follows: in Chap. 2, we provide literature survey on prior work related to control of bipedal robots, footstep placement, robust and adaptive control; Chap. 3 revises background on Hybrid Zero Dynamics and control Lyapunov functions based quadratic programs controller; in Chap. 4, we analyze the effect of model uncertainty on the nominal CLF-QP controller and then introduce the methods of robust CLF-QP and L1 adaptive controller for CLF-QP controller. We apply these controllers to the problem of dynamic walking while carrying unknown load; Chap. 6 presents different results in our method of using control Barrier functions for the application of dynamic walking on stepping stones, time-varying stepping stones, 3D stepping stones with simultaneous variation in both step length and step width. In the last section of this Chapter, we incorporate CBF and gait library to enlarge the range of step length variation; we then introduce a novel unified robust optimal control based on quadratic programs in Chap. 8. This proposed method can help us to handle robust stability, robust safety, robust constraints at the same time. Numerical validation on the problem of bipedal walking on stepping stones while carrying unknown load and experimental validation on a spring-cart system were conducted using this method; in Chap. 7 we achieve better step transitions for the problem of stepping stones using 2-step periodic gait library. This

method also offers the ability to handle the problem of changing both step length and step height at the same time. We validated the method experimentally on ATRIAS, an underatuated bipedal robot. Lastly, Chapter. 10 presents conclusion and future work.

Chapter 2

Literature Survey

In this chapter, we will review related state-of-the-art research in control of dynamic walking of legged robots, footstep placement for bipedal walking, nonlinear control using Lyapunov theory and robust and adaptive control.

2.1 Walking Control of Bipedal Robots

For humanoid robot walking, the zero-moment point (ZMP) method [76], [36] was long established and is widely used even today. In this method, the trajectory of a walking robot is planned using the angular momentum equation to ensure the dynamical postural stability of the robot, which usually is quantified by the distance of the zero moment point from the boundary of a predefined stability region. However, this method results in quasi-static walking and is not energy efficient as well as not applicable for dynamic walking. The ZMP constraint also assumes flat ground contact, therefore is difficult to apply to walking on uneven terrains. For dynamic walking, people introduce the notion of passive dynamic walking [48], where a robot without on-board power uses passive gravitational energy from descending a shallow slope to settle into an energy-efficient walking gait. Walking control for actuated robots based on passive dynamic walking is also deployed in [17], [73]. Energy efficiency is often quantified using the notion of dimensionless "specific cost of transport" [74], which is the amount of energy consumed to carry a unit weight a unit distance. The Cornell Biped [18], a passive dynamic walker, has the same specific cost of transport as humans, equal to 0.20; while the humanoid Honda's ASIMO using ZMP method has a specific cost of transport of 3.23 [17]. Although passive dynamic based methods offer very energy efficient walking, the resulting gaits are not robust to perturbations and the stability is fragile. For handling uneven terrain, Raibert's method [33] uses step-by-step feedback control to determine the footstep placement in order to stabilize the robot. The methods gained some very impressive results [63] for rough terrain walking. However, because the control

design focuses on keeping the robot's balance and doesn't take into account the energy consumption of the motion, it therefore still has very hight specific cost of transport. Recently, bipedal walking template using Bipedal Spring-Mass model (or Dual-SLIP model) was presented in [26], that can reproduce the key characteristics in human walking, including the center of mass (CoM) vertical oscillations and human-like ground reaction force pattern. Since the method tends to mimic human-like walking in bipedal robots, it could potentially lead to better energy efficient walking. Interestingly, bipedal robot walking and running based on spring-mass models has been shown to be robust to uneven terrain [29], [77],[66], [16]. Instead of using the simplified model, the method of Hybrid Zero Dynamics(HZD) [80] is based on the full nonlinear dynamics of the robot and has a provable closed-loop stability. In this method, the walking gait optimization process tries to minimize the energy consumption with respect to the system dynamics and physical constraints, resulting in energy efficiency for the walking gait. Based on HZD, the work on human-inspired control [2] using human walking data for the optimization to get a more human-like walking gait. Because the HZD method is designed systematically based on the robot dynamics, it can be extended to a high degree-of-freedom humanoid robot [30]. However, primary shortcomings of the method are that it assumes perfect knowledge of the dynamic model and addresses periodic walking only. This thesis will try to resolve these shortcomings by addressing the problem of model uncertainty as well as aperiodic walking.

2.2 Footstep Placement for Bipedal Walking

Footstep placement control for fully actuated legged robots initially relied on quasi-static walking and resulted in slow walking speeds [36],[41],[14]. Impressive results in footstep planning and placements in obstacle filled environments with vision-based sensing have been carried out in [50],[13]. The DARPA Robotics Challenge inspired several new methods, some based on mixed-integer quadratic programs [20]. However, as mentioned in [21, Chap. 4], mixed-integer-based footstep planning does not offer dynamic feasibility even on a simplified model. These methods are therefore not applicable for dynamic walking with faster walking gaits. On the other hand, the approach developed in [85] allows aperiodic gaits with varying step lengths designed on a complete dynamical model, but requires the *a priori* design of controllers that realize precise transitions between each pair of elements of the gait library, resulting in exponential (factorial) growth in the number of pre-designed controllers.

2.3 Control Lyapunov Functions based Controller

Lyapunov functions and control Lyapunov functions (CLFs) are a classical tool for design and analysis of feedback control that stabilize the closed-loop dynamics of both linear and nonlinear dynamical systems,

see [24]. Traditional CLF-based controllers involve closed-form control expressions such as the min-norm and the Sontag controllers [68]. Recently, a novel approach of expressing CLF-based controllers via online quadratic programs (QPs) in [25] opened an effective way for dealing with stability while also enabling the incorporation of additional constraints, such as input constraints. In [25], the CLF-based controller is expressed in the QP as an inequality constraint on the time-derivative of the CLF, which easily enables adding additional constraints such as input saturation through the relaxation of the CLF inequality condition. The QP can be executed online at 1kHz in real-time as a state-dependent feedback controller. However, this work did not handle model uncertainty or safety-critical constraints.

In addition to CLFs, we also draw inspiration from recent methods of control barrier functions (CBFs) that can be incorporated with control Lyapunov function based quadratic programs to result in the CBF-CLF-QP, as introduced in [3]. This framework enables handling safety-critical constraints effectively in real-time. Experimental validation of this type of controller for the problem of Adaptive Cruise Control was presented in [49]. This framework has also been extended to various interesting application domains, such as safety-critical geometric control for quadrotor systems [82] and safety-critical dynamic walking for bipedal robots [58], [34]. Although this work can handle safety-critical constraints, however a precise model of the system is required to enforce the constraints.

Moreover, as presented in [84], preliminary robustness analysis of the CBFs indicate that the safety-critical constraint will be violated in the presence of model uncertainty, with the amount of violation being bounded by the value of the upper bound of the model uncertainty. In particular, model uncertainty leads to constraint violation of the safety-critical constraints. In this thesis, we seek a method to simultaneously handle robust stability, robust input-based constraints and robust state-dependent constraint in the presence of significant model uncertainty. We will do this through robust control formulations of the CLF, CBF, and constraints.

2.4 Robust and Adaptive Control

Robust control is an extensively studied topic. We have established methods, such as H_∞-based robust control and linear quadratic Guassian (LQG) based robust control [43, 87] for robust control of linear systems. For robust control of nonlinear systems, input-to-state stability (ISS) and sliding mode control (SMC) are two main methods. The ISS technique (see [69, 70, 5]) can be used to both analyze the robustness of nonlinear systems as well as design robust controllers based on control Lyapunov functions. However, a primary disadvantage of ISS based controllers is that the resulting controller only maintains the system errors in a sufficiently small neighborhood of the origin, resulting in non-zero tracking errors. In recent years, there has been work on robust control of hybrid systems based on the ISS technique, for instance see [75, 32, 8].

In contrast, sliding mode control techniques can deal with a wide range of uncertainties and drive the system to follow a given sliding surface, thereby driving outputs to desired values without any tracking errors (see [23, 22, 86]). However, the primary disadvantage of SMC is the chattering phenomenon caused by discrete switching for keeping the system attracted to the sliding surface.

Robust control techniques have also been extensively applied to robotic manipulator arms, see [71, 45], however manipulator arms do not have challenges such as underactuation and unilateral ground contact forces making their control easier. Robust control techniques have also been applied to bipedal walking. For instance, the work in [44] extends adaptive robust control of manipulator arms to bipedal robots, the work in [38] considers a simple 2D inverted-pendulum model and pre-computes a control policy through offline nonlinear optimization to prevent falls under the assumption of bounded disturbances. Similarly, recent results on robust feedback motion planning for the problem of UAVs avoiding obstacles in [46] also precomputed a library of "funnels" via convex optimization that represents different maneuvers of the system under bounded disturbances. A real-time planner then composes motion trajectories based on the resulting funnel library. Offline optimization for stabilization of walking and running with robustness to discrete-time uncertainties such as terrain perturbations has been carried out in [27, 28]. Our method, in contrast, is based on real-time feedback controller to guarantee robust stability as well as robust safety of robotic systems through an online optimization without the need of precomputed motion plans. Additional robust planning and control techniques exist for legged robots where the robustness is with respect to stochastic uncertainty in the model / terrain, for instance see [37, 65, 6, 7].

In addition to that, in recent years, the introduction of L_1 adaptive control technique has enabled decoupling of adaptation and robustness in adaptive control techniques. In particular, by applying a low-pass filter as part of the adaptation laws help the L_1 adaptive controller to guarantee not only stability [9] but also transient performance [10]. L_1 adaptive control appears to have great potential for application in aerospace systems, illustrated in [11], [62]. However, to the best of our knowledge, using L_1 adaptive control to deal with uncertainty for control of bipedal robots, systems that are hybrid, high-dimensional, nonlinear and underactuated, has not been considered. Furthermore, while standard L_1 adaptive control typically solves the problem of tracking a given linear reference system, in this thesis, we present an adaptive control for nonlinear uncertainty with a nonlinear reference model that arises as the closed-loop system on application of a rapidly exponential stabilizing control Lyapunov function (RES-CLF) [4]. For control of bipedal robots, guaranteeing a suitable rate of convergence is very important for the stability of its hybrid dynamics. Thus we need to drive the robot to follow a fast reference dynamical model. The presence of a low-pass filter in the L_1 adaptive control allows us to prevent high-frequency control signals that are typical and frequently seen in adaptive control problems. This will be critical to keep motor torques less noisy, and will contribute to ensuring the validity of the unilateral ground contact constraints, as well as retaining the energy efficiency of walking control of bipedal robot.

Having reviewed related works, in the next Chapter, we will revise the control Lyapunov function based controller that will be used throughout the thesis.

Chapter 3

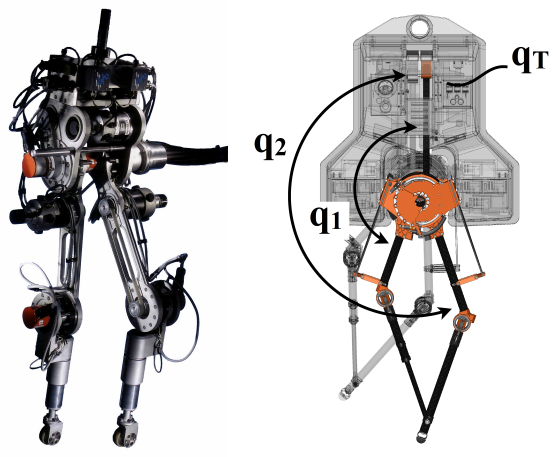
Hybrid Zero Dynamics and Control Lyapunov Functions based Quadratic Program for Dynamic Walking

In this chapter, we will present background on dynamic walking of bipedal robots using Hybrid Zero Dynamics and then control Lyapunov functions (CLF) based quadratic program controller that will be used widely throughout the thesis. This nonlinear controller will guarantee exponential convergence of the control outputs and the closed-loop hybrid system. The CLF condition can be expressed as a quadratic constraints therefore can be solved using quadratic program, opening up the ability to address additional constraints such as input saturation, friction constraints that cannot be done using PD control. Firstly, we will present input-output linearization to cancel out the nonlinearity in the system dynamics. We then can easily design our controller for the linearized system.

3.1 Dynamical Model for Walking

In this thesis, we will use two robot models (a) RABBIT [15] and (b) a planar representation of ATRIAS [64] that are shown in Fig. 3.1. We will next briefly describe the robot kinematics, construct the dynamical model, perform nonilinear optimization to search for a periodic gait, and design a controller to stabilize the gait on the ATRIAS robot model. Mote that a similar approach can be used for RABBIT or other bipedal robots.

ATRIAS's total mass is 63 kg, with approximately 50% of the mass in the hips, 40% in the torso, and with light legs formed by a four-bar linkage. The robot is approximately left-right symmetric. The



(a) RABBIT (b) ATRIAS

Figure 3.1: The two robot models are used in the thesis.

configuration variables for the system can be defined as $q := (q_T, q_{1R}, q_{2R}, q_{1L}, q_{2L}) \in \mathbb{R}^5$. The variable q_T corresponds to the world frame pitch angle of the torso, while the variables $(q_{1R}, q_{2R}, q_{1L}, q_{2L})$ refer to the local coordinates for the linkages. Each of the four linkages are actuated by a DC motor behind a 50:1 gear ratio harmonic drive, with the robot having one degree of underactuation. The four-bar linkage mechanism comprising of the leg coordinates (q_1, q_2) map to the leg angle and knee angle (q_{LA}, q_{KA}), as $q_{LA} := \frac{1}{2}(q_1 + q_2)$ and $q_{KA} := q_2 - q_1$. With the state $x = (q, \dot{q})$ denoting the generalized positions and velocities of the robot and u denoting the joint torques, a hybrid model of walking can be expressed as

$$\begin{cases} \dot{x} &= f(x) + g(x)\, u, & x^- \notin \mathcal{S} \\ x^+ &= \Delta(x^-), & x^- \in \mathcal{S}, \end{cases} \tag{3.1}$$

where $\mathcal{S}$ is the impact surface and Δ is the reset or impact map. A more complete description of the robot and a derivation of its model is given in [64].

Having described the dynamical model of ATRIAS, we will now present a model-based approach for designing a periodic walking gait that satisfy physical constraints arising from the robot and its environment.

3.2 Gait Design Using Virtual Constraints

The nominal feedback controller is based on the virtual constraints framework presented in [79]. Virtual constraints are kinematic relations that synchronize the evolution of the robot's coordinates via continuous-time feedback control. One virtual constraint in the form of a parametrized spline can be imposed for each

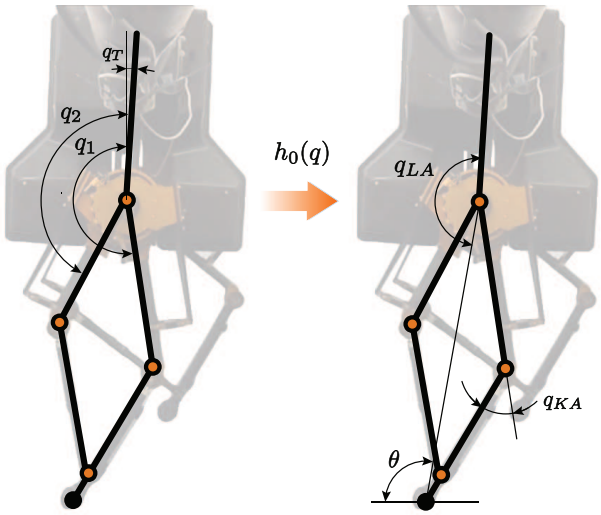


Figure 3.2: Biped coordinates and outputs. The world frame pitch angle is denoted by q_T, while (q_1, q_2) are body coordinates. The outputs to be controlled are denoted by q_{LA} and q_{KA}. The model is assumed left-right symmetric.

(independent) actuator. Parameter optimization is used to find the spline coefficients so as to create a periodic orbit satisfying a desired step length, while respecting physical constraints on torque, motor velocity, and friction cone. The optimization method used here is the direct collocation code from [35], although other methods, such as [31] or `fmincon` can be used as well.

The virtual constraints are expressed as an output vector

$$y = y_0(q) - y_d(s(q), \alpha), \tag{3.2}$$

to be asymptotically zeroed by a feedback controller. Here, $y_0(q)$ specifies the quantities to be controlled

$$h_0(q) = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} q_1^{st} \\ q_2^{st} \\ q_1^{sw} \\ q_2^{sw} \end{bmatrix} =: \begin{bmatrix} q_{LA}^{st} \\ q_{KA}^{st} \\ q_{LA}^{sw} \\ q_{KA}^{sw} \end{bmatrix}, \tag{3.3}$$

where st and sw designate the stance and swing legs, respectively, and $y_d(s, \alpha)$ is a 4-vector of Beziér polynomials in the parameters α specifying the desired evolution of $y_0(q)$, where s is a gait phasing variable

Table 3.1: Optimization constraints

Motor Torque	$\|u\| \leq 5$ Nm
Impact Impulse	$F_e \leq 15$ Ns
Friction Cone	$\mu_f \leq 0.4$
Vertical Ground Reaction Force	$F_{st}^v \geq 200$ N
Mid-step Swing Foot Clearance	$h_f\|_{s=0.5} \geq 0.1$ m
Dynamic Constraints	Eq. (3.1)

defined as

$$s := \frac{\theta - \theta_{init}}{\theta_{final} - \theta_{init}}, \tag{3.4}$$

with $\theta = q_T + q_{LA}^{st}$ being the absolute stance leg angle and $\theta_{init}, \theta_{final}$ being the values of θ at the beginning and end of the gait respectively.

The cost function and constraints for the optimization are formulated as in [79] [Chap. 6.6.2], with the optimization constraints given in Table 3.1 and the cost taken as integral of squared torques over step length:

$$J = \frac{1}{L_{step}} \int_0^T ||u(t)||_2^2 \, dt. \tag{3.5}$$

We have presented the representation of system dynamics for bipedal robots and the construction of control outputs using Hybrid Zero Dynamics. In the next Sections, we will present input-output linearization and the nonlinear control design using control Lyapunov functions to guarantee exponential stability for the closed-loop system.

3.3 Input-Output Linearizing Control

Consider a nonlinear control affine hybrid model

$$\begin{aligned} \mathcal{H} : &\begin{cases} \dot{x} = f(x) + g(x)u, & x \notin S \\ x^+ = \Delta(x^-), & x \in S \end{cases} \\ &y = y(x), \end{aligned} \tag{3.6}$$

where $x \in \mathbb{R}^n$ is the system state, $u \in \mathbb{R}^m$ is the control input, S is the switching surface of the hybrid system, and $y \in \mathbb{R}^m$ is a set of outputs. For the application on bipedal walking, the system dynamic is constructed in (3.1) and the control input $y(x)$ is designed in (3.2).

If the control output $y(x)$ has relative degree 2, then the time-derivative $\dot{y}(x)$ will be a function of the state x and not dependent on the control input u. Considering the second time-derivative $\ddot{y}$, we have:

$$\ddot{y} = \frac{\partial \dot{y}}{\partial x}\dot{x} = L_f^2 y(x) + L_g L_f y(x) u. \tag{3.7}$$

where L represents the Lie derivatives. To be more specific:

$$L_f^2 y(x) \triangleq \frac{\partial \dot{y}}{\partial x} f(x), \qquad L_g L_f y(x) \triangleq \frac{\partial \dot{y}}{\partial x} g(x). \tag{3.8}$$

If the decoupling matrix $L_g L_f y(x)$ is invertible, then the controller

$$u(x, \mu) = u_{ff}(x) + (L_g L_f y(x))^{-1} \mu, \tag{3.9}$$

with the feed-forward control input

$$u_{ff}(x) = -(L_g L_f y(x))^{-1} L_f^2 y(x), \tag{3.10}$$

input-output linearizes the system. The dynamics of the system (3.6) can then be described in terms of dynamics of the transverse variables, $\eta \in \mathbb{R}^{2m}$, and the coordinates $z \in \mathcal{Z}$ with $\mathcal{Z}$ being the co-dimension $2m$ manifold

$$\mathcal{Z} = \{x \in \mathbb{R}^n \mid \eta(x) \equiv 0\}. \tag{3.11}$$

One choice for the transverse variables is,

$$\eta = \begin{bmatrix} y(x) \\ \dot{y}(x) \end{bmatrix}. \tag{3.12}$$

The input-output linearized hybrid system then is,

$$\mathcal{H}^{IO}: \begin{cases} \dot{\eta} = \bar{f}(\eta) + \bar{g}(\eta)\mu, & \\ \dot{z} = p(\eta, z), & (\eta, z) \notin S, \\ \eta^+ = \Delta_X(\eta^-, z^-), & \\ z^+ = \Delta_Z(\eta^-, z^-), & (\eta^-, z^-) \in S. \end{cases} \tag{3.13}$$
$$y = y(\eta),$$

where z represents uncontrolled states [4], and

$$\bar{f}(\eta) = F\eta, \quad \bar{g}(\eta) = G, \tag{3.14}$$

with,

$$F = \begin{bmatrix} O & I \\ O & O \end{bmatrix} \text{ and } G = \begin{bmatrix} O \\ I \end{bmatrix}. \tag{3.15}$$

The linear system in (3.14) is in controllable canonical form, and with the PD control

$$\mu = \begin{bmatrix} -\frac{1}{\varepsilon^2}K_P & -\frac{1}{\varepsilon}K_D \end{bmatrix} \eta, \tag{3.16}$$

the closed-loop system will become

$$\ddot{y} + \frac{1}{\varepsilon}K_D\dot{y} + \frac{1}{\varepsilon^2}K_P y = 0, \tag{3.17}$$

and will be exponentially stable if we choose K_P and K_D such that

$$A = \begin{bmatrix} 0 & I \\ -K_P & -K_D \end{bmatrix} \tag{3.18}$$

is Hurwitz. The ε factor is used to control the rate of convergence and is needed to ensure we converge sufficiently fast so as to account for the expansion produced by the impact map, see [51] for more details. A corresponding quadratic Lyapunov function can then be established through the Lyapunov equation as done in the next subsection.

3.4 Control Lyapunov Function based Quadratic Programs

3.4.1 CLF-QP

Instead of a linear control design such as PD in (3.16), an alternative control design is through a control Lypapunov function $V(\eta)$, wherein a control is chosen pointwise in time such that the time deriviative of the Lyapunov function $\dot{V}(\eta, \mu) \leq 0$, resulting in stability in the sense of Lyapunov, or $\dot{V}(\eta, \mu) < 0$ for asymptotic stability, or $\dot{V}(\eta, \mu) + \lambda V(\eta) \leq 0, \lambda > 0$ for exponential stability.

To enable directly controlling the rate of convergence, we use a *rapidly exponentially stabilizing control Lyapunov function (RES-CLF)*, introduced in [4]. RES-CLFs provide guarantees of *rapid exponential stability* for the transverse variables η. In particular, a function $V_\varepsilon(\eta)$ is a RES-CLF for the system (3.6) if there exist positive constants $c_1, c_2, c_3 > 0$ such that for all $0 < \varepsilon < 1$ and all states (η, z) it holds that

$$c_1\|\eta\|^2 \leq V_\varepsilon(\eta) \leq \frac{c_2}{\varepsilon^2}\|\eta\|^2, \tag{3.19}$$

$$\dot{V}_\varepsilon(\eta, \mu) + \frac{c_3}{\varepsilon}V_\varepsilon(\eta) \leq 0. \tag{3.20}$$

The RES-CLF will take the form:

$$V_\varepsilon(\eta) = \eta^T \begin{bmatrix} \frac{1}{\varepsilon}I & 0 \\ 0 & I \end{bmatrix} P \begin{bmatrix} \frac{1}{\varepsilon}I & 0 \\ 0 & I \end{bmatrix} \eta =: \eta^T P_\varepsilon \eta, \tag{3.21}$$

where P is the solution of the Lyapunov equation $A^T P + PA = -Q$ (where A is given by (3.18) and Q is any symmetric positive-definite matrix); and the time derivative of the RES-CLF (3.21) is computed as

$$\dot{V}_\varepsilon(\eta, \mu) = \frac{\partial V_\varepsilon}{\partial \eta}\dot{\eta} = L_{\bar{f}}V_\varepsilon(\eta) + L_{\bar{g}}V_\varepsilon(\eta)\mu, \tag{3.22}$$

where

$$\begin{aligned} L_{\bar{f}}V_\varepsilon(\eta) &= \frac{\partial V_\varepsilon}{\partial \eta}\bar{f}(\eta) = \eta^T(F^T P_\varepsilon + P_\varepsilon F)\eta, \\ L_{\bar{g}}V_\varepsilon(\eta) &= \frac{\partial V_\varepsilon}{\partial \eta}\bar{g}(\eta) = 2\eta^T P_\varepsilon G. \end{aligned} \tag{3.23}$$

It can be show that for any Lipschitz continuous feedback control law μ that satisfies the RES condition (3.20), it holds that

$$V(\eta) \leq e^{-\frac{c_3}{\varepsilon}t}V(\eta(0)), \quad \|\eta(t)\| \leq \frac{1}{\varepsilon}\sqrt{\frac{c_2}{c_1}}e^{-\frac{c_3}{2\varepsilon}t}\|\eta(0)\|, \tag{3.24}$$

i.e. the rate of exponential convergence can be directly controlled with the constant ε through $\frac{c_3}{\varepsilon}$. One such controller is the pointwise min-norm control law [24], formulated as,

---

Pointwise min-norm:

$$\mu_\varepsilon(\eta, z) = \left\{ \begin{array}{ll} -\frac{\psi_{0,\varepsilon}(\eta,z)\psi_{1,\varepsilon}(\eta,z)}{\psi_{1,\varepsilon}(\eta,z)^T\psi_{1,\varepsilon}(\eta,z)} & \text{if} \quad \psi_{0,\varepsilon}(\eta, z) > 0 \\ 0 & \text{if} \quad \psi_{0,\varepsilon}(\eta, z) \leq 0 \end{array} \right\}. \tag{3.25}$$

---

where,

$$\begin{aligned} \psi_{0,\varepsilon}(\eta, z) &= L_{\bar{f}} V_\varepsilon(\eta, z) + \frac{c_3}{\varepsilon} V_\varepsilon(\eta, z) \\ \psi_{1,\varepsilon}(\eta, z) &= L_{\bar{g}} V_\varepsilon(\eta, z). \end{aligned} \tag{3.26}$$

The min-norm controller can be also expressed as a QP-based controller called (CLF-QP)-based controller, introduced in [25], where μ is directly selected through an online quadratic program to satisfy (3.20):

CLF-QP:

$$\begin{aligned} u^*(x) = \underset{u,\mu}{\text{argmin}} \quad & \mu^T\mu && (3.27) \\ \text{s.t.} \quad & \dot{V}_\varepsilon(\eta, \mu) + \frac{c_3}{\varepsilon} V_\varepsilon(\eta) \leq 0, && \textbf{(CLF)} \\ & u = u_{ff}(x) + (L_g L_f y(x))^{-1}\mu, && \textbf{(IO)} \end{aligned}$$

Note that the above minimization problem is a quadratic program since the ineqality constraint on the time-derivative of the Lyapunov function can be written as a linear ineqality constraint

$$A_{clf}\mu \leq b_{clf}, \tag{3.28}$$

where

$$A_{clf} = L_{\bar{g}} V_\varepsilon(\eta); b_{clf} = -L_{\bar{f}} V_\varepsilon(\eta) - \frac{c_3}{\varepsilon} V_\varepsilon(\eta). \tag{3.29}$$

Moreover, the IO equality constraint is linear in u, μ, and can be written as:

$$A_{IO} \begin{bmatrix} u \\ \mu \end{bmatrix} = b_{IO}, \tag{3.30}$$

where

$$A_{IO} = \begin{bmatrix} I & -(L_g L_f y(x))^{-1} \end{bmatrix}, \quad b_{IO} = u_{ff}(x). \tag{3.31}$$

This optimization is solved pointwise in time. Furthermore the existence of efficient quadratic program solvers, such as CVXGEN [47], enable solving the problem in real-time speeds over 1 kHz.

Remark 3.1. *Note that the solution of the above QP has been shown to be Lipschitz continuous [53].*

3.4.2 CLF-QP with Constraints

Formulating the control problem as a quadratic program now enables us to incorporate constraints into the optimization. These constraints could be input constraints for input saturation or state-based constraints such as friction constraints, contact force constraints, etc., for robotic locomotion and manipulation. These types of constraints can be expressed in a general form as

$$A_c(x)u \leq b_c(x). \tag{3.32}$$

The CLF-QP based controller with additional constraints then takes the form,

---

CLF-QP with Constraints:

$$\begin{aligned} u^*(x) = & \underset{u,\mu,\delta}{\operatorname{argmin}} && \mu^T \mu + p\delta^2 && (3.33) \\ & \text{s.t.} && \dot{V}_\varepsilon(\eta, \mu) + \frac{c_3}{\varepsilon} V_\varepsilon(\eta) \leq \delta, && \textbf{(CLF)} \\ & && A_c(x)u \leq b_c(x), && \textbf{(Constraints)} \\ & && u = u_{ff}(x) + (L_g L_f y(x))^{-1}\mu, && \textbf{(IO)} \end{aligned}$$

---

where p is a large positive number that represents the penalty of relaxing the inequality.

The constraints above could be input saturation constraints expressed as,

$$
\begin{aligned}
u^*(x) = \underset{u,\mu,\delta}{\operatorname{argmin}} \quad & \mu^T\mu + p\delta^2 && (3.34)\\
\text{s.t.} \quad & \dot{V}_\varepsilon(\eta,\mu) + \frac{c_3}{\varepsilon}V_\varepsilon(\eta) \le \delta, && \textbf{(CLF)}\\
& u_{min} \le u \le u_{max}, && \textbf{(Input Saturation)}\\
& u = u_{ff}(x) + (L_gL_fy(x))^{-1}\mu. && \textbf{(IO)}
\end{aligned}
$$

Note that, as illustrated in Appendix A.1, the CLF inequality condition in the above CLF-QPs is affine in μ, ensuring that these are actually quadratic programs.

This formulation opened a novel method to guarantee stability of nonlinear systems with respect to additional constraints such as torque saturation [25], wherein experimental demonstration of bipedal walking with strict input constraints was demonstrated, and L1 adaptive control for bipedal robots [56].

Remark 3.2. ***(Stability of CLF-QP with relaxed CLF inequality.)*** *In the subsequent sections, we develop different types of controllers based on the relaxed CLF-QP* (3.33)*. Allowing the violation of the RES-CLF condition* (3.20) *enables us to incorporate various other input and state constraints as we saw in* (3.33)*. Additionally, the relaxed RES-CLF condition also enables incorporating safety constraints through barriers (see Section 5.1), as well as enabling modification of the controller to increase the robustness of the closed-loop system (see Section 4.1.2). However, it must be noted that the relaxation of the RES-CLF condition could lead to potential instability. In Appendix B, we establish sufficient conditions under which the relaxed CLF-QP controller can still retain the exponential stability of the hybrid periodic orbit.*

3.5 Summary

We have presented background on periodic walking gait optimization using Hybrid Zero Dynamics and the nonlinear controller using control Lyapunov functions and its incorporation into a quadratic program with constraints that can guarantee exponential stability of the closed-loop control system. However, since the design of CLF is based on the full nonlinear dynamics of the robot, it will therefore be sensitive with model uncertainty. To address this shortcoming, in the next Chapter, we will explore the effects of model uncertainty and then propose robust and adaptive control based on CLF-QP.

Chapter 4

Control Lyapunov Function based Controller and Model Uncertainty

This chapter will present two control schemes to address model uncertainy including robust CLF-QP and L1 adaptive control. The robust CLF-QP handles uncertainty by expressing a min-max problem that is converted to a quadratic program and is also able to address robust safety and robust constraints as we will see later in Chapter. 8. The L1 adaptive control method on the other hand can estimate and compensate model uncertainty whenever it arises.

Notations	Model types
f, g	true nonlinear model
$\tilde{f}, \tilde{g}$	nominal nonlinear model
$\bar{f}, \bar{g}$	true I-O linearized model
$\tilde{\bar{f}}, \tilde{\bar{g}}$	nominal I-O linearized model

Table 4.1: A list of notations for different models used in this Section. A true model represents the actual (possibly not perfectly known) model of the physical system, while the nominal model represents the model that the controller uses. While most controllers assume the true model is known, the robust and adaptive controllers in this work use the nominal models and offer robustness guarantees to the uncertainty between the two models.

4.1 Robust Control Lyapunov Functions based Quadratic Programs

4.1.1 Adverse Effects of Uncertainty in Dynamics on the CLF-QP controller

In order to analyze the effect of model uncertainty in our controllers, we assume that the vector fields, $f(x), g(x)$ of the real dynamics (3.6), are unknown. Instead, we have to design our controller based on the nominal vector fields $\tilde{f}(x), \tilde{g}(x)$. Then, the pre-control law (3.9) get's reformulated as

$$u(x) = \tilde{u}_{ff}(x) + (L_{\tilde{g}} L_{\tilde{f}} y(x))^{-1} \mu, \tag{4.1}$$

with

$$\tilde{u}_{ff}(x) := -(L_{\tilde{g}} L_{\tilde{f}} y(x))^{-1} L_{\tilde{f}}^2 y(x), \tag{4.2}$$

where we have used the nominal model rather than the unknown real dynamics. Substituting $u(x)$ from (4.1) into (3.7), the input-output linearized system then becomes

$$\ddot{y} = \mu + \Delta_1 + \Delta_2 \mu, \tag{4.3}$$

where $\Delta_1 \in \mathbb{R}^m, \Delta_2 \in \mathbb{R}^{m \times m}$, are given by,

$$\begin{aligned} \Delta_1 &:= L_f^2 h(x) - L_g L_f h(x) (L_{\tilde{g}} L_{\tilde{f}} h(x))^{-1} L_{\tilde{f}}^2 h(x), \\ \Delta_2 &:= L_g L_f h(x) (L_{\tilde{g}} L_{\tilde{f}} h(x))^{-1} - I. \end{aligned} \tag{4.4}$$

Remark 4.1. *In the definitions of Δ_1, Δ_2, note that when there is no model uncertainty, i.e., $\tilde{f} = f, \tilde{g} = g$, then $\Delta_1 = \Delta_2 = 0$.*

Using F and G as in (3.15), the closed-loop system now takes the form

$$\dot{\eta} = F\eta + \begin{bmatrix} O \\ \Delta_1 \end{bmatrix} + (G + \begin{bmatrix} O \\ \Delta_2 \end{bmatrix})\mu. \tag{4.5}$$

In fact for $\Delta_1 \neq 0$, the closed-loop system does not have an equilibrium, and for $\Delta_2 \neq 0$, the controller could potentially destabilize the system. This raises the question of whether it's possible for controllers to account for this model uncertainty, and if so, how do we design such a controller. In particular, the time-derivative of the CLF in (3.22) becomes more complex and dependent on Δ_1, Δ_2.

4.1.2 Robust CLF-QP

Having discussed the effect of model uncertainty on the control Lyapunov function based controllers in Section 4.1.1, we now develop a robust controller that can guarantee tracking and stability in the presence of bounded uncertainty. As we will see, both stability and tracking performance (rate of convergence and errors going to zero) are still retained for all uncertainty within a particular bound. For uncertainty that exceeds the specified bound, there is graceful degradation in performance.

We start with the closed-loop model with uncertainty as developed in (4.5) and develop the robust controller as follows. Note that for the following sections, we will abuse notation and redefine $\bar{f}, \bar{g}$ from (3.14) as

$$\bar{f} = F\eta + \begin{bmatrix} \mathbf{0} \\ \Delta_1 \end{bmatrix}, \quad \bar{g} = G + \begin{bmatrix} \mathbf{0} \\ \Delta_2 \end{bmatrix}, \tag{4.6}$$

where the **0** denotes a vector / matrix with all zero elements. With the CLF defined in (3.21), we then have:

$$\dot{V}_\varepsilon = L_{\bar{f}} V_\varepsilon(\eta) + L_{\bar{g}} V_\varepsilon(\eta)\mu, \tag{4.7}$$

where,

$$\begin{aligned} L_{\bar{f}} V_\varepsilon(\eta) &= \eta^T (F^T P_\varepsilon + P_\varepsilon F)\eta + 2\eta^T P_\varepsilon \begin{bmatrix} \mathbf{0} \\ \Delta_1 \end{bmatrix}, \\ L_{\bar{g}} V_\varepsilon(\eta) &= 2\eta^T P_\varepsilon (G + \begin{bmatrix} \mathbf{0} \\ \Delta_2 \end{bmatrix}) = 2\eta^T P_\varepsilon G(I + \Delta_2). \end{aligned} \tag{4.8}$$

Remark 4.2. *Note that the last equality in* (4.8) *follows from the fact that* $G = \begin{bmatrix} O \\ I \end{bmatrix}$ *from* (3.15)*. This is a consequence of the pre-control used in* (3.9)*, and enables formulating the maximization problem described next in a quadratic program (see Appendix A.3 for more details.) In particular, using a pre-control* $u(x, \mu) = u_{ff}(x) + \mu$*, would not have given us this particular* G*.*

The RES condition (3.20) now becomes:

$$\dot{V}_\varepsilon(\eta, \Delta_1, \Delta_2, \mu) + \frac{c_3}{\varepsilon} V_\varepsilon \leq 0. \tag{4.9}$$

In general, we can not satisfy this inequality for all possible unknown Δ_1, Δ_2. To address this, we assume the uncertainty is bounded as follows

$$\|\Delta_1\| \leq \Delta_1^{max}, \qquad \|\Delta_2\| \leq \Delta_2^{max}, \tag{4.10}$$

where the first norm is a vector norm, while the second norm is a matrix norm.

Under this assumption, we have the following robust RES condition:

$$\max_{\substack{||\Delta_1|| \leq \Delta_1^{max} \\ ||\Delta_2|| \leq \Delta_2^{max}}} \dot{V}_\varepsilon(\eta, \Delta_1, \Delta_2, \mu) + \frac{c_3}{\varepsilon} V_\varepsilon \leq 0. \tag{4.11}$$

Note that choosing μ that satisfies (4.11) implies (3.24) for every Δ_1, Δ_2 satisfying $||\Delta_1|| \leq \Delta_1^{max}, ||\Delta_2|| \leq \Delta_2^{max}$.

Then, we can express (4.11) as the following QP:

Robust CLF-QP:

$$\begin{aligned} u^*(x) = \underset{u,\mu}{\text{argmin}} \quad & \mu^T \mu && (4.12) \\ \text{s.t.} \quad & \max_{\substack{||\Delta_1|| \leq \Delta_1^{max} \\ ||\Delta_2|| \leq \Delta_2^{max}}} \dot{V}_\varepsilon(\eta, \Delta_1, \Delta_2, \mu) + \frac{c_3}{\varepsilon} V_\varepsilon \leq 0 && \textbf{(Robust CLF)} \\ & u = \tilde{u}_{ff}(x) + (L_{\tilde{g}} L_{\tilde{f}} y(x))^{-1} \mu. && \textbf{(IO)} \end{aligned}$$

Appendix A.3 illustrates how the above optimization problem can be expressed as a QP.

Remark 4.3. *Note that since $V(\eta)$ is a local CLF for the nonlinear system* (3.6) *and the fact that $\dot{V}$ depends continuously on Δ_1, Δ_2, there always exists μ satisfying* (4.11) *for a local region around $\eta = 0, \Delta_1 = 0, \Delta_2 = 0$. Moreover, if $\Delta_1^{max}, \Delta_2^{max}$ are chosen to be within this region, then the above QP is guaranteed to be feasible.*

4.1.3 Robust CLF-QP with Constraints

Having developed the robust version of the CLF-QP based controller, we now incorporate constraints into the robust control formulation. We note that this first formulation is for non-robust constraints, i.e., these constraints are only evaluated on the nominal model available to the controller. Thus this control is valid and makes sense for only those constraints that are not dependent on the true model.

The incorporation of constraints into the CLF-QP controller required relaxation of the RES condition. Similarly, here we relax the robust RES condition to obtain,

---

Robust CLF-QP with Constraints:

$$
\begin{aligned}
u^*(x) = \underset{u,\mu,d}{\operatorname{argmin}} \quad & \mu^T\mu + pd^2 && (4.13)\\
\text{s.t.} \quad & \max_{\substack{||\Delta_1||\le\Delta_1^{max}\\ ||\Delta_2||\le\Delta_2^{max}}} \dot{V}_\varepsilon(\eta,\Delta_1,\Delta_2,\mu) + \frac{c_3}{\varepsilon}V_\varepsilon \le \delta, && \textbf{(Robust CLF)}\\
& A_c(x)u \le b_c(x), && \textbf{(Constraints)}\\
& u = \tilde{u}_{ff}(x) + (L_{\tilde{g}}L_{\tilde{f}}y(x))^{-1}\mu. && \textbf{(IO)}
\end{aligned}
$$

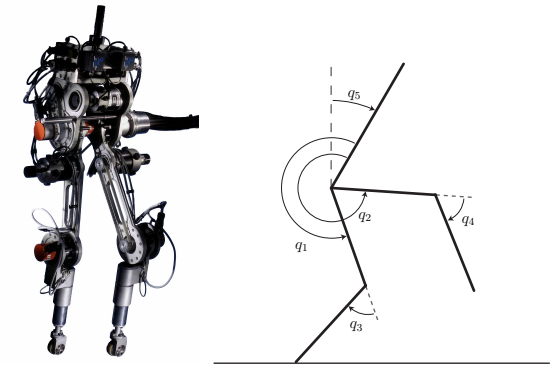
Remark 4.4. *It's critical to note that the additional constraints in* (4.13) *are not robust, and that this method is only applicable for constraints that are invariant to model uncertainty. Constraints such as torque saturation, like in* (3.34)*, are invariant to model uncertainty since the control inputs are computed directly from the nominal model $\tilde{f}(x), \tilde{g}(x)$, and do not depend on the real model $f(x), g(x)$. However, constraints with the model uncertainty invariance property are not common, and we will have to explicitly address and capture the effect of uncertainty on constraints. We will address this in Section 8.2.3.*

Remark 4.5. *Note that, as illustrated in Appendix A.3, the max CLF inequality condition in the above Robust CLF-QPs can be converted to two simple linear inequality constraints, thus ensuring the above min-max problems are actually quadratic programs.*

4.1.4 Numerical Validation of Robust CLF with Torque Saturation

To demonstrate the effectiveness of the proposed robust CLF-QP controller, we will conduct numerical simulations on the model of RABBIT (shown in Figure 4.1), a planar five-link bipedal robot with a torso and two legs with revolute knees that terminate in point feet. RABBIT weighs 32 kg, has four brushless DC actuators with harmonic drives to control the hip and knee angles, and is connected to a rotating boom which constrains the robot to walk in a circle, approximating planar motion in the sagittal plane. Further description of RABBIT and the associated mathematical model can be found in [15, 78]. Fundamental issues in dynamic walking and running on RABBIT can be found in [78] and [52].

For RABBIT, the stance phase is parametrized by a suitable set of coordinates, given by $q := (q_1, q_2, q_3, q_4, q_5)$ and as illustrated in Fig.4.1. Here, q_1 and q_2 are the femur angles (referenced to the torso), q_3 and q_4 are the knee angles, and q_5 is the absolute angle of the torso. Because RABBIT has point feet (while many other legged robots have flat feet), the stance phase dynamics are underactuated with the system possessing 4 actuated degrees-of-freedom (DOF) and 1 underactuated DOF.



(a) RABBIT (b) Coordinate system

Figure 4.1: (a) RABBIT, a planar five-link bipedal robot with nonlinear, hybrid and underactuated dynamics. (b) The the associated generalized coordinate system used, where q_1, q_2 are the relative stance and swing leg femur angles referenced to the torso, q_3, q_4 are the relative stance and swing leg knee angles, and q_5 is the absolute torso angle in the world frame.

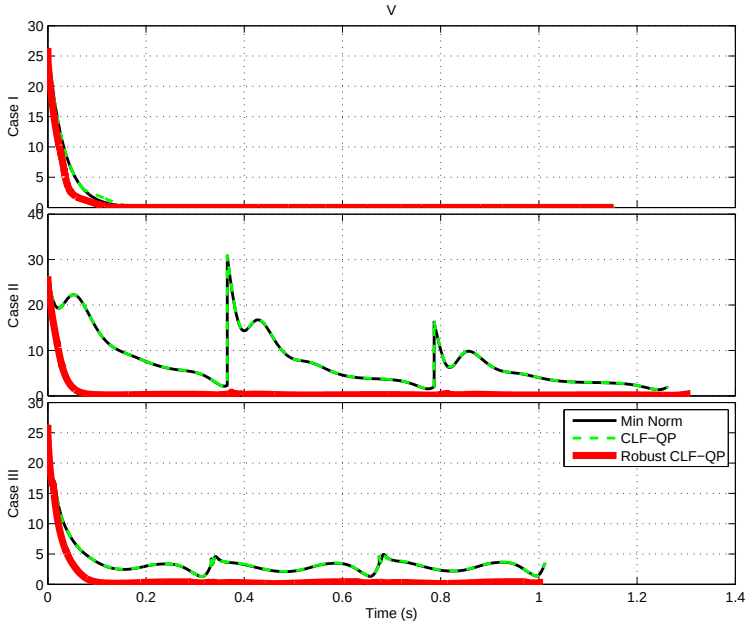


Figure 4.2: Control Lyapunov function for the three controllers: Min Norm, CLF-QP, Robust CLF-QP, for the cases I-III of model perturbations. Notice that the convergence performance (both rate of convergence and tracking error being driven to zero) of the proposed robust CLF-QP controller remains the same across the different model perturbations.

For the purpose of evaluating the robust CLF-QP controller, we will consider a periodic walking gait, with step speed of 0.9 m/s and step length of 0.45 m, that is developed for a nominal model of RABBIT. The controller is also developed for this nominal model. The simulation is then carried out on a perturbed model of RABBIT, where the perturbation is introduced by scaling all mass and inertia parameters of each link by a fixed constant scale factor. The perturbed model is unknown to the controller and will serve as an uncertainty injected into the model. We will illustrate four separate cases of scaling the mass and inertia:

$$\begin{aligned} Case\ I &: \text{model scale = 1} \\ Case\ II &: \text{model scale = 1.5} \\ Case\ III &: \text{model scale = 0.7} \\ Case\ IV &: \text{model scale = 3;} \end{aligned}$$

For each case, we will compare the quality of three controllers

$$\begin{aligned} Controller\ A: &\quad \text{Min-norm Controller (3.25)} \\ Controller\ B: &\quad \text{CLF-QP torque saturation controller (3.34)} \\ Controller\ C: &\quad \text{CLF-QP robust controller (A.16).} \end{aligned}$$

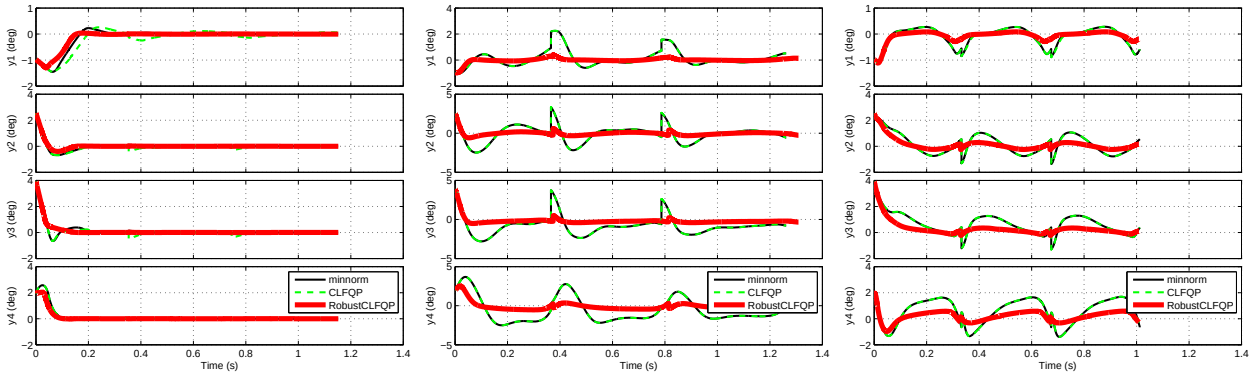
Note that controllers B, C enable limiting the torque to be within a user specified bound. For a fair comparison, we set these torque bounds slightly below the maximum torque that controller A uses for each particular case of model perturbation. In particular, the torque saturations were set as follows: 60 Nm, 80 Nm, 150 Nm, and 300 Nm for the model scale equals to 0.7, 1, 1.5, 3 respectively.

Each of the four models are evaluated with the three sets of controllers. The results are presented in Figures 4.2, 4.3, 4.4, 4.6.

Remark 4.6. *From Figure 4.2, note that the performance (both rate of convergence and driving the tracking error to zero) of the proposed robust CLF-QP controller remains same for all the considered model variations.*

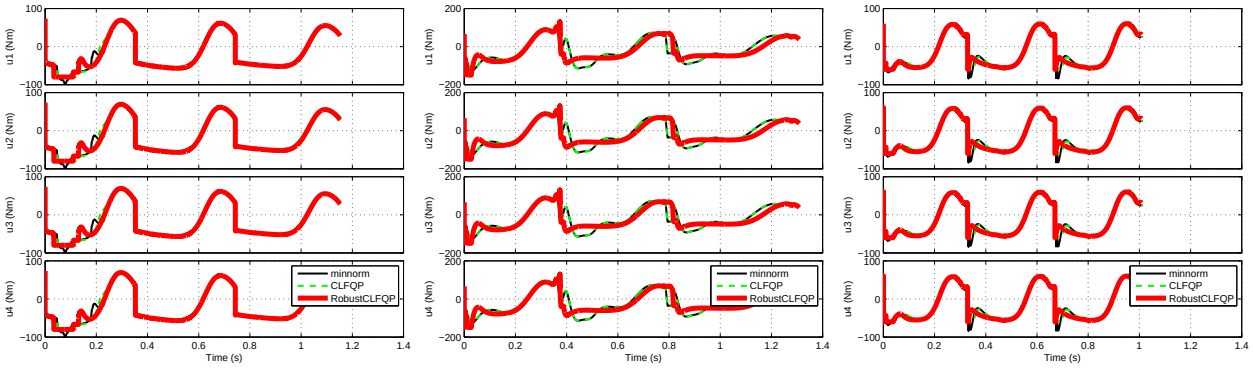
Remark 4.7. *For Case I: model scale = 1, when there is no uncertainty, we can observe from the top plot of Figure 4.2 and from Figure 4.3a that the robust CLF-QP controller (controller C) performs better than the two other controllers. This is because the robust controller is trying to satisfy the convergence bound on the worst possible model perturbation. Also notice that the CLF-QP with torque saturation (controller B) does slightly worse than the min-norm controller (controller A). This is due to the torque saturation that is slightly below the maximum torque the min-norm controller uses.*

For Case II: model scale = 1.5 and Case III: model scale = 0.7, where mass and inertia were increased by



(a) Case I: model scale = 1 (b) Case II: model scale = 1.5 (c) Case III: model scale = 0.7

Figure 4.3: Tracking errors of the control outputs based on the simulation of Cases I-III of perturbed model of RABBIT with three controllers as described in Section 4.1.4. Simulation of three walking steps are shown.



(a) Case I: model scale = 1 (b) Case II: model scale = 1.5 (c) Case III: model scale = 0.7

Figure 4.4: Control inputs (motor torques for stance and swing legs) based on the simulation of cases I-III of perturbed model of RABBIT with three controllers as described in Section 4.1.4. Simulation of three walking steps are shown.

50% and reduced by 30% respectively, we can observe from the tracking error plots in Figures 4.3b, 4.3c, that while the min-norm controller (controller A) and the CLF-QP torque saturation controller (controller B) have non-zero tracking errors during each step, the robust CLF-QP controller (controller C) still performs perfectly, with tracking errors converging to zero. Thus, even with this level of uncertainty, the proposed robust CLF-QP controller is able to maintain the same control performance quality as in Case I (no model uncertainty). This is especially evident in Figure 4.2, where we clearly see that the robust CLF-QP maintains the same rate of convergence for all model perturbations. Note that these comparisons are fair between the controllers since we maintain the same range of control inputs for each case of model perturbation, as evidenced in the control input plots in Figures 4.4.

In Case IV: model scale = 3, where we set up a highly significant level of uncertainty by scaling all mass and inertia properties by a factor of 3, while the Controllers A and B fail to sustain walking, the proposed CLF-QP controller is still able to regulate the outputs with only a slight degradation, see Figure 4.5. However, it must be noted that the underlying periodic orbit $\mathscr{O}_Z$ is unstable, and the walking slows down after several steps to a complete stop. Nonetheless, the controller is able to perform its task with such a large model perturbation. Future research will be directed towards addressing what can potentially be done for handling the effect of uncertainty for the underactuated dynamics.

Figure 4.6 illustrates the simulation of 25 steps of the proposed CLF-QP based robust controller for three cases of uncertainty as specified in Case I-III. This illustrates the controller converging to three periodic orbits that slightly vary from each other. As mentioned, the uncertainty in the model causes the underlying periodic orbit to change.

Although the proposed robust CLF-QP based controller appears impressive, a primary limitation of the controller is that it always assumes the worst-case scenario with maximum model perturbation within the specified bounds, even when the model is known perfectly and there is no uncertainty. This results in the controller being unnecessarily aggressive, and may cause problems in experiments.

4.2 L1 Adaptive Control with Application on Bipedal Walking

As presented in the previous Section, while the robust CLF-QP controller can guarantee robustness via quadratic constraints that can be extended further to robust safety-critical control, one shortcoming of the method, and robust control in general, is that it always assumes the worst-case scenario and therefore is aggressive even without or with a small amount of model uncertainty. In order to address this issue, in this Section, we will present the method of using L1 adaptive control to estimate and adapt to the presence of model uncertainty in the system dynamics, such that the controller has the same performance as the baseline controller if there is no model uncertainty. L1 adaptive control based approach therefore has smother and

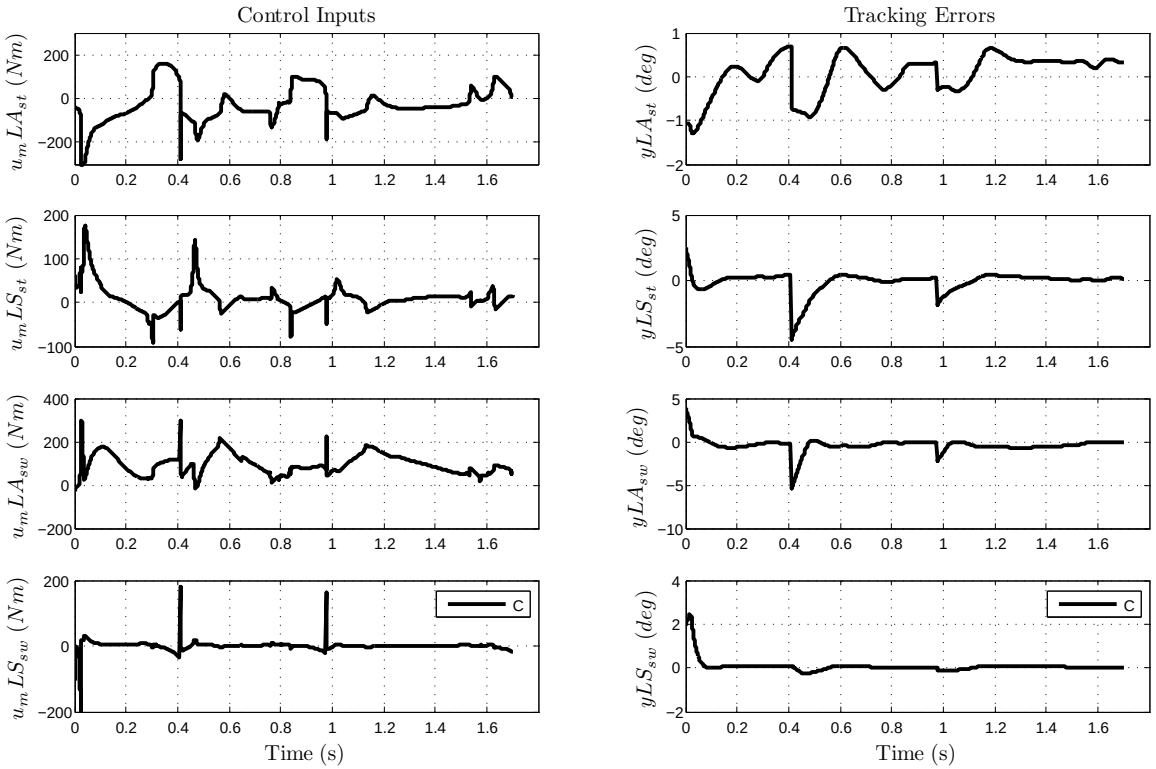


Figure 4.5: Tracking errors and control outputs for Case IV: model scale = 3.

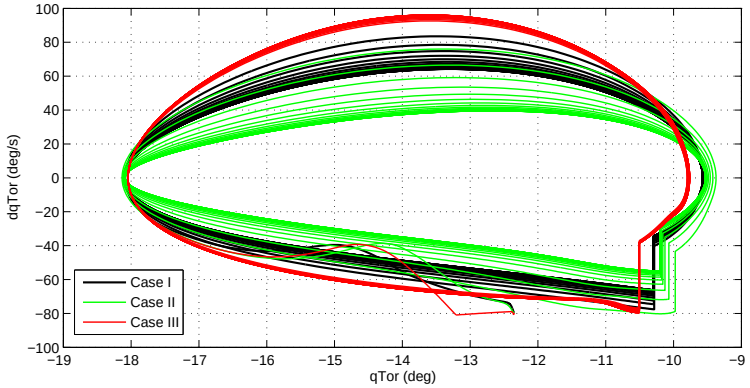


Figure 4.6: Phase portrait of the torso angle for walking simulation in 25 steps for Cases I-III with the proposed CLF-QP robust controller, (A.16). As is evident, due to the different levels of uncertainty in the model, the walking settles to three different periodic orbits for each of the three cases respectively. Also notice that for a heavier robot (Case II with mass scale = 1.5), the velocities are slower.

smaller control inputs in most of the cases.

4.2.1 Adverse Effects of Uncertainty in Dynamics on the CLF-QP controller

Similar to what we have done in Section. 4.1.1, we will analyse the effects of model uncertainty on the CLF controller so that we can design a L1 adaptive controller to compensate these effects. If we consider uncertainty in the dynamics and assume that the functions, $f(x), g(x)$ of the real dynamics (3.6), are unknown, we then have to design our controller based on nominal functions $\tilde{f}(x), \tilde{g}(x)$. Thus, the pre-control law (3.9) is reformulated as (4.1)

Substituting $u(x)$ from (4.1) into (3.7), the second derivative of the output, $y(x)$, then becomes

$$\ddot{y} = \mu + \theta, \tag{4.14}$$

where,

$$\begin{aligned} \theta &= \Delta_1 + \Delta_2 \mu \\ \Delta_1 &= L_f^2 y(x) - L_g L_f y(x) (L_{\tilde{g}} L_{\tilde{f}} y(x))^{-1} L_{\tilde{f}}^2 y(x), \\ \Delta_2 &= L_g L_f y(x) (L_{\tilde{g}} L_{\tilde{f}} y(x))^{-1} - I. \end{aligned} \tag{4.15}$$

The closed-loop system now takes the form

$$\dot{\eta} = F\eta + G(\mu + \theta). \tag{4.16}$$

where F and G are defined in (3.15).

Clearly for $\theta \neq 0$, the closed-loop system does not have an equilibrium and therefore the PD control (3.16) does not stabilize the system dynamics. This raises the question of whether it's possible for controllers to account for this model uncertainty, and if so, how do we design such a controller. Note that the uncertainty θ is a nonlinear function of (x, μ), and therefore a nonlinear function of η and time (since $(\eta, z) = \Phi(x), \mu = \mu(\eta), z = z(t)$).

4.2.2 L1 Adaptive Control with Control Lyapunov Function based Quadratic Program

From the Section 4.2.1, we have the system with uncertainty described by (4.16) where the nonlinear uncertainty $\theta = \theta(\eta, t)$. As a result, for every time t, we can always find out $\alpha(t)$ and $\beta(t)$ such that [12]:

$$\theta(\eta, t) = \alpha(t)||\eta|| + \beta(t) \tag{4.17}$$

The principle of our method is to design a combined controller $\mu = \mu_1 + \mu_2$, where μ_1 is to control the model to follow the desired reference model and μ_2 is to compensate the nonlinear uncertainty θ. The reference model could be linear when we apply conventional PD control (3.16) for μ_1.

In the perfect case, when there is no model uncertainty, we will have the following desired linear model with PD control

$$\dot{\eta} = A_m \eta \tag{4.18}$$

where $A_m = A$ in (3.18), which serves as a linear reference model for L_1 adaptive control.

In this work, we present a method to consider a reference model for L_1 adaptive control that arises from a rapidly exponentially stabilizing CLF-based controller. In particular, we consider the reference model that arises when μ_1 is chosen to be the solution of the QP (3.27). This reference model is nonlinear and has no closed-form analytical expression.

The state predictor can then be expressed as follows,

$$\dot{\hat{\eta}} = F\hat{\eta} + G\hat{\mu}_1 + G(\mu_2 + \hat{\theta}), \tag{4.19}$$

where,

$$\hat{\theta} = \hat{\alpha}||\eta|| + \hat{\beta}, \tag{4.20}$$

and $\hat{\mu}_1$ is computed as the solution of the following QP:

CLF-QP for the state predictor:

$$\begin{aligned} \underset{\hat{\mu}_1, d_1}{\text{argmin}} \quad & \hat{\mu}_1^T \hat{\mu}_1 \\ \text{s.t.} \quad & \psi_{0,\varepsilon}(\hat{\eta}, z) + \psi_{1,\varepsilon}(\hat{\eta}, z)\, \hat{\mu}_1 \leq 0, \end{aligned} \tag{4.21}$$

---

where $\psi_{0,\varepsilon}, \psi_{0,\varepsilon}$ are defined in (3.26).

In order to compensate the estimated uncertainty $\hat{\theta}$, we can just simply choose $\mu_2 = -\hat{\theta}$ to obtain

$$\dot{\hat{\eta}} = F\hat{\eta} + G\hat{\mu}_1 \tag{4.22}$$

which satisfies the rapid exponential stability [4] since $\hat{\mu}_1$ is designed through the CLF-QP in (4.21). However, $\hat{\theta}$ typically has high-frequency content due to fast estimation. For the reliability of the control scheme and in particular to not violate the unilateral ground force constraints for bipedal walking, it is very important to not have high-frequency content in the control signals. Thus, we apply the L_1 adaptive control scheme to decouple estimation and adaptation [10]. Therefore, we will have

$$\mu_2 = -C(s)\hat{\theta} \tag{4.23}$$

where $C(s)$ is a low-pass filter with magnitude being 1.

Define the difference between the real model and the reference model $\tilde{\eta} = \hat{\eta} - \eta$, we then have,

$$\dot{\tilde{\eta}} = F\tilde{\eta} + G\tilde{\mu}_1 + G(\tilde{\alpha}||\eta|| + \tilde{\beta}), \tag{4.24}$$

where

$$\tilde{\mu}_1 = \hat{\mu}_1 - \mu_1,\ \tilde{\alpha} = \hat{\alpha} - \alpha,\ \tilde{\beta} = \hat{\beta} - \beta. \tag{4.25}$$

As a result, we will estimate θ indirectly through α and β, or the values of $\hat{\alpha}$ and $\hat{\beta}$ computed by the following adaptation laws based on the projection operators [42],

$$\begin{aligned} \dot{\hat{\alpha}} &= \Gamma\textbf{Proj}(\hat{\alpha}, y_\alpha), \\ \dot{\hat{\beta}} &= \Gamma\textbf{Proj}(\hat{\beta}, y_\beta). \end{aligned} \tag{4.26}$$

where Γ is a symmetric positive define matrix.

We now have the control diagram of the L_1 adaptive control with CLF-QP described in Fig. 4.7.

In order to find out a suitable function y_α and y_β for the adaptation laws in (4.26), we will consider the following control Lyapunov candidate function

$$\tilde{V} = \tilde{\eta}^T P_\varepsilon \tilde{\eta} + \tilde{\alpha}^T \Gamma^{-1} \tilde{\alpha} + \tilde{\beta}^T \Gamma^{-1} \tilde{\beta} \tag{4.27}$$

Because $\tilde{\eta} = \hat{\eta} - \eta$ satisfies the RES condition imposed by the two CLF-QP (3.27) and (4.21), it implies

that

$$(F\tilde{\eta} + G\tilde{\mu}_1)^T P_\varepsilon \tilde{\eta} + \tilde{\eta}^T P_\varepsilon (F\tilde{\eta} + G\tilde{\mu}_1) \leq -\frac{c_3}{\varepsilon}\tilde{\eta}^T P_\varepsilon \tilde{\eta} \tag{4.28}$$

Furthermore, with the property of projection operator [42], we have:

$$\begin{aligned}(\hat{\alpha} - \alpha)^T(\mathbf{Proj}(\hat{\alpha}, y_\alpha) - y_\alpha) \leq 0,\\ (\hat{\beta} - \beta)^T(\mathbf{Proj}(\hat{\beta}, y_\beta) - y_\beta) \leq 0.\end{aligned} \tag{4.29}$$

So, if we choose the projection functions y_α and y_β as,

$$\begin{aligned}y_\alpha &= -G^T P_\varepsilon \tilde{\eta}||\eta||,\\ y_\beta &= -G^T P_\varepsilon \tilde{\eta},\end{aligned} \tag{4.30}$$

then from (4.28), (4.29), we will have

$$\begin{aligned}\dot{\tilde{V}} + \frac{c_3}{\varepsilon}\tilde{V} \leq \frac{c_3}{\varepsilon}\tilde{\alpha}^T\Gamma^{-1}\tilde{\alpha} + \frac{c_3}{\varepsilon}\tilde{\beta}^T\Gamma^{-1}\tilde{\beta}\\ -\tilde{\alpha}^T\Gamma^{-1}\dot{\alpha} - \dot{\alpha}^T\Gamma^{-1}\tilde{\alpha}\\ -\tilde{\beta}^T\Gamma^{-1}\dot{\beta} - \dot{\beta}^T\Gamma^{-1}\tilde{\beta}.\end{aligned} \tag{4.31}$$

We assume that the uncertainties α, β and their time derivatives are bounded. Furthermore, the projection operators (4.26) will also keep $\tilde{\alpha}$ and $\tilde{\beta}$ bounded (see [12] for a detailed proof about these properties.) We define these bounds as follows:

$$\begin{aligned}||\tilde{\alpha}|| \leq &\tilde{\alpha}_b, \;\; ||\tilde{\beta}|| \leq \tilde{\beta}_b,\\ ||\dot{\alpha}|| \leq &\dot{\alpha}_b, \;\; ||\dot{\beta}|| \leq \dot{\beta}_b.\end{aligned} \tag{4.32}$$

Combining this with (4.31), we have,

$$\dot{\tilde{V}} + \frac{c_3}{\varepsilon}\tilde{V} \leq \frac{c_3}{\varepsilon}\delta_{\tilde{V}}, \tag{4.33}$$

where

$$\delta_{\tilde{V}} = 2||\Gamma||^{-1}(\tilde{\alpha}_b^2 + \tilde{\beta}_b^2 + \frac{\varepsilon}{c_3}\tilde{\alpha}_b\dot{\alpha}_b + \frac{\varepsilon}{c_3}\tilde{\beta}_b\dot{\beta}_b). \tag{4.34}$$

Thus, if $\tilde{V} \geq \delta_{\tilde{V}}$ then $\dot{\tilde{V}} \leq 0$. As a result, we always have $\tilde{V} \leq \delta_{\tilde{V}}$. In other words, by choosing the

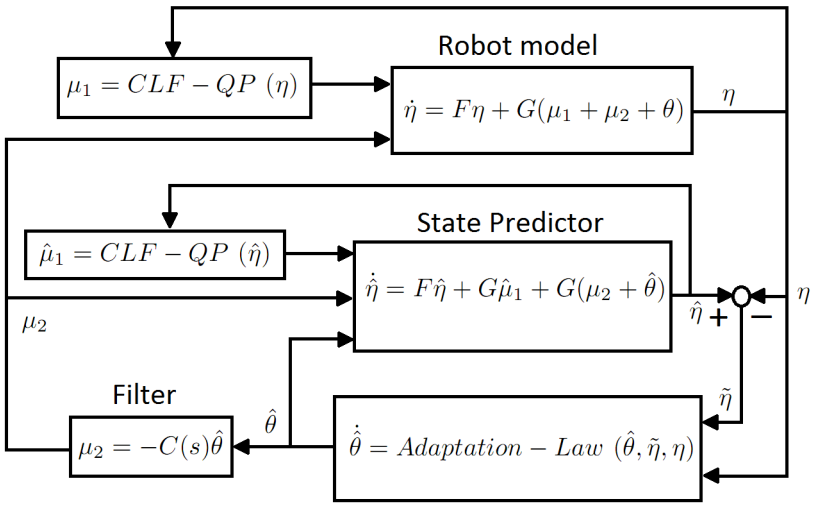


Figure 4.7: Control diagram illustrating L_1 adaptive control with a CLF-QP based closed-loop reference model.

adaptation gain Γ sufficiently large, we can limit the Control Lyapunov Function (4.27) in an arbitrarily small neighborhood $\delta_{\tilde{V}}$ of the origin. Therefore the tracking errors between the dynamics model (4.5) and the reference model (4.19), $\tilde{\eta}$, and the error between the real and estimated uncertainty, $\tilde{\alpha}$, $\tilde{\beta}$ are bounded as follows:

$$||\tilde{\eta}|| \leq \sqrt{\frac{\delta_{\tilde{V}}}{||P_\varepsilon||}}, ||\tilde{\alpha}|| \leq \sqrt{||\Gamma||\delta_{\tilde{V}}}, ||\tilde{\beta}|| \leq \sqrt{||\Gamma||\delta_{\tilde{V}}}. \tag{4.35}$$

Another interesting property of this controller is that $\delta_{\tilde{V}}$ can also be decreased by choosing a sufficiently small $\varepsilon < \bar{\varepsilon}$.

4.2.3 L_1 Adaptive Control with Control Lyapunov Function based Quadratic Program and Torque Saturation

CLF-QP based controllers can be extended to incorporate other constraints, such as strict torque constraints as carried out in [25]. This can also be combined with L_1 adaptive control. The controller design is almost equivalent to the L_1 adaptive control with CLF-QP presented in Section 4.2.2. We retain μ_2 as in (4.23) and adaptation laws for $\hat{\alpha}, \hat{\beta}$ as in (4.26), while we redesign μ_1 and $\hat{\mu}_1$ based on the CLF-QP with torque saturation [25], as below:

CLF-QP with torque saturation for the dynamics model:

$$\begin{aligned} \underset{\mu_1, d_1}{\text{argmin}} \quad & \mu_1^T \mu_1 + p\lambda^2 \\ \text{s.t.} \quad & \psi_{0,\varepsilon}(\eta, z) + \psi_{1,\varepsilon}(\eta, z)\, \mu_1 \leq \lambda, \\ & (L_g L_f y(q, \dot{q}))^{-1}\, \mu_1 \geq (u_{min} - u^*), \\ & (L_g L_f y(q, \dot{q}))^{-1}\, \mu_1 \leq (u_{max} - u^*). \end{aligned} \tag{4.36}$$

CLF-QP with torque saturation for the state predictor:

$$\begin{aligned} \underset{\hat{\mu}_1, d_1}{\text{argmin}} \quad & \hat{\mu}_1^T \hat{\mu}_1 + p\lambda^2 \\ \text{s.t.} \quad & \psi_{0,\varepsilon}(\hat{\eta}, z) + \psi_{1,\varepsilon}(\hat{\eta}, z)\, \hat{\mu}_1 \leq \lambda, \\ & (L_g L_f y(q, \dot{q}))^{-1}\, \hat{\mu}_1 \geq (u_{min} - u^*), \\ & (L_g L_f y(q, \dot{q}))^{-1}\, \hat{\mu}_1 \leq (u_{max} - u^*). \end{aligned} \tag{4.37}$$

Here, λ is a relaxation of the stability criterion to respect potentially conflicting torque saturation constraints, and p the penalty for relaxation. Note, that the proposed controller only respects saturation for the CLF-QP controller component (μ_1), and not the L1 adaptive controller component (μ_2).

4.2.4 Numerical Validation of L1 Adaptive Control

Having developed the L_1 adaptive control with CLF-QP, both with and without torque saturation (see Sections 4.2.2, 4.2.3), we now demonstrate the performance of these controllers through numerical simulations and offer comparisions with the standard CLF-QP controller in [4, 25]. We will conduct simulations with the following controllers:

$$\begin{aligned} &Controller\ A: \text{CLF-QP} \\ &Controller\ B: L_1\text{-CLF-QP} \\ &Controller\ C: L_1\text{-CLF-QP with torque saturation} \end{aligned} \tag{4.38}$$

For Controller 'B' and 'C', we choose the bandwidth for the low-pass filter: 150 rad/s (23 Hz). For Controller 'C', we choose the following torque saturations: $u_{max} = u_b; u_{min} = -u_b$ with $u_b = \begin{bmatrix} 65 & 65 & 65 & 65 \end{bmatrix}^T$.

We perform simulations using a model of RABBIT, wherein, the stance phase is parametrized by a suitable set of coordinates as illustrated in Fig. 4.1. Here, q_1 and q_2 are the thigh angles (referenced to the torso), q_3 and q_4 are the knee angles, and q_5 is the absolute angle of the torso. Because RABBIT has point

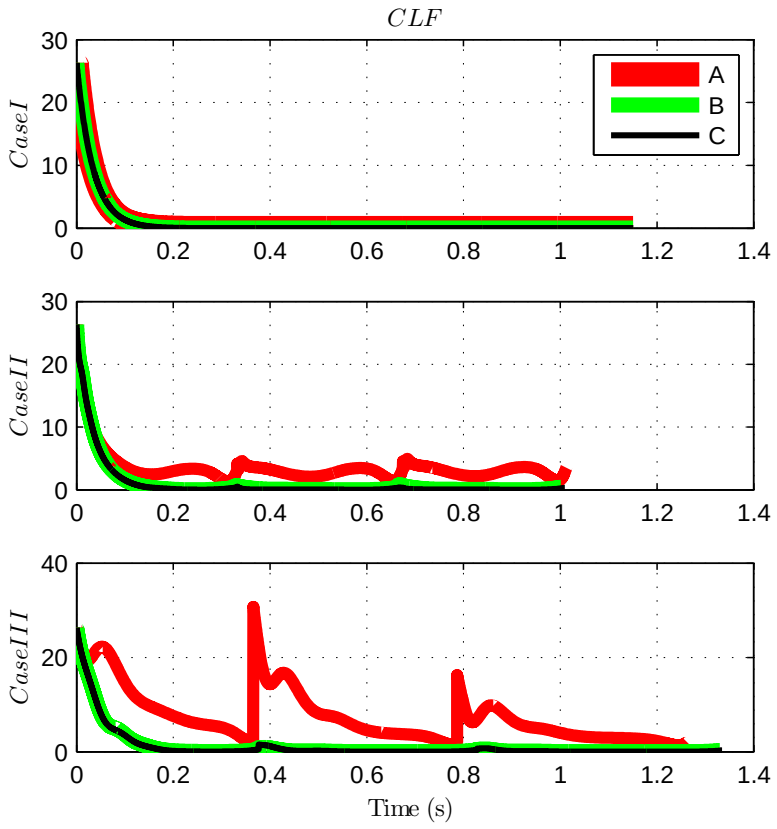


Figure 4.8: Control Lyapunov function for the three controllers A-C indicated in (4.38), for the cases I-III (4.39) of model perturbations. The simulation is for three walking steps.

feet, the stance phase dynamics are underactuated with the system possessing 4 actuated degrees-of-freedom (DOF) and 1 underactuated DOF.

For the purpose of evaluating the L_1 adaptive control with CLF-QP controller, we will consider a periodic walking gait and an associated controller that is developed for a nominal model of RABBIT. The simulation is then carried out on a perturbed model of RABBIT, where the perturbation is introduced by scaling all mass and inertia parameters of each link by a fixed constant scale factor. The perturbed model is unknown to the controller and will serve as an uncertainty injected into the model. We will illustrate three separate cases of scaling the mass and inertia:

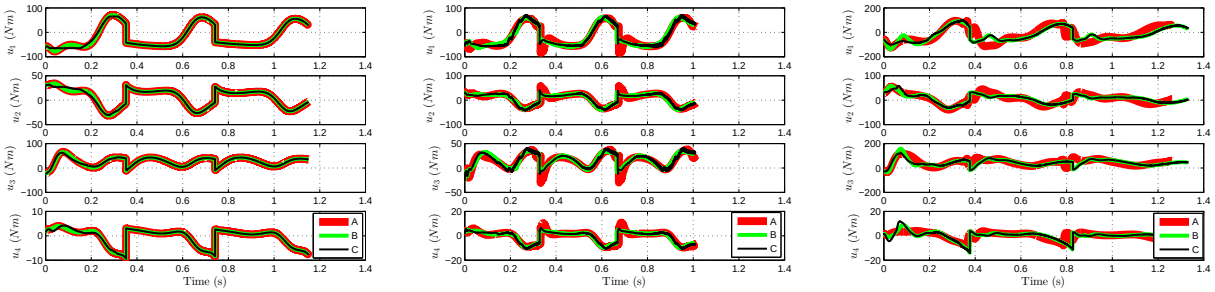
$$\begin{cases} Case\ I & : \quad \text{model scale = 1} \\ Case\ II & : \quad \text{model scale = 0.7} \\ Case\ III & : \quad \text{model scale = 1.5} \end{cases} \tag{4.39}$$

As we can see from the Fig. 4.8, in Case I, when we set the model scale equal to 1, i.e., no uncertainty, the performance of the three controllers in (4.38) are nearly the same.

When we present a high level of uncertainty, Cases II-III with model scale = 0.7 and 1.5, Controller 'A' cannot guarantee a zero tracking error. However, Controllers 'B' and 'C', not only drive the output y to converge to zero but also keep the rate of convergence unchanged through the three cases of model uncertainty. This property is important for bipedal walking since a sufficiently fast rate of convergence is required to guarantee stability of the hybrid system [4]. The rates of convergence of controller 'C' in the Cases II-III are a bit slower than those of the controller 'B' due to the additional constraint of torque saturation.

We note that, the performance of the two L_1 adaptive controllers ('B' and 'C'), are much better than the standard CLF-QP controller ('A'). Further, although the control signals of the three controller are similar, as is evident from close observation of Fig. 4.4, Controller 'C', the torque is saturated. Finally, Fig. 4.10 illustrates the phase plot of the torso angle for the three controllers with a model uncertainty corresponding to case III.

We also apply the L1 adaptive control scheme onto the problem of carrying unknown mass on torso. Our approach is not only able to handle up to load of 94% robot weight, but also deal with under-actuated robot and tracking the entire walking gait. Fig.4.11a shows the first 5 steps in the simulation of carrying randomly unknown mass [0:30] kg. The comparison for the norm of control torques for dynamic walking while carrying 10, 15, and 20 kg unknown masses is shown in Fig. 4.11b.



(a) Case I: model scale = 1 (b) Case II: model scale = 0.7 (c) Case III: model scale = 1.5

Figure 4.9: Control inputs (motor torques for stance and swing legs) based on the simulation of three cases of perturbed model of RABBIT (4.39) with three controllers as described in (4.38). Simulation of three walking steps are shown.

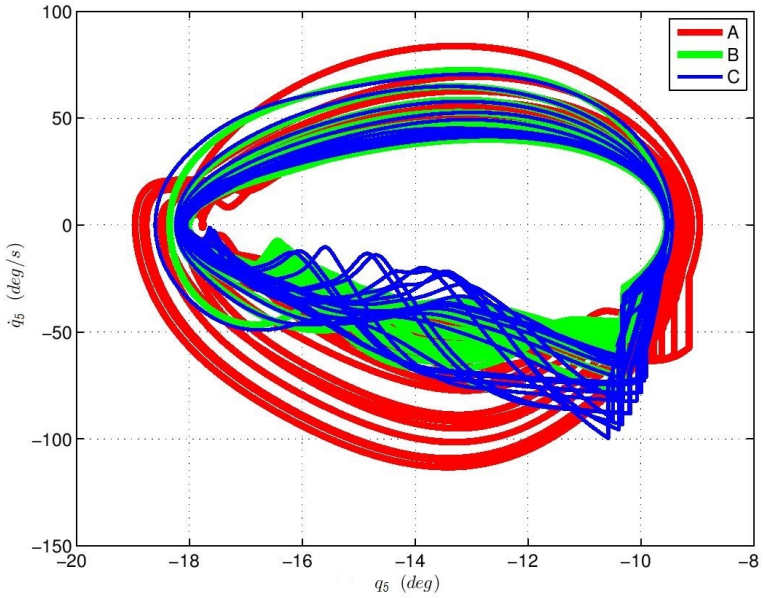
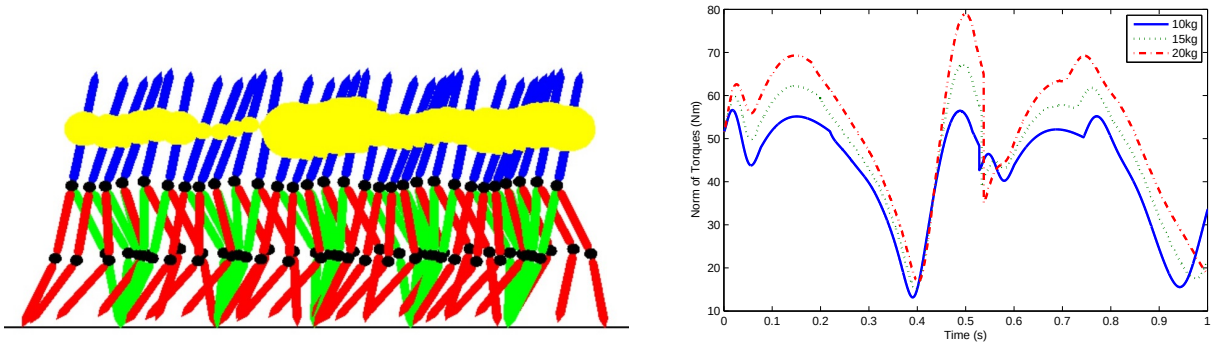


Figure 4.10: Phase portrait of the torso angle for walking simulation for 20 steps for model perturbation Case III (model scale = 1.5) with three different controllers (A-C). Note that the uncertainty causes a change in the periodic orbit. This is as expected, as the controller only tracks the outputs (even in the presence of uncertainty), and the unactuated dynamics on Z evolve passively.



(a) (b)

Figure 4.11: (a) Dynamic bipedal dynamic walking while carrying an unknown load whose mass randomly varies between 0-30 Kg on every step. (b) Norm of control torques for dynamic walking while carrying 10, 15, and 20 Kg unknown masses.

4.2.5 Experimental Validation of L1 Adaptive Control

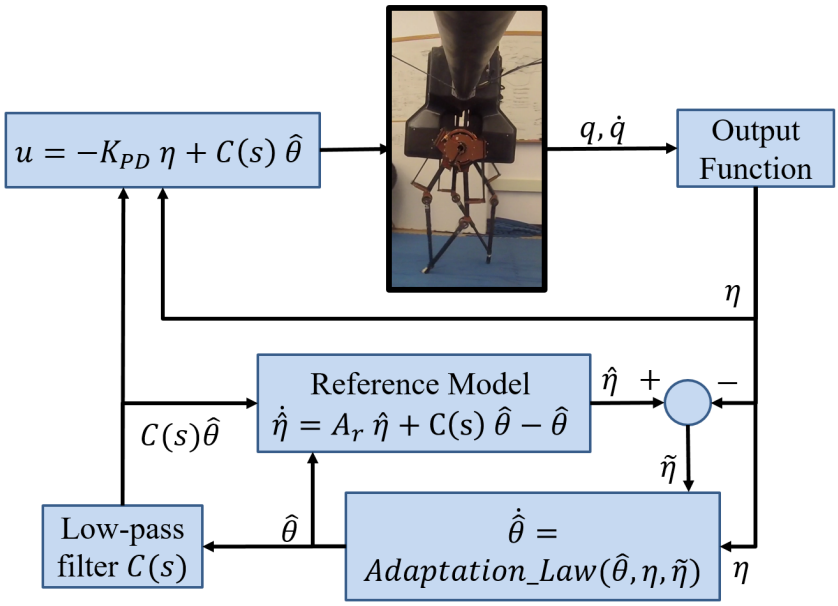
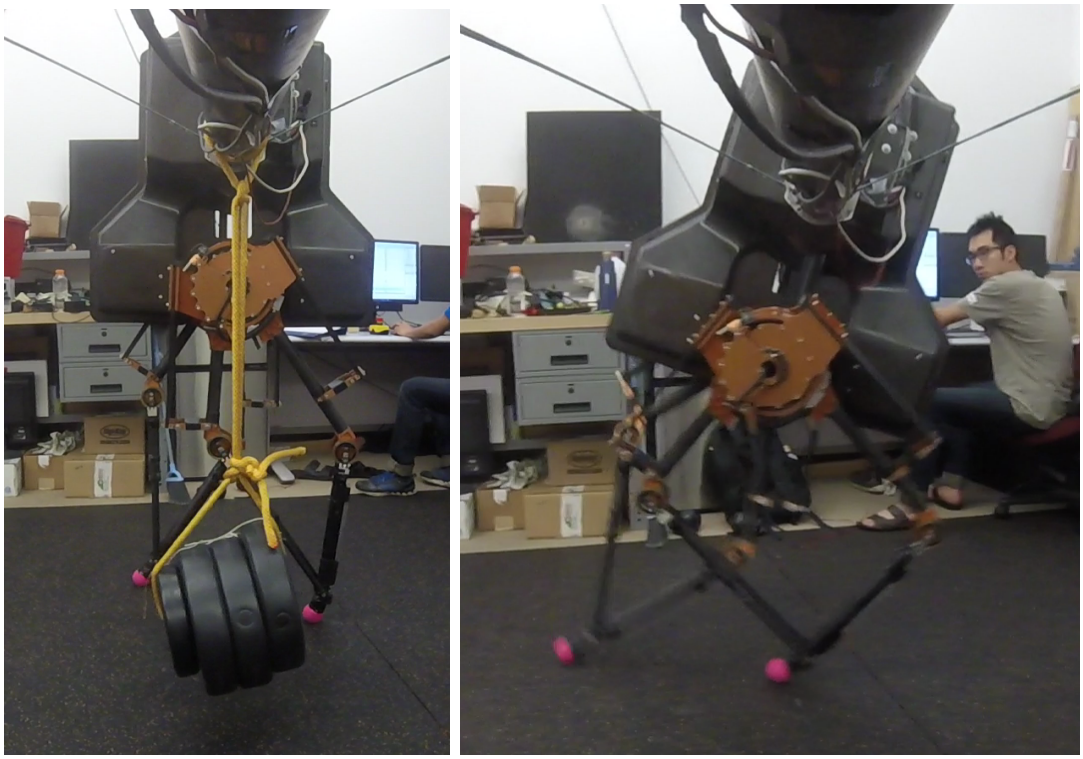


Figure 4.12: Proposed control diagram for experimental validation of L_1 adaptive control scheme with a linear reference model.

As presented in Section. 4.2, we have introduced our theoretical framework on L1 Adaptive Control for bipedal walking. In this section, we will present experimental validation for the method. The control diagram for the experiment is presented in Fig. 4.12.



(a) L1 adaptive control with 68 lb load

(b) PD control with no load

Figure 4.13: Comparison between L_1 adaptive control and PD control. While L1 adaptive control with low-gain PD ($K_P = 200, K_D = 20$) can hold a heavy load of 68 (lb) or 31 (kg) with almost zero errors at all joints (about 0.01 deg); PD controller with $K_P = 500, K_D = 50]$ fails to hold the robot without any load.

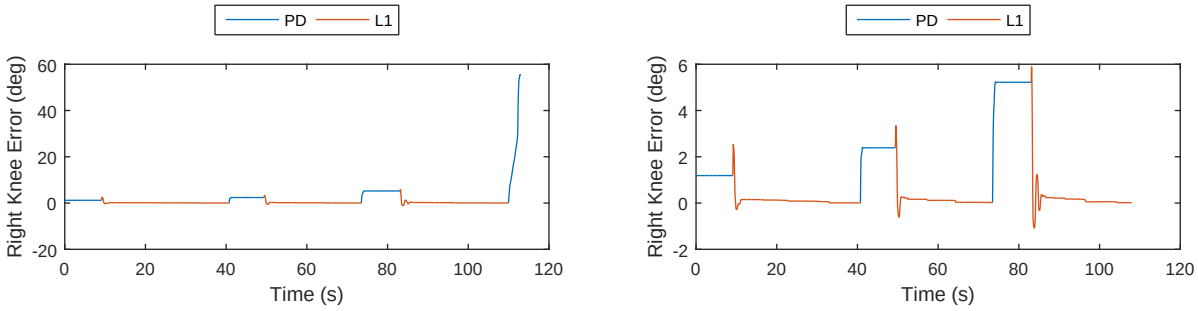


Figure 4.14: Comparison between L_1 adaptive control and PD control. For this experiment, we pose the robot on the ground and compare the performance of L1 adaptive control with low-gain PD ($K_P = 200, K_D = 20$) and PD with different gain. We switch between different controllers with the following order: PD with ($K_P = 4000; K_D = 100$) => L1 => PD with ($K_P = 2000; K_D = 70$) => L1 => PD with ($K_P = 1000; K_D = 50$) => L1 => PD with ($K_P = 500, K_D = 50$). This figure shows the knee angle error during the experiment. Figure (a) shows that PD with ($K_P = 500, K_D = 50$) fails to hold the robot (the knee angle keeps increasing at the end of the experiment). In Figure (b), we zoom in to ignore the last experiment when the robot falls to better compare the tracking errors.

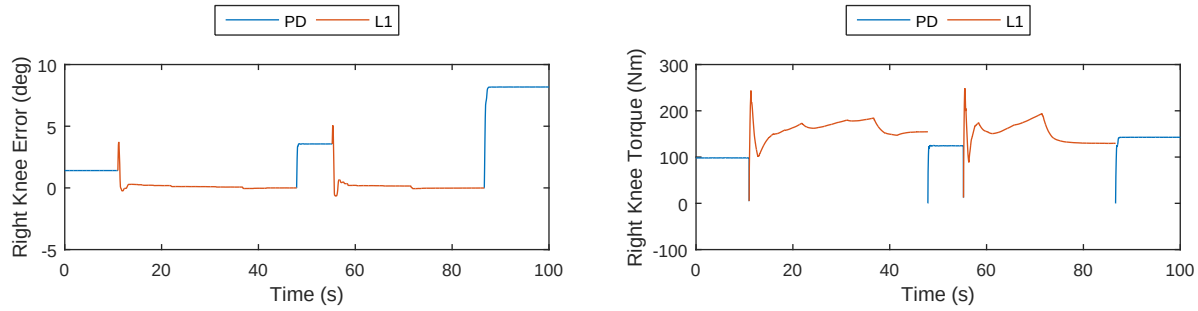
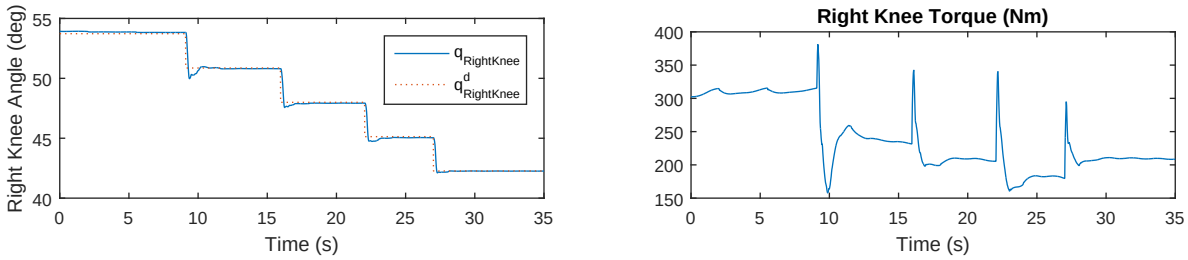
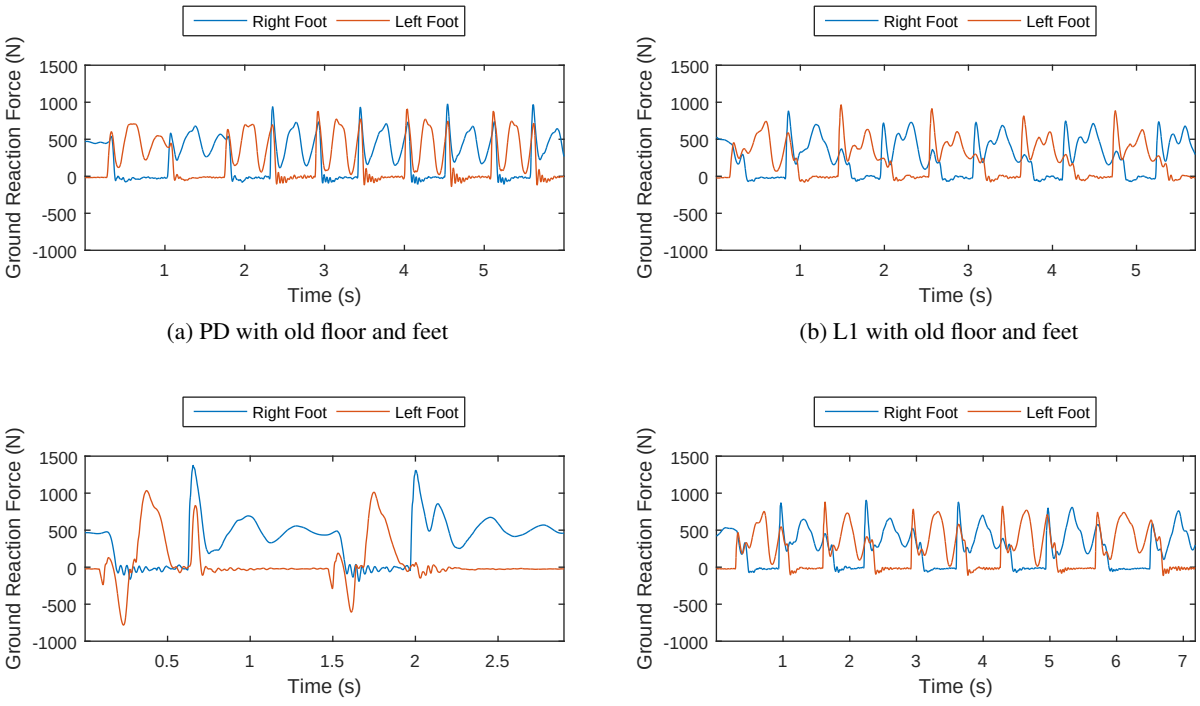


Figure 4.15: Comparison between L_1 adaptive control and PD control. For this experiment, we pose the robot on the ground while carrying an 68 lb of unknown load and compare the performance of L1 adaptive control with low-gain PD ($K_P = 200, K_D = 20$) and PD with different gain. We switch between different controllers with the following order: PD with ($K_P = 4000; K_D = 100$) => L1 => PD with ($K_P = 2000; K_D = 70$) => L1 => PD with ($K_P = 1000; K_D = 50$). This figure shows the plots of (a) tracking error and (b) control input of the right knee angle.



(a) (b)

Figure 4.16: L1 adaptive control with low-gain PD $(K_P = 200; K_D = 20)$ for the problem of posing the robot on the ground while carrying an $68\ lb$ of unknown. For this experiment we control the robot to lift up the weight by reducing the knee angles. This figure shows the plots of (a) output tracking and (b) control input of the right knee angle.



(a) PD with old floor and feet

(b) L1 with old floor and feet

(c) PD with new floor and feet

(d) L1 with new floor and feet

Figure 4.17: Walking experiment with original and new floor and feet: compare L1 adaptive control with low-gain PD $(K_P = 500; K_D = 50)$ and PD control that is often used for our walking experiment with $K_P = [4000; 3000; 6000; 5000], K_D = [100; 70; 200; 150]$ (Note that this gain was hand-tuned to get the best walking experiment possible,therefore there are different gains for each of 4 joints). This figure shows the ground reaction force of the experiment.

In this part, we will present following experiments on the ATRIAS robot:

(a) Posing the robot on the ground: compare L1 adaptive control with low-gain PD $K_{PD} = [200; 20]$ (i.e.,$K_P = 200, K_D = 20$) and PD with different gains $K_{PD} = [4000; 100]$, $K_{PD} = [2000; 70]$, $K_{PD} = [1000; 50]$, $K_{PD} = [500; 50]$ (see Fig. 4.14).

(b) Posing the robot on the ground while carrying $68\ lb$ of unknown load: compare L1 adaptive control with low-gain PD ($[200; 20]$) and PD with different gains $K_{PD} = [4000; 100]$, $K_{PD} = [2000; 70]$, $K_{PD} = [1000; 50]$ (see Fig. 4.15).

(c) Posing the robot on the ground while carrying $68\ lb$ of unknown load: apply L1 adaptive control with low-gain PD ($[200; 20]$) to lift the weigh up and down by tracking different set points of the knee angles (see Fig. 4.16).

(d) Walking with different the original and the new floor and feet: compare L1 adaptive control with low-gain PD ($[500\ 50]$) and PD control with $K_{PD} = [4000\ 100; 3000\ 70; 6000\ 200; 5000\ 150]$ (see Fig. 4.17).

In the first experiment, we pose the robot on the ground (no carrying load for this case) and compare the performance of L1 adaptive control with low-gain PD ($[200; 20]$) and PD with different gains. We switch between different controllers with the following order: PD with $K_{PD} = [4000; 100]$ => L1 => PD with $K_{PD} = [2000; 70]$ => L1 => PD with $K_{PD} = [1000; 50]$ => L1 => PD with $K_{PD} = [500; 50]$. The tracking error of the right knee angle is presented in Fig. 4.14. Figure (a) shows that PD with $K_{PD} = [500; 50]$ fails to hold the robot (the knee angle keeps increasing at the end of the experiment). The snapshot of this experiment is also shown in Fig. 4.13b. In Figure (b), we zoom in to ignore the last experiment when the robot falls to better compare the tracking errors.

We then conduct the experiment of posing the robot on the ground while carrying an unknown load of $68\ lb$ (see Fig. 4.13a). For both this experiment (with $68\ lb$ of load) and the above experiment (no load), the L_1 adaptive control outperform the PD control with very small tracking errors at all joints (about $0.01\ deg$ or $0.0017\ rad$). Note that we use a very low PD gain of $[200; 20]$ and the adaptation law of L_1 adaptive control play the role to bring down the tracking error. Fig. 4.13 shows clearly the comparison between the two controllers: the L_1 adaptive controller with low PD gain of $[200; 20]$ can hold a heavy load of $68\ lb$ (or $31\ kg$) with almost zero errors at all joints; the PD controller with $K_{PD} = [500; 50]$ fails to hold the robot even without any load.

We also implemented the experiment of lifting the heavy load with L_1 adaptive control by reducing the desired knee angles. Fig. 4.16 shows the right knee angle output tracking and control input.

We then conducted walking experiment to test the performance of L_1 adaptive control with time-varying desired trajectory and also to compare its performance with the PD controller for this case. For walking, since we need to deal with time-varying desired trajectory, we need to increase the PD gain to make sure that the reference model has good enough tracking errors. We use the PD gain of $[500\ 50]$ for walking with L_1 adaptive control, which is still relatively low in comparison with the gain used for the PD controller ($K_P = [4000; 3000; 6000; 5000], K_D = [100; 70; 200; 150]$). Note that this PD gain was hand tuned to get the best performance possible for our walking experiment,therefore there are different gains for each of 4 joints. We also use this gain for other walking experiments in this paper. Fig. 4.17 compares the performance of the two controllers with the original and new floor and feet. With the original floor and feet, both PD and L_1 adaptive control have good walking. Since, the L_1 adaptive control use a small gain of $[500; 50]$, it introduces compliance to the walking and therefore has smaller GRF profile and smother walking. Interestingly, thanks to this advantage, the L_1 adaptive control can adapt to the new floor and feet to perform a good walking, while the PD controller fails after the first two steps due to hard impacts and foot slipping.

4.3 Summary

This Chapter presented adverse effect analysis of model uncertainty on the control Lyapunov functions based controller and then offered two solutions for improve the robustness of the system. The first one is robust CLF-QP that handles uncertainty by expressing the problem via quadratic constraints and therefore can be able develop to robust safety and robust constraints later in Chapter. 8. Although we get some impressive results using the proposed robust CLF-QP controller, there is still an disadvantage that it is unnecessarily aggressive even without model uncertainty due to its worst-case scenario assumption. We then introduce a second solution using L1 adaptive control. This method compare the difference between the real system state and that from the reference model to estimate model uncertainty and then take control action in order to adapt to that. The L1 adaptive control based method therefore performs similarly as the baseline CLF-QP controller if there is no model uncertainty.

Chapter 5

Safety-Critical Control via Control Barrier Functions

In this chapter, we will present the method using control barrier functions (CBF) to guarantee sate-based constraints in dynamical systems as means to keep the system state to stay within a safety set. For this method, we firstly present a barrier function associated with a safety set which can be represented by a state-dependent constraint. The existence of a barrier function satisfying a CBF condition will imply forward invariance of the set. This formulation results in an inequality constraint in the control input that can be enforced via a quadratic program. However, with the nominal CBFs, if the system states are close to the safety set or happen to escape from it due to model uncertainty, the controller will require high control inputs or become infeasible. The nominal CBFs are also difficult to be extended to high relative-degree systems, for example mechanical systems with springs or series elastic actuators that are often seen in many dynamic walking robots. In order to address these disadvantages of the current method, we introduce Exponential Control Barrier Functions (ECBFs) as means to enforce high relative-degree safety constraints for nonlinear systems. We also develop a systematic design method that enables creating the Exponential CBFs for non-linear systems making use of tools from linear control theory. Our method creates a smooth boundary for the safety set via an exponential function, therefore is called Exponential CBFs. Similar to exponential stability and linear control, the exponential boundary of our proposed method helps to have smoother control inputs and guarantee the robustness under model uncertainty. The proposed control design is numerically validated on a relative-degree 4 nonlinear system (the two-link pendulum with elastic actuators and experimentally validated on a relative-degree 6 linear system (the serial cart-spring system). Thanks to these advantages of Exponential CBFs, we then can apply the method to the problem of 3D dynamic walking with varied step length and step width as well as dynamic walking on time-varying stepping stones that will be presented in the next Chapter.

5.1 Control Barrier Function based Quadratic Programs Revisited

We begin with the control affine system (3.6) with the goal to design a controller to keep the state x in the safe set

$$\mathcal{C} = \{x \in \mathbb{R}^n : h(x) \geq 0\}, \tag{5.1}$$

where $h : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function. Then a function $B : \mathcal{C} \to \mathbb{R}$ is a Control Barrier Function (CBF) [3] if there exists class $\mathcal{K}$ function α_1 and α_2 such that, for all $x \in Int(\mathcal{C}) = \{x \in \mathbb{R}^n : h(x) > 0\}$,

$$\frac{1}{\alpha_1(h(x))} \leq B(x) \leq \frac{1}{\alpha_2(h(x))}, \tag{5.2}$$

$$\dot{B}(x,u) \leq \frac{\gamma}{B(x)}, \tag{5.3}$$

where

$$\dot{B}(x,u) = \frac{\partial B}{\partial x}\dot{x} = L_f B(x) + L_g B(x) u, \tag{5.4}$$

with the Lie derivatives computed as,

$$L_f B(x) = \frac{\partial B}{\partial x} f(x), \qquad L_g B(x) = \frac{\partial B}{\partial x} g(x). \tag{5.5}$$

Thus, if there exists a Barrier function $B(x)$ that satisfies the CBF condition in (5.3), then $\mathcal{C}$ is forward invariant, or in other words, if $x(0) = x_0 \in \mathcal{C}$, i.e., $h(x_0) \geq 0$, then $x = x(t) \in \mathcal{C}, \forall t$, i.e., $h(x(t)) \geq 0, \forall t$. Note that, as mentioned in [3], this notion of a CBF is stricter than standard notions of CBFs in prior literature that only require $\dot{B} \leq 0$.

In this work, we will use the following reciprocal control Barrier candidate function:

$$B(x) = \frac{1}{h(x)}. \tag{5.6}$$

Incorporating the CBF condition (5.3) into the CLF-QP, we have the following CBF-CLF-QP based controllers, both without and with constraints respectively:

CBF-CLF-QP:

$$
\begin{aligned}
u^*(x) = \underset{u,\mu,\delta}{\operatorname{argmin}} \quad & \mu^T\mu + p\,\delta^2 && (5.7)\\
\text{s.t.} \quad & \dot{V}_\varepsilon(\eta,\mu) + \frac{c_3}{\varepsilon}V_\varepsilon(\eta) \le \delta, && \textbf{(CLF)}\\
& \dot{B}(x,u) - \frac{\gamma}{B(x)} \le 0, && \textbf{(CBF)}\\
& u = u_{ff}(x) + (L_gL_fy(x))^{-1}\mu, && \textbf{(IO)}
\end{aligned}
$$

CBF-CLF-QP with Constraints:

$$
\begin{aligned}
u^*(x) = \underset{u,\mu,\delta}{\operatorname{argmin}} \quad & \mu^T\mu + p\,\delta^2 && (5.8)\\
\text{s.t.} \quad & \dot{V}_\varepsilon(\eta,\mu) + \frac{c_3}{\varepsilon}V_\varepsilon(\eta) \le \delta, && \textbf{(CLF)}\\
& \dot{B}(x,u) - \frac{\gamma}{B(x)} \le 0, && \textbf{(CBF)}\\
& A_c(x)u \le b_c(x), && \textbf{(Constraints)}\\
& u = u_{ff}(x) + (L_gL_fy(x))^{-1}\mu. && \textbf{(IO)}
\end{aligned}
$$

Note that, as illustrated in Appendix A.2, the CBF inequality condition in the above CBF-CLF-QPs is affine in u, ensuring that these are actually quadratic programs.

5.2 Exponential Control Barrier Function

Having revisited control Barrier function based quadratic programs, in this section, we introduce "Exponential Control Barrier Functions" (ECBFs) and present a novel method to systematically design a exponential CBF for high relative-degree constraints. The term Exponential CBF is used since the resulting CBF constraint is an exponential function of the initial condition. Furthermore, the design and enforcement of ECBFs is based on linear control theory, and as a result, we can easily take advantage of conventional linear control design techniques such as pole placement.

Definition 5.1. *(*Exponential Control Barrier Function*): Consider the dynamical system* (3.6) *and the set* $\mathcal{C}_0 = \{x \in \mathbb{R}^n \mid B(x) \ge 0\}$*, where* $B : \mathbb{R}^n \to \mathbb{R}$ *has relative-degree* r_b*.* $B(x)$ *is an exponential control*

Barrier function (ECBF) if there exists $K_b \in \mathbb{R}^{r_b \times 1}$ s.t.,

$$\inf_{u \in U} [L_f^{r_b} B(x) + L_g L_f^{r_b - 1} B(x) u + K_b \eta_b(x)] \geq 0, \forall x \in \mathcal{C}_0, \tag{5.9}$$

and $B(x(t)) \geq C_b e^{A_b t} \eta_b(x_0) \geq 0$, when $B(x_0) \geq 0$, where, the matrix A_b is dependent on the choice of K_b, and

$$\eta_b(x) = \begin{bmatrix} B(x) \\ \dot{B}(x) \\ \ddot{B}(x) \\ \vdots \\ B^{(r_b-1)}(x) \end{bmatrix} = \begin{bmatrix} B(x) \\ L_f B(x) \\ L_f^2 B(x) \\ \vdots \\ L_f^{r_b - 1} B(x) \end{bmatrix}, \tag{5.10}$$

$$C_b = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}. \tag{5.11}$$

Remark 5.1. *Note that K_b, and consequently A_b, need to satisfy certain properties. We will develop these properties later in this section.*

For the simple case of relative-degree 1 ECBFs, the above definition can be reformulated as the following:

Definition 5.2. *(*Relative degree 1 Exponential Control Barrier Function*): Consider the dynamical system* (3.6) *and the set $\mathcal{C}_0 = \{x \in \mathbb{R}^n \mid B(x) \geq 0\}$, where $B : \mathbb{R}^n \to \mathbb{R}$ has relative-degree 1 and is continuously differentiable. $B(x)$ is an exponential control Barrier function (ECBF) with relative-degree 1 if there exists $k_b \in \mathbb{R}$ s.t.,*

$$\inf_{u \in U} [L_f B(x) + L_g B(x) u + k_b B(x)] \geq 0, \forall x \in \mathcal{C}_0, \tag{5.12}$$

and $B(x(t)) \geq B(x_0) e^{-k_b t} \geq 0$, when $B(x_0) \geq 0$.

Remark 5.2. *Note that with relative-degree 1, K_b and A_b in Definition 5.1 become scalar k_b, a_b and the condition for $B(x)$ to be an Exponential CBF is $a_b = k_b > 0$.*

Remark 5.3. *(*Relation between Exponential CBF and Zeroing CBF*): The relative-degree 1 ECBF is a Zeroing CBF (ZCBF), as defined in [84] since $k_b \circ B(x)$ is a class $\mathcal{K}$ function of $B(x)$, and thus retains all the robustness properties of the ZCBF, as detailed in [84, Sec. 2.2].*

We will next introduce the notion of Virtual Input-Output Linearization followed by the design of an Exponential CBF.

5.2.1 Virtual Input-Output Linearization

As mentioned in Section 3.4, CLFs are an effective tool to handle stability for both linear and nonlinear systems. Furthermore, there is a systematic way to design CLFs for regulating outputs with arbitrary relative-degree $r \geq 1$. If we can derive the CBF to the same form as a CLF, by using another Input-Ouput linearization for the CBF (5.3), we can then develop a general CBF for constraints with arbitrary relative-degree $r_b \geq 1$. However, input-output linearizing $\dot{B}(x)$ is not directly feasible due to: (a) the decoupling matrix ($L_g B(x)$ when $r_b = 1$) being a vector and obviously not invertible, and (b) the control input u in (3.9) being already used to Input-Output linearize the output dynamics resulting in (3.13).

In order to solve this problem, we introduce the notion of Virtual Input-Ouput Linearization (VIOL) where an invertible decoupling matrix is not required and the control input μ is chosen to satisfy both the CLF condition (3.19) as well as input-output linearize the Barrier dynamics as follows. For a CBF with relative-degree 1, let's define a virtual control input μ_b as follows:

$$\dot{B}(x, \mu) = L_f B(x) + L_g B(x) u(x, \mu) =: \mu_b(x, \mu). \tag{5.13}$$

Note that μ_b is a scalar. The CBF condition (5.3) then simply becomes:

$$\mu_b(x, \mu) \leq \frac{\gamma}{B(x)}. \tag{5.14}$$

We can then let a QP compute μ, μ_b so as to simultaneously satisfy the CLF condition (3.19) as well as both (5.13) and (5.14):

CBF-CLF-QP:

$$\begin{aligned}
\mu^* = \underset{\mu, \mu_b, d_1}{\operatorname{argmin}} \quad & \mu^T \mu + p_1 \, d_1^2 && (5.15)\\
\text{s.t.} \quad & \dot{V}(\eta, \mu) + \lambda V(\eta) \leq d_1 && \textbf{(CLF)}\\
& \mu_b - \frac{\gamma}{B(x)} \leq 0 && \textbf{(CBF)}\\
& A_c(x)\mu \leq b_c(x) && \textbf{(Constraints)}\\
& \dot{B}(x, \mu) = \mu_b. && \textbf{(VIOL)}
\end{aligned}$$

Remark 5.4. *Note that the solutions of the two controllers* (5.15) *and* (5.7) *are identical. However, the*

VIOL in the above CBF-CLF-QP opens up an effective and systematic way of designing the exponential CBFs.

5.2.2 Designing Exponential Control Barrier Functions

Consider the closed set $\mathcal{C}_0 = \{x \in \mathbb{R}^n \mid B(x) \geq 0\}$, with our control goal being to find an input u that guarantees forward invariance of $\mathcal{C}_0$, i.e., if $x_0 := x(0) \in \mathcal{C}_0$ then find u s.t., $B(x(t)) \geq 0, \forall t \geq 0$.

Simple Case

We first consider the problem of $B(x)$ having relative-degree 1. Using VIOL, we have,

$$\dot{B}(x, \mu) = \mu_b. \tag{5.16}$$

If we want to drive $B(x)$ to zero, we can simply apply the control

$$\begin{aligned} \mu_b &= -k_b B(x),\ k_b > 0 \qquad (5.17)\\ \implies B(x(t)) &= B(x_0)e^{-k_b t} \to 0 \text{ as } t \to \infty. \end{aligned}$$

Making use of this property, we can guarantee the inequality constraint $B(x) \geq 0$ by imposing:

$$\begin{aligned} \mu_b &\geq -k_b B(x),\ k_b > 0 \qquad (5.18)\\ \implies \dot{B}(x, \mu) &\geq -k_b B(x) \qquad (5.19)\\ \implies B(x(t)) &\geq B(x_0)e^{-k_b t} \geq 0 \text{ when } B(x_0) \geq 0. \qquad (5.20) \end{aligned}$$

Now, we can develop this approach for the general case with $B(x)$ having relative-degree $r_b \geq 1$.

General Case

Let $r_b \geq 1$ be the relative-degree of $B(x)$. We have,

$$\begin{aligned} B^{(r_b)}(x, \mu) &= L_f^{r_b} B + L_g L_f^{r_b - 1} B\, u(x, \mu)\\ &= L_f^{r_b} B + L_g L_f^{r_b - 1} B \left(u^*(x) + (L_g L_f^{r-1})^{-1} h(x) \mu \right), \end{aligned} \tag{5.21}$$

where we substitued for $u(x, \mu)$ from (3.9). Further, using VIOL we have

$$B^{(r_b)}(x, \mu) = \mu_b, \tag{5.22}$$

such that the Input-Output linearized system becomes

$$\begin{cases} \dot{\eta}_b(x) & = F_b \eta_b(x) + G_b \mu_b \\ B(x) & = C_b \eta_b(x), \end{cases} \tag{5.23}$$

where $\eta_b(x)$ is as defined in (5.10), $F_b \in \mathbb{R}^{r_b \times r_b}, G_b \in \mathbb{R}^{r_b \times 1}$ are as defined below

$$F_b = \begin{bmatrix} 0 & 1 & . & . & . & 0 \\ 0 & 0 & 1 & . & . & 0 \\ . & . & . & & & \\ 0 & . & . & . & . & 1 \\ 0 & . & . & . & . & 0 \end{bmatrix}, \quad G_b = \begin{bmatrix} 0 \\ . \\ . \\ . \\ 1 \end{bmatrix}, \tag{5.24}$$

and C_b is as defined in (5.11).

From this controllable canonical form, if we want to drive $B(x)$ to zero, we can easily find a pole placement controller $\mu_b = -K_b \eta_b$ with all negative real poles $p_b = -\begin{bmatrix} p_1 & p_2 & ... & p_{r_b} \end{bmatrix}$, with $p_i > 0,\ i = 1, \cdots, r_b$, that obtains the closed loop matrix $A_b = F_b - G_b K_b$ with all negative real eigenvalues.

Motivated by (5.17), we can then apply,

$$\mu_b \geq -K_b \eta_b, \tag{5.25}$$

$$\implies \dot{\eta}_b \geq A_b \eta_b. \tag{5.26}$$

Assuming $K_b = \begin{bmatrix} k_b^1 & k_b^2 & \cdots & k_b^{r_b - 1} \end{bmatrix}$ and from the definition of η_b in (5.10), the last row of the above vector inequality (5.26) results in

$$B^{(r_b)}(x) \geq k_b^1 B(x) + k_b^2 \dot{B}(x) + \cdots + k_b^{r_b - 1} B^{(r_b - 1)}(x). \tag{5.27}$$

This inequality constraint can also be written in terms of the pole locations p_i. To do this, we first define a family of outputs $y_i : \mathbb{R}^n \to \mathbb{R}$ for $i = 1, \cdots, r_b$, as follows,

$$y_i(x) = (\frac{d}{dt} + p_1) \circ (\frac{d}{dt} + p_2) \circ \cdots \circ (\frac{d}{dt} + p_{r_b}) \circ B(x), \tag{5.28}$$

with $y_0(x) := B(x)$. Associated with these outputs, we define a family of closed sets for $i = 0, \cdots, r_b$, as

$$\mathcal{C}_i = \{x \in \mathbb{R}^n \mid y_i(x) \geq 0\}. \tag{5.29}$$

Remark 5.5. *Note that from the definition in* (5.28)*, the output functions can also be written recursively as,*

$$y_i(x) = \dot{y}_{i-1}(x) + p_i \, y_{i-1}(x). \tag{5.30}$$

Remark 5.6. *Note that because of choice of K_b resulting in poles p_b,* (5.27) *is identical to $y_{r_b}(x) \geq 0$.*

Theorem 5.1. *Suppose the closed-set $\mathcal{C}_{r_b}$ is forward invariant for the system* (3.6)*, then the closed-set $\mathcal{C}_0$ is forward invariant whenever initially $x_0 \in \mathcal{C}_i$ for each $i = 0, \cdots, r_b$, and $p_i > 0$ for each $i = 1, \cdots, r_b$.*

Before we prove Theorem 5.1, we note the following corollaries:

Corollary 5.1. *Suppose $y_i(x_0) \geq 0$ and $p_i > 0$ for $i = 1, \cdots, r_b$, then $B(x(t)) \geq 0, \forall t > 0$ when $B(x_0) \geq 0$.*

Proof. This follows directly from Theorem 5.1 and the definition of the family of outputs y_i in (5.28) and closed sets $\mathcal{C}_i$ in (5.29). □

Corollary 5.2. *Suppose $p_i \geq \max(-\frac{\dot{y}_{i-1}(x_0)}{y_{i-1}(x_0)}, \delta)$, $\delta > 0$, for $i = 1, \cdots, r_b$, then $B(x(t)) \geq 0, \forall t > 0$ when $B(x_0) \geq 0$.*

Proof. This follows from Corollary 5.1 and the fact from (5.30) that

$$y_i(x_0) \geq 0 \Leftrightarrow p_i \geq -\frac{\dot{y}_{i-1}(x_0)}{y_{i-1}(x_0)}.$$

□

Proposition 5.1. *Suppose the closed-set $\mathcal{C}_i$ is forward invariant for the system* (3.6)*, then the closed-set $\mathcal{C}_{i-1}$ is forward invariant whenever initially $x_0 \in \mathcal{C}_i$, $x_0 \in \mathcal{C}_{i-1}$ and $p_i > 0$.*

Proof. The proof essentially follows from [82, Prop. 1]. Since $x_0 \in \mathcal{C}_i$, then by forward invariance of $\mathcal{C}_i$, we have $x(t) \in \mathcal{C}_i$, for $t \in [0, \infty)$. From the definition of $\mathcal{C}_i$ and (5.30), this is equivalent to

$$\dot{y}_{i-1}(x(t)) + p_i \, y_{i-1}(x(t)) \geq 0, \;\; \forall t \in [0, \infty).$$

Since the trajectory starts in $\mathcal{C}_{i-1}$, consider the extreme case when the system trajectory reaches the boundary of $\mathcal{C}_{i-1}$ at time T. Then, $y_{i-1}(x(T)) = 0$. However, according to the previous inequality, it follows that

$$\dot{y}_{i-1}(x(T)) \geq 0,$$

which means that the trajectory would never escape $\mathcal{C}_{i-1}$. □

We are now ready to prove Theorem 5.1.

Proof. (of Theorem 5.1:) The result follows from recursive application of Proposition 5.1. In particular, suppose $\mathcal{C}_{r_b}$ is forward invariant and $x_0 \in \mathcal{C}_{r_b}$, $x_0 \in \mathcal{C}_{r_b-1}, p_{r_b} > 0$, then from Proposition 5.1, $\mathcal{C}_{r_b-1}$ is forward invariant. Further if $x_0 \in \mathcal{C}_{r_b-2}$ and $p_{r_b-1} > 0$, then from Proposition 5.1, $\mathcal{C}_{r_b-2}$ is forward invariant. We can continue applying Proposition 5.1 so on to show $\mathcal{C}_0$ is forward invariant. □

Theorem 5.2. *(*Main Result*): Suppose K_b is chosen s.t. A_b as defined in* (5.11) *is Hurwitz and total negative, and moreover $-\lambda_i(A_b) \geq -\frac{\dot{y}_{i-1}(x_0)}{y_{i-1}(x_0)}$. Then $\mu_b \geq -K_b\eta_b(x)$ with $\eta_b(x)$ as in* (5.10)*, guarantees $B(x)$ is a Exponential CBF.*

Proof. The choice of μ_b establishes the invariance of C_{r_b}, and K_b being Hurwitz and total negative ensures that the eigenvalues of A_b are real and negative. The rest follows from Corollary 5.2 and Theorem 5.1. □

In summary, if $B(x_0) \geq 0$ and the designed poles are chosen sufficiently small so that the condition in Corollary 5.2 holds, we can guarantee the state-dependent constraint $B(x) \geq 0$ by applying the exponential CBF condition (5.25).

Remark 5.7. *(*Relation between Exponential CBF and Modified CBF with position-based constraints [82]*): For safety constraints with relative-degree 2, the modification of CBF presented in [82] extends the CBF condition for position-based (relative-degree 2) constraints of the form $g(x) \geq 0$ by enforcing the standard CBF condition [3] on $h(x) = (\frac{d}{dt} + \gamma_b) \circ g(x) \geq 0, \gamma_b > 0$. Since, $h(x) \geq 0 \Leftrightarrow \dot{g}(x) \geq -\gamma_b g(x)$, enforcing the modified CBF $h(x) \geq 0$ results in $g(x) \geq 0$. $g(x)$ can then be seen as a relative-degree 2 Exponential CBF.*

We now present a QP that incorporates an Exponential CBF into a CLF-QP:

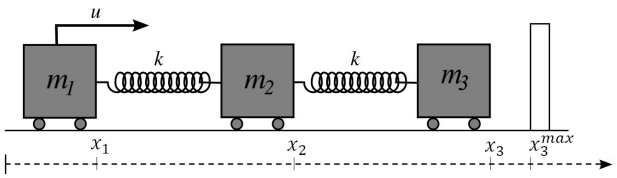


Figure 5.1: The Serial Spring Mass System (Relative degree 6). The control goal is to drive the system state from the initial condition x_0 to a desired position for the 3^{rd} cart, x_{3d}, while strictly enforcing the safety constraint $x_3 \leq x_3^{max}$. This system has three degrees-of-freedom and two degrees-of-underactuation.

ECBF-CLF-QP:

$$\begin{aligned}
\underset{\mu,\delta}{\text{argmin}} \quad & \mu^T\mu + p\delta^2 && (5.31)\\
\text{s.t.} \quad & \dot{V}(\eta,\mu) + \lambda V(\eta) \leq \delta && \textbf{(CLF)}\\
& A_C^{\mu}(x)\mu \leq b_C^{\mu}(x) && \textbf{(Constraints)}\\
& \mu_b \geq -K_b\eta_b && \textbf{(Exponential CBF)}\\
& B^{(r_b)}(x,\mu) = \mu_b && \textbf{(VIOL)}
\end{aligned}$$

In the next Section, we will validate our proposed method through two systems: serial spring mass system (linear system with relative-degree 6) and a two-link pendulum with elastic actuators (nonlinear system with relative 4).

5.2.3 Numerical Validation of Exponential CBF

Serial Spring Mass System (Relative degree 6)

We validate our proposed method on a simple system comprising of serial masses connected through springs as shown in Fig. 5.1. The equations of motion for the system is as follows:

$$m_1\ddot{x}_1 = u + k(x_2 - x_1) \tag{5.32}$$

$$m_2\ddot{x}_2 = k(x_1 - x_2) + k(x_3 - x_2) \tag{5.33}$$

$$m_3\ddot{x}_3 = k(x_2 - x_3) \tag{5.34}$$

Defining the system state $x = [x_1\ x_2\ x_3\ \dot{x}_1\ \dot{x}_2\ \dot{x}_3]^T$, we have the linear system: $\dot{x} = Ax + Bu$, where,

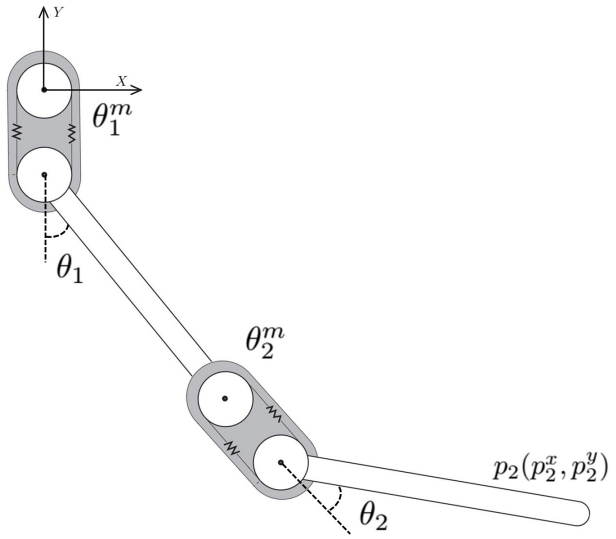


Figure 5.2: The 2-link Pendulum with Elastic Actuators (Relative degree 4). The control goal is to drive the link angles from an initial configuration $\theta_1(0), \theta_2(0)$ to a desired configuration θ_{1d}, θ_{2d} while strictly enforcing the safety constraint on the vertical position of the end-effector, $p_2^y \geq p_{2min}$. This system is nonlinear with four degrees-of-freedom and two degrees-of-underactuation.

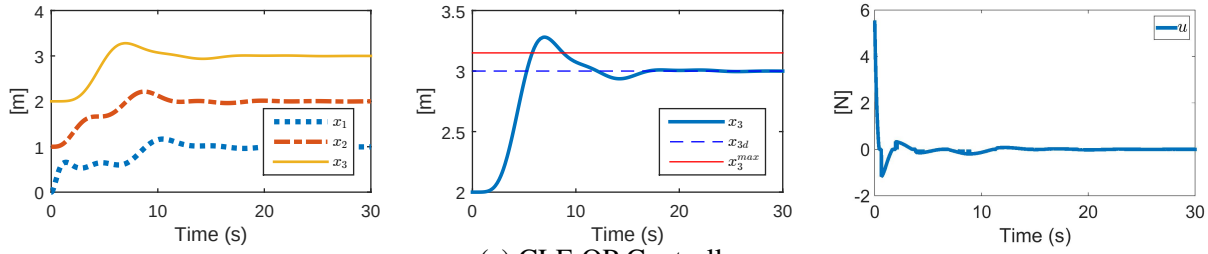
$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -\frac{k}{m_1} & \frac{k}{m_1} & 0 & 0 & 0 & 0 \\ \frac{k}{m_2} & -\frac{2k}{m_2} & \frac{k}{m_2} & 0 & 0 & 0 \\ 0 & \frac{k}{m_3} & -\frac{k}{m_3} & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{m_1} \\ 0 \\ 0 \end{bmatrix} \tag{5.35}$$

Our control goal is to drive the system state from the initial condition x_0 to the desired position x_{3d} while considering the safety constraint $x_3 \leq x_3^{max}$.

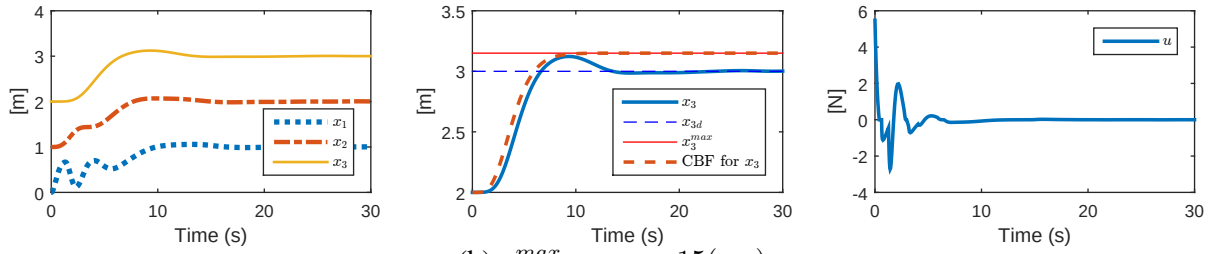
As illustrated in Fig.5.3, the CLF-QP controller violates the constraint with $max(x_3) = 3.27(m) > x_3^{max} = 3.15(m)$, the ECBF-CLF-QP controller with the desired poles $p_b = -0.12\times\begin{bmatrix}10 & 11 & 12 & 13 & 14 & 15\end{bmatrix}$ can handle different safety constraints $x_3 \leq x_3^{max}$.

2-Link Pendulum with Elastic Actuators (Relative degree 4)

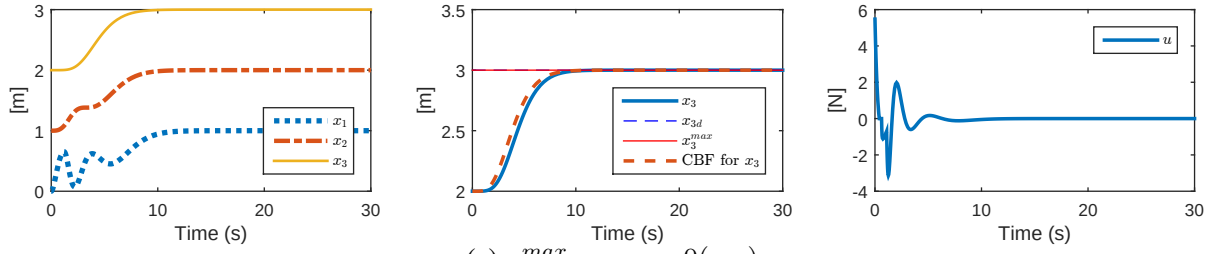
We consider a two-link pendulum with elastic actuators, as shown in Fig.5.2. The two-link pendulum is a nonlinear system. With elastic actuators, the commanded torques τ_1, τ_2 of the two motors will generate



(a) CLF-QP Controller



(b) $x_3^{max} - x_{3d} = 15(cm)$



(c) $x_3^{max} - x_{3d} = 0(cm)$

Figure 5.3: Serial Spring Mass System (relative-degree 6): Cart positions and input force for the Exponential CBF-CLF-QP controller with desired cart position $x_{3d} = 3(m)$ and with safety constraint $x_3 \leq x_3^{max}$. The pole locations p_b for the CBF encode the performance specifications of enforcing a safety constraint. As seen above, varying the safety constraint while keeping the poles fixed keeps the peak forces and speed of system response the same. Simulation video: `http://youtu.be/okojFUtaiDk`.

torques at two joints indirectly through the following motor dynamics:

$$\begin{aligned} J_m\ddot{\theta}_1^m &= k(\theta_1 - \theta_1^m) + \tau_1, \\ J_m\ddot{\theta}_2^m &= k(\theta_2 - \theta_2^m) + \tau_2, \end{aligned} \tag{5.36}$$

where θ_1, θ_2 are joint angles of the robot, θ_1^m, θ_2^m are angles of two motors, J_m, k are inertia and stiffness of motors. Then, the torque at two joints of the robot would be:

$$\begin{aligned} u_1 &= -k(\theta_1 - \theta_1^m) - \xi\dot{\theta}_1, \\ u_2 &= -k(\theta_2 - \theta_2^m) - \xi\dot{\theta}_2, \end{aligned} \tag{5.37}$$

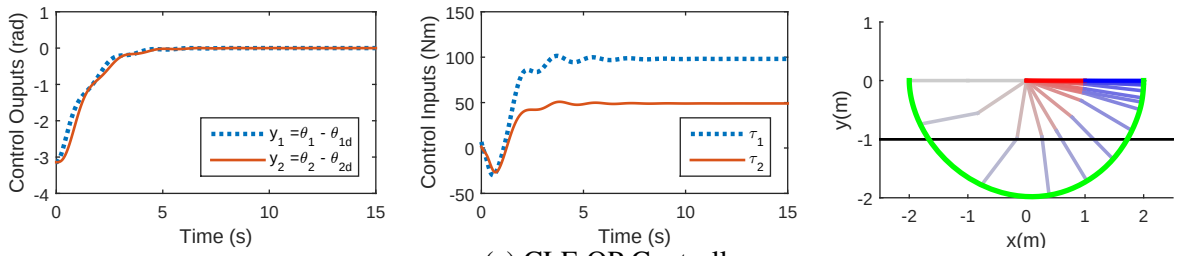
where ξ is the damping coefficient at the joints.

We apply the Exponential CBF-CLF-QP controller on the two-link pendulum nonlinear system with the above motor dynamics to enforce a constraint on the position of the end effector $p_2(p_2^x, p_2^y)$ (see Fig.5.4). While the nominal CLF-QP controller violates the constraint, the ECBF-CLF-QP with poles $p_b = -\begin{bmatrix} 5 & 5.5 & 6 & 10 \end{bmatrix}$, can handle different safety constraints.

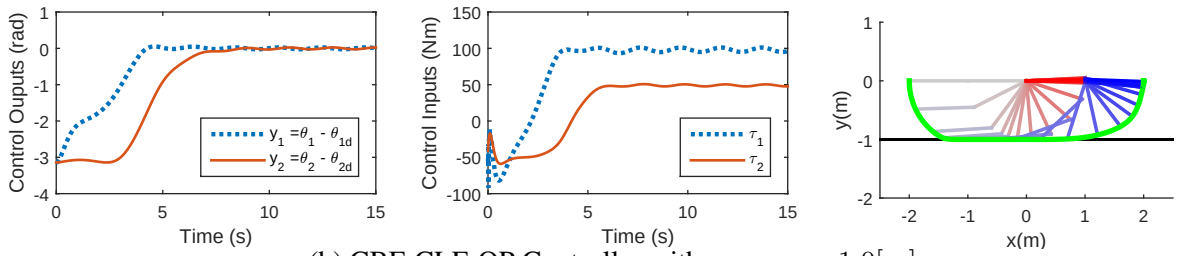
Exponential CBFs have the advantage that they can be easily designed for high relative degree safety constraints using tools from linear control theory. The pole locations for the designed CBF and the poles used to design the CLF can be chosen to more smoothly tradeoff stability of tracking and enforcement of safety. Despite these advantages, Exponential Control Barrier Function have some limitations. The choice of pole location depends on initial conditions as stated in Corollary 5.2, requiring careful choice of these poles. Although, if the initial conditions are bounded, the poles can be chosen based on these bounds. Furthermore, the presented Exponential CBF-based control design is dependent on the system model and could be sensitive to model uncertainty. Preliminary results to address safety constraints with model uncertainty are presented in [60].

5.3 Summary

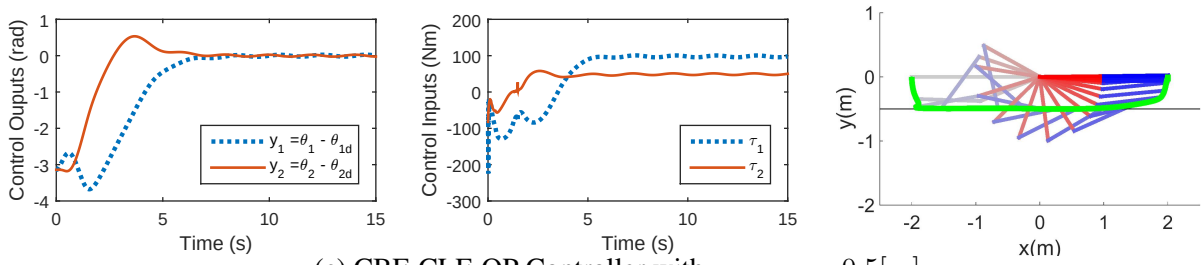
This Chapter has introduced Exponential Control Barrier Functions (ECBFs) as means to enforce high relative degree safety constraints for nonlinear systems. We have presented a systematic design method that enables creating the Exponential CBFs based on pole placement. The designed exponential CBFs along with control Lyapunov functions (CLFs) were formulated as a unified quadratic program. The proposed control design has been numerically validated on a relative degree 6 linear system (the serial cart-spring system) and on a relative degree 4 nonlinear system (the two-link pendulum with elastic actuators.)



(a) CLF-QP Controller



(b) CBF-CLF-QP Controller with $p_{2min} = -1.0[m]$.



(c) CBF-CLF-QP Controller with $p_{2min} = -0.5[m]$.

Figure 5.4: 2-link Pendulum with Elastic Actuators (nonlinear system with relative-degree 4): The control goal is to drive the link angles from an initial configuration $\theta_1(0) = -\pi, \theta_2(0) = 0$ to a desired configuration $\theta_{1d} = \pi, \theta_{2d} = 0$ while strictly enforcing the safety constraint on the vertical position of the end-effector, $p_2^y \geq p_{2min}$. Note that the links have unit length and the controller has to aggressively move the links to enforce the strict safety constraint. Simulation video: `http://youtu.be/okojFUtaiDk`.

Chapter 6

Dynamic Walking on Stepping Stones with Control Barrier Functions

In this chapter, we will present the method of using control barrier functions (CBF) for the problem of dynamic bipedal robotic walking on stepping stones where the footstep placement is maintained as a safety-critical constraint. To be more specific, we represent footstep placement constraint as state-based constraints that guarantee the footstep location to stay within a desired range when the swing foot has impact with the ground. In the QP based controller, we enforce the CBF constraints that means to keep those state-based constraints to be strictly satisfied, resulting in the desired footstep placement at impact. For the problem of planar dynamic walking on stepping stones using only one nominal walking gait, we achieve the step length variation of ($[25 : 60](cm)$). Thanks to advantages of Exponential CBFs presented in Section. 5.2, we then can apply the method to the problem of 3D dynamic walking with varied step length and step width as well as dynamic walking on time-varying stepping stones. For the work of using CBF for stepping stones, we use only one nominal walking gait. Therefore the range of step length variation is limited ($[25 : 60](cm)$). In order to further improve the performance, we incorporate CBFs along with a library of gaits and are able to increase the step length range significantly ($[10 : 100](cm)$).

6.1 Safety-Critical Control for Dynamical Bipedal Walking

6.1.1 Modification of CBF for position based constraints

As presented in [3], the standard CBF is for velocity based constraint, i.e, $h(q, \dot{q}) \geq 0$ with relative degree one. For application of CBF to mechanical systems in general and to the bipedal robotics system in particular, we need to consider position based constraints, i.e, functions of the form $g_b(q) \geq 0$ (“b" is

for “Barrier") with relative degree two. The modification of CBF with position based constraint is first mentioned in [82] as a safety criteria for dynamical systems on manifolds. The idea is that, in order to guarantee the condition $g_b(q) \geq 0$, we construct a barrier constraint with relative degree one:

$$h_{CBF}(q, \dot{q}) = \gamma_b g_b(q) + \dot{g}_b(q, \dot{q}) \geq 0. \tag{6.1}$$

From this condition, we can guarantee that if $g_b(q)$ starts with a non-negative initial condition $g_b(q_0) \geq 0$ and the constant γ_b is strictly positive, then the condition $h_{CBF} \geq 0$ will guarantee that $g_b(q)$ will never cross zero. Because, if $g_b(q)$ crosses zero from 0^+ to 0^-, it means that:

$$g_b(q) = 0; \dot{g}_b(q, \dot{q}) < 0 \implies h_{CBF}(q, \dot{q}) = \gamma_b g_b(q) + \dot{g}_b(q, \dot{q}) < 0,$$

which violates the CBF condition in (6.1).

In other words, we can maintain $g_b(q) \geq 0$ by barrier constraint (6.1). Based on this modification, we now can apply the CBF-CLF-QP based controller (5.8) presented in Section 5.1, with the barrier function formed by $h = h_{CBF}(q, \dot{q})$ from (6.1).

We next present application of this approach for bipedal robotics walking with different additional constraints and criteria.

6.1.2 Avoiding overhead obstacles during walking

We first illustrate the method for the problem of avoiding an overhead low ceiling, denoted as (C), by limiting the head height of the robot during a walking step. The problem is that there is a low ceiling at a distance h_r from the ground at the position of the i^{th} walking step. We will formulate and apply a CBF in the i^{th} step to guarantee the head height of the robot, h_H, will be always lower than the overhead ceiling ($h_H(q) \leq h_r$) so that the robot is able to clear the low ceiling obstacle. Therefore, we just need to guarantee the following position constraint (with relative degree two) during the whole i^{th} step:

$$g_C(q) = h_r - h_H(q) \geq 0. \tag{6.2}$$

This corresponds to the head height being below the ceiling. The barrier constraint with relative degree 1:

$$h_C(q, \dot{q}) = \gamma_b g_C(q) + \dot{g}_C(q, \dot{q}) \geq 0. \tag{6.3}$$

The CBF-CLF-QP from (5.8) can then be used with the Barrier function B as defined in (5.6) with $h = h_C$ from above.

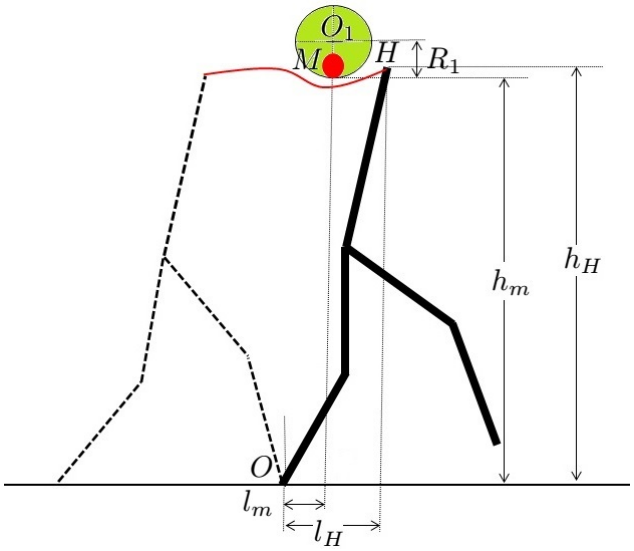


Figure 6.1: Geometric explanation of CBF constraints for the problem of avoiding an overhead obstacle located at coordinates (l_m, h_m) with respect to the stance foot. If we can guarantee the trajectory of the robot head H (the red line) to be limited outside the green domain, we can ensure the biped avoids the obstacle (small red circle).

In the above constraint, the horizontal position of the head l_H, did not matter. For a better construction for this problem, we can also consider avoiding an overhead obstacle (O) at a specific location, as presented in Fig. 6.1. Here, the red circle with center M represents the overhead obstacle located at coordinates (l_m, h_m) with respect to the stance foot, and the green circle with center O_1 represents the region that the position of the robot head should avoid to ensure no collision with the obstacle. The geometric constraints in the figure can be mathematically represented as follows:

$$O_1H = \sqrt{(R_1 + h_m - h_H)^2 + (l_H - l_m)^2} \geq R_1, \tag{6.4}$$

Then, the following position constraint,

$$g_O(q) = \sqrt{(R_1 + h_m - h_H)^2 + (l_H - l_m)^2} - R_1 \geq 0, \tag{6.5}$$

corresponds to the head being outside the green circle with center O_1 in Fig. 6.1.

The CBF-CLF-QP for this approach will totally be the same as mentioned above, but the simple constraint (6.2) now becomes more complex as (6.5).

Based on these problems of avoiding overhead obstacles with a single barrier constraint, we can now further develop a control strategy for the problem of footstep placement. One interesting application is the problem of walking over a set of discrete footholds.

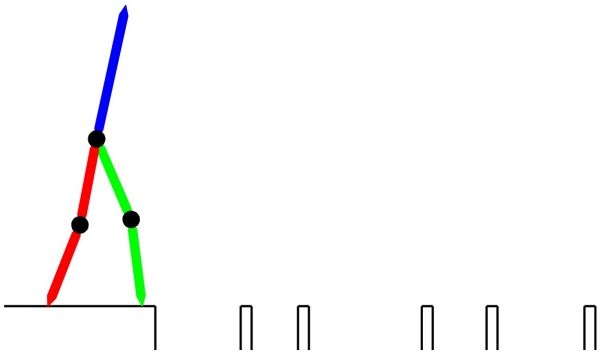


Figure 6.2: The problem of dynamically walking over a randomly generated set of discrete footholds. The discrete footholds serve as strict safety constraints that need to be enforced with formal guarantees for the safety of the bipedal robot.

6.1.3 Problem of Walking over Discrete Footholds

If we want to force the robot to step onto a specific position (see Fig. 6.2), we need to guarantee that the step length when the robot swing foot hits the ground is bounded within a given range $[l_{min}; l_{max}]$. Let $h_f(q)$ be the height of the swing foot to the ground and $l_f(q)$ be the distance between the stance and swing feet. We define the step length at impact as,

$$l_s := l_f(q)|_{h_f(q)=0, \dot{h}_f(q,\dot{q})<0}. \tag{6.6}$$

The discrete foothold constraint to be enforced then becomes,

$$l_{min} \leq l_s \leq l_{max}. \tag{6.7}$$

However, in order to guarantee this final impact-time constraint, we construct a state-based constraint for the evolution of the swing foot during the whole step, so that at impact the swing foot satisfies the discrete foothold constraint (6.14). We now offer a solution for this issue. The geometric explanation for this is presented in Fig. 6.3. If we can guarantee the trajectory of the swing foot, F, to be bounded between the domain of the two circles O_1 and O_2, it will imply that the step length when the swing foot hits the ground is bounded within $[l_{min}; l_{max}]$. Mathematical representation for these constraints is stated as follows:

$$O_1F = \sqrt{(R_1 + l_f)^2 + h_f^2} \leq R_1 + l_{max},$$
$$O_2F = \sqrt{(R_2 + h_f)^2 + (l_f - \frac{l_{min}}{2})^2} \geq \sqrt{R_2^2 + (\frac{l_{min}}{2})^2}.$$

When the swing foot hits the ground at the end of the step, $h_f = 0, \dot{h}_f < 0$, the step length is l_s, and

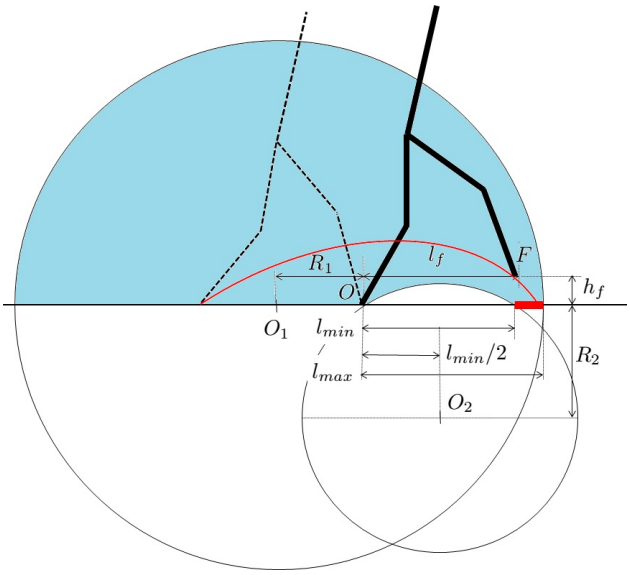


Figure 6.3: Geometric explanation of CBF constraints for the problem of bipedal walking over discrete footholds. If we can guarantee the trajectory of the swing foot F (the red line) to be limited in the blue domain, we will force our robot to step onto a discrete foothold position (thick red range on the ground). This approach therefore also provides a safety guarantee against foot scuffing or swing foot being always above the ground prior to contact.

therefore,

$$\begin{aligned} \sqrt{(R_1 + l_s)^2} &\leq R_1 + l_{max}, \\ \sqrt{R_2^2 + (l_s - \frac{l_{min}}{2})^2} &\geq \sqrt{R_2^2 + (\frac{l_{min}}{2})^2}. \end{aligned} \tag{6.8}$$

This then implies the discrete foothold constraint (6.14).

We now define the two barrier constraints based on this approach, through the position constraints

$$\begin{aligned} g_{ST1}(q) &= R_1 + l_{max} - \sqrt{(R_1 + l_f(q))^2 + h_f(q)^2} \geq 0, \\ g_{ST2}(q) &= \sqrt{(R_2 + h_f)^2 + (l_f - \frac{l_{min}}{2})^2} \\ &\quad - \sqrt{R_2^2 + (\frac{l_{min}}{2})^2} \geq 0, \end{aligned} \tag{6.9}$$

to obtain

$$\begin{aligned} h_{ST_1}(q, \dot{q}) &= \gamma_b g_{ST_1}(q) + \dot{g}_{ST_1}(q, \dot{q}) \geq 0, \\ h_{ST_2}(q, \dot{q}) &= \gamma_b g_{ST_2}(q) + \dot{g}_{ST_2}(q, \dot{q}) \geq 0. \end{aligned} \tag{6.10}$$

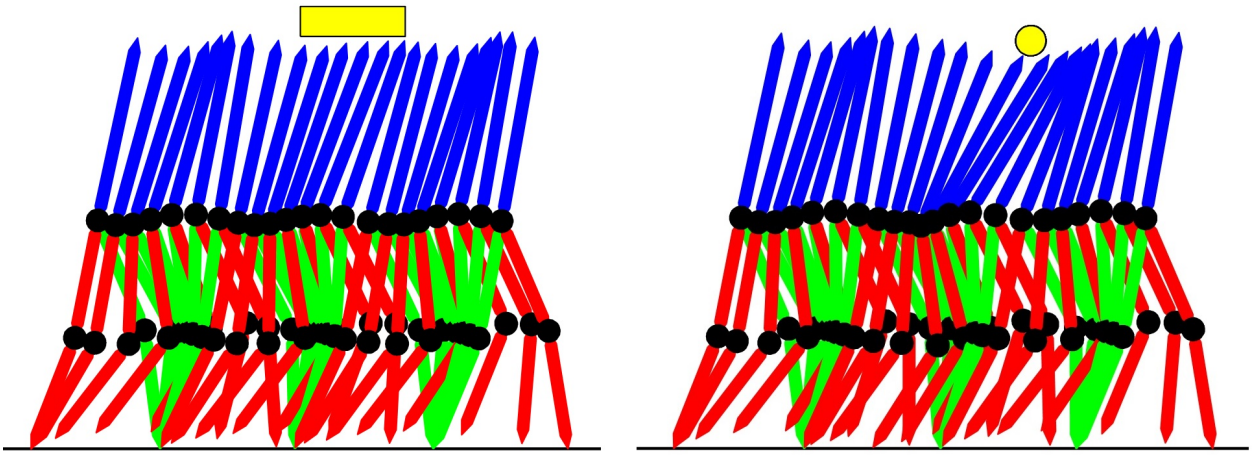


Figure 6.4: Simulation of 3 steps of RABBIT walking while avoiding overhead obstacles. The left figure is for the simpler constraint (6.2) that reduces the head height during the whole step, with the yellow rectangle representing the low ceiling. The right figure is for the more involved constraint (6.5) to avoid an overhead obstacle at a specific location, with the yellow circle representing the location of the obstacle.

The Control Barrier Candidate Functions then are

$$B_1(q,\dot{q}) = \frac{1}{h_{ST_1}(q,\dot{q})}; B_2(q,\dot{q}) = \frac{1}{h_{ST_2}(q,\dot{q})}. \tag{6.11}$$

Remark 6.1. *We note that the two problems of avoiding overhead obstacles and walking over discrete footholds, presented here, are just two examples for the application of the CBF-CLF-QP for bipedal robotic walking. With this methodology, and different design for barrier constraints, we can further apply this approach for a variety of additional constraints and criteria during walking control. We can also increase the performance by designing a better barrier constraint.*

6.1.4 Numerical Validation of Dynamic Bipedal Walking with CBF

To demonstrate the effectiveness of the proposed CLF-CBF-QP based controller, we will conduct numerical simulations on the model of RABBIT (shown in Figure 4.1), a planar five-link bipedal robot with a torso and two legs with revolute knees. As mentioned earlier in Section 4.1.4, RABBIT has four actuators to control hip and knee angles, and is connected to a rotating boom which constrains the robot to walk in a circle, approximating planar motion in the sagittal plane. Detailed descriptions of RABBIT and the associated mathematical model can be found in [15, 78]. Fundamental issues in dynamic walking and running on RABBIT can be found in [78] and [52].

Note that our model of RABBIT that we consider here is the most detailed model of the experimental system, incorporating not only the mass and inertia of the links and rotors, but also friction in the various

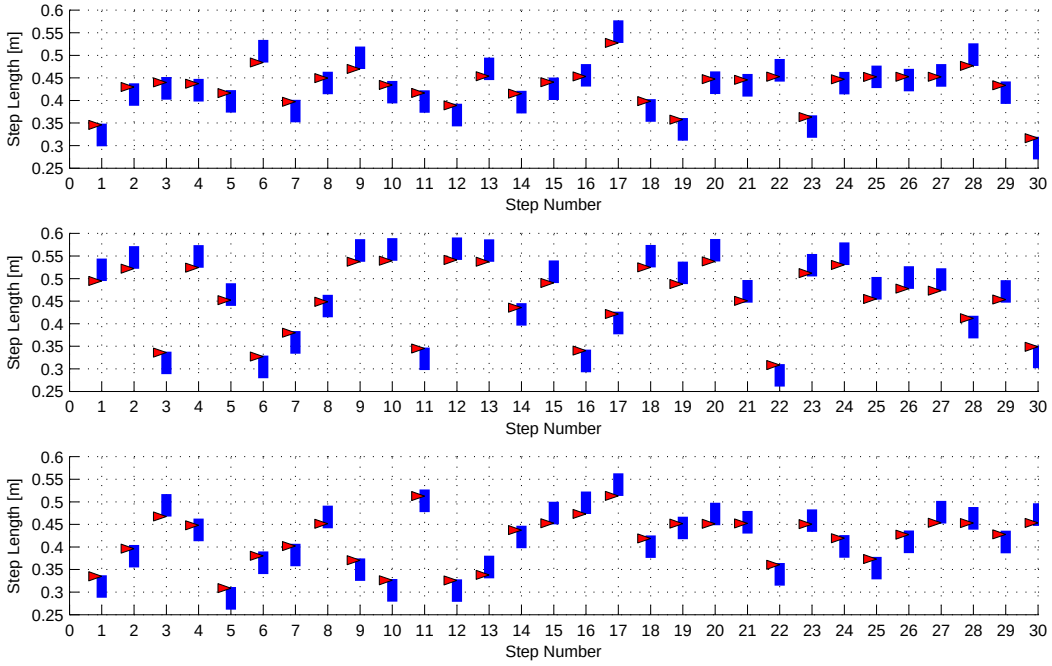


Figure 6.5: Three sample runs of RABBIT walking on a set of 30 randomly generated discrete footholds. The desired step lengths, indicating the distance between the footholds, are chosen randomly in the range $0.25m$ to $0.6m$. Red arrows indicate the robot's resulting step length l_s, and represent the placement of the foot. Blue bars are given ranges of desired step length $[l_{min}, l_{max}]$, indicating the size of the footholds.

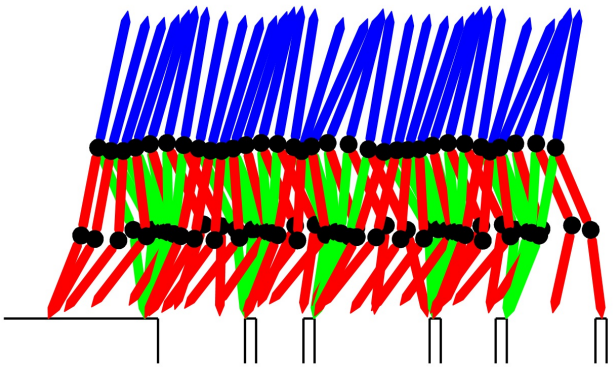


Figure 6.6: Simulation of RABBIT walking over a set of five discrete footholds. The black pillars are locations that the robot needs to step onto, and depict the location and size of the discrete footholds. (Video available at `http://youtu.be/AN-nSHsRLEo`)

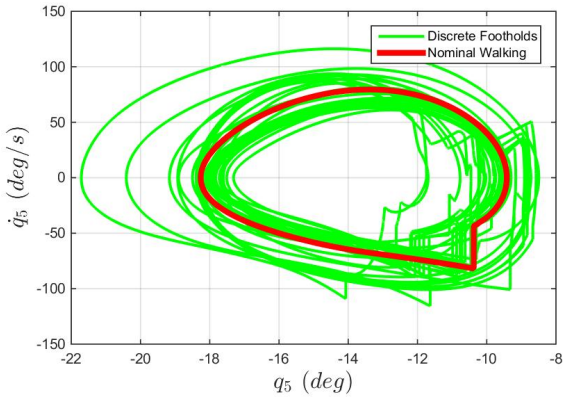


Figure 6.7: Phase plot of RABBIT walking over 30 random discrete footholds. The figure illustrates torso velocity versus the torso angle. The thick red line depicts the nominal limit cycle of the periodic walking gait for comparison.

joints and transmission. Furthermore, we note that this model is not a reduced-order model like the inverted pendulum or the spring-loaded inverted pendulum (SLIP). Finally, although we are illustrating our controller particularly for RABBIT, the proposed method is generalizable and applicable to a variety of bipedal robots with various morphologies.

For RABBIT, the stance phase is parametrized by a suitable set of coordinates, given by $q := (q_1, q_2, q_3, q_4, q_5)$ as illustrated in Fig. 4.1. Here, q_1 and q_2 are the femur angles (referenced to the torso), q_3 and q_4 are the knee angles, and q_5 is the absolute angle of the torso. Because RABBIT has point feet (while many other legged robots have flat feet), the stance phase dynamics are underactuated with the system possessing 4 actuated degrees-of-freedom (DOF) and 1 underactuated DOF.

For the problem of avoiding overhead obstacles, the simulation result for the simpler constraint (6.2) and for the more involved constraint (6.5) is illustrated in Fig. 6.4. For the problem of walking over discrete footholds, we will consider a nominal periodic walking gait or virtual constraint that will provide a stable walking gait for RABBIT. This nominal gait has a step length of $0.45m$. We will run the simulation in multiple steps and apply CBF to change the step length as detailed in Sec. 6.1.3. Desired values of step length will be chosen randomly in the range of $0.25m$ to $0.60m$, corresponding to a -44% to $+33\%$ change in the nominal step length of the gait. These values were empirically determined as the regions for which the controller can guarantee safety while respecting the hardware torque saturation without switching from the nominal gait to another gait. Fig. 6.5 illustrates three sample runs of 30 walking steps. As can be seen from this figure, the desired range $[l_{min}, l_{max}]$ is changed randomly for every step. With the proposed feedback control method, the robot's step length at impact always satisfies this randomly chosen range for each step. Fig. 6.6 illustrates the stick figure of RABBIT for 5 steps of walking, clearly illustrating the precise foot placement on the discrete footholds. Note that the proposed CLF-CBF-QP is solved in under $1ms$ and lends

itself easily to experimental implementations at real-time speeds.

6.2 Dynamic Walking on Time-Varying Stepping Stones

This work presents a novel methodology to achieve dynamic walking for underactuated and hybrid dynamical bipedal robots subject to time-varying position-based constraints, with application to bipedal walking over moving stepping stones (see Fig. 6.8). The proposed controller builds off recent work that addresses the challenges of stabilizing hybrid periodic orbits for dynamic bipedal walking, subject to input and contact constraints [25], model uncertainty [57], safety-critical constraints [58], and simultaneously subject to all the above challenges [60]. The proposed controller is based on the combination of control Barrier functions and control Lyapunov functions implemented as a state-based online quadratic program to achieve stability under input, state and safety constraints. Evaluation of our proposed control design is presented on a model of RABBIT, a five-link planar underacted bipedal robot with point feet.

Note that prior work on footstep planning for dynamic obstacles approaches the problem through ZMP-based controllers [13], resulting in slow, quasi-static locomotion. This work builds off our recent work on precise footstep placement for dynamic bipedal walking [58]. The primary contribution of this work is to develop a method to achieve dynamic walking on moving (time-varying) stepping stones.

The continuous-time dynamics of a bipedal robot is $\dot{x} = f(x) + g(x)u$, where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ are the states and controls respectively. Walking gaits can be formulated through virtual constraints expressed as outputs $y(x) \in \mathbb{R}^m$. Input-output linearizing controllers can be developed resulting in the closed-loop linear output dynamics $\dot{\eta} = F\eta + G\mu$ where $\eta = [y,\ \dot{y}]^T$. While the CLF condition $\dot{V}(x,u) + \lambda V(x) \leq 0$ will ensure stability [25], the state-dependent and time-varying constraints $B(x,t) \geq 0$ can be guaranteed by maintaining the Exponential CBF condition $\dot{B}(x,t,u) + \gamma B(x,t) \geq 0$, see [83, 59] for more details. Combination of these two constraints in a single quadratic programs results in the following CBF-CLF-QP controller:

$$
\begin{aligned}
u^* = \underset{u,\delta}{\operatorname{argmin}} \quad & u^T u + p\delta^2 && (6.12)\\
\text{s.t.} \quad & \dot{V}(x,u) + \lambda V(x) \leq \delta && \textbf{(Stability)}\\
& \dot{B}(x,t,u) + \gamma B(x,t) \geq 0 && \textbf{(Stepping Stones)}\\
& u_{min} \leq u \leq u_{max} && \textbf{(Input Constraints)}
\end{aligned}
$$

To ensure that the swing foot impacts the moving stepping stone, we construct two Barrier functions that ensure the swing foot always moves in the shaded region in Fig. 6.9. We begin by defining the step length

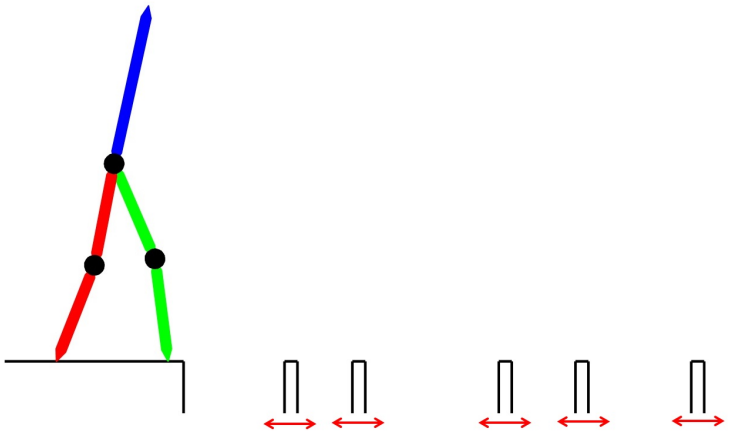


Figure 6.8: Dynamic bipedal dynamic walking with time-varying stepping stones. The stepping stones act as discrete footholds that the robot needs to step onto to ensure safe transversal over the terrain. These stepping stones move with time, stopping once a foot is placed on it. The control problem is to guarantee the safety-critical constraint of placing the swing foot on the stepping stone while dynamically walking. Simulation video: `https://youtu.be/MLBCLE8XBuM`.

at impact as,

$$l_s := l_f(q)|_{h_f(q)=0, \dot{h}_f(q,\dot{q})<0}, \tag{6.13}$$

where $h_f(q)$ is the height of the swing foot to the ground and $l_f(q)$ is the horizontal distance between the stance and swing feet. The discrete foothold constraint to be enforced then becomes,

$$l_{min}(t) \leq l_s \leq l_{max}(t). \tag{6.14}$$

From Fig. 6.9, if we can guarantee the trajectory of the swing foot, F, to be bounded between the domain of the two circles O_1 and O_2, it will imply that the step length when the swing foot hits the ground is bounded within $[l_{min}(t), l_{max}(t)]$. Mathematically, these constraints are expressed as,

$$\begin{aligned} O_1F &= \sqrt{(R_1 + l_f)^2 + h_f^2} \leq R_1 + l_{max}(t), \\ O_2F &= \sqrt{(R_2 + h_f)^2 + (l_f - \frac{l_{min}(t)}{2})^2} \geq \sqrt{R_2^2 + (\frac{l_{min}(t)}{2})^2}. \end{aligned}$$

When the swing foot hits the ground at the end of the step, $h_f = 0, \dot{h}_f < 0$, the step length is l_s, and therefore,

$$\begin{aligned} \sqrt{(R_1 + l_s)^2} &\leq R_1 + l_{max}(t), \\ \sqrt{R_2^2 + (l_s - \frac{l_{min}(t)}{2})^2} &\geq \sqrt{R_2^2 + (\frac{l_{min}(t)}{2})^2}. \end{aligned} \tag{6.15}$$

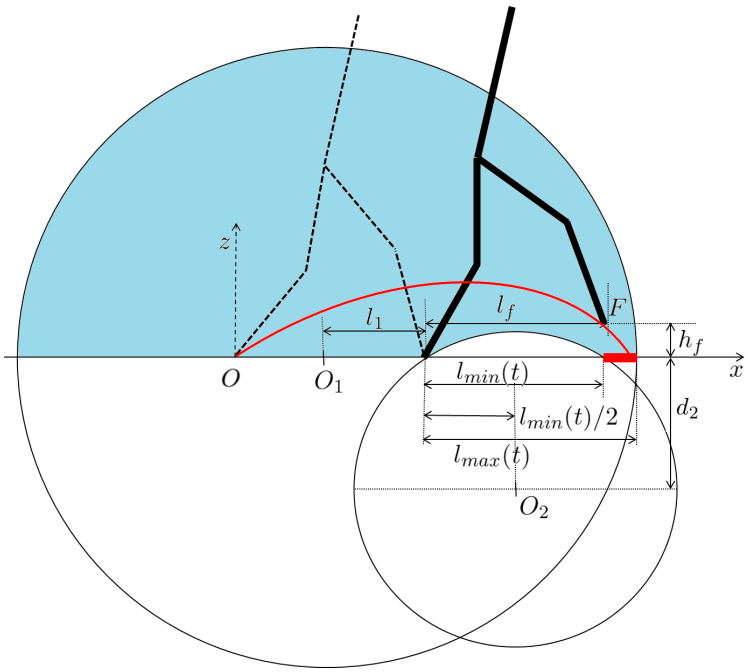


Figure 6.9: The time-varying stepping stones problem reduces to the constraint (6.14) on l_s, the step length at impact as defined by (6.13). This constraint is enforced through the Barriers (6.16).

This then implies the discrete foothold constraint (6.14). We then define the two barrier constraints,

$$\begin{aligned} B_1(x,t) &= R_1 + l_{max}(t) - \sqrt{(R_1+l_f)^2 + h_f^2} \geq 0, \\ B_2(x,t) &= \sqrt{(R_2+h_f)^2 + (l_f - \frac{l_{min}(t)}{2})^2} \\ &\quad - \sqrt{R_2^2 + (\frac{l_{min}(t)}{2})^2} \geq 0, \end{aligned} \tag{6.16}$$

and then apply the CBF-CLF-QP controller (6.12).

Simulating this controller on a model of RABBIT, results in Fig. 6.25, which illustrates the plot of step length for 10 walking steps. The variation of the desired foothold over time and the satisfaction of footstep placement at the end of every walking step is also seen. Note that the proposed controller requires a perfect model of the robot. Our results in [60] can potentially be used to address model uncertainty.

6.3 3D Stepping Stones with Control Barrier Functions

To enforce precise foot placements on the stepping stones, we need to enforce strict constraints on the step length and step width at swing foot touchdown (see Fig. 6.11). To achieve this, we will first develop

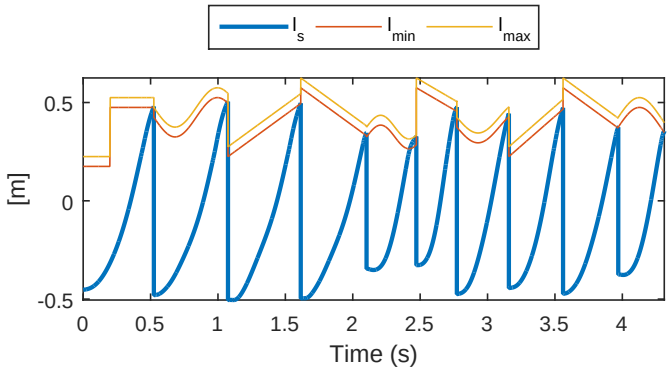


Figure 6.10: Ten steps of the horizontal position of the swing leg with respect to the stance leg (thick blue), and desired stone locations varying either linearly or sinusoidally over time (thin orange) are shown. The proposed controller guarantees that the swing leg impacts the moving stepping stone.

CBFs that enforce strict step length constraints, followed by CBFs that enforce strict step width constraints. We will finally put these together to achieve simultaneous step length and step width constraints to achieve 3D walking on the stepping stones.

6.3.1 Stepping constraints for step length

If we want to force the robot to step onto a specific position (see Fig.6.9), we need to guarantee that the step length when the robot swing foot hits the ground is bounded within a given range $[l_{min}, l_{max}]$. Let $h_f(q)$ be the height of the swing foot to the ground and $l_f(q)$ be the distance between the stance and swing feet, as defined in Fig. 6.11. We then can apply the same approach to enforce desired step length for 2D walking presented in Section. 6.1.3.

6.3.2 Stepping constraints for step width

A sufficient condition to guarantee the step width $w_f \leq w_{max}$, is to maintain the swing foot position to be inside the circle O_3 (see Fig.6.12), i.e.,

$$O_3F \leq R_3, \tag{6.17}$$

where,

$$O_3F = \sqrt{(l_0 - l_f)^2 + (d_3 + w_f)^2}. \tag{6.18}$$

The circle O_3 is defined so that it is tangent to the maximum boundary of desired step width and containing the initial swing foot position (O). Therefore the radius and center of the circle O_3 can be derived from l_3

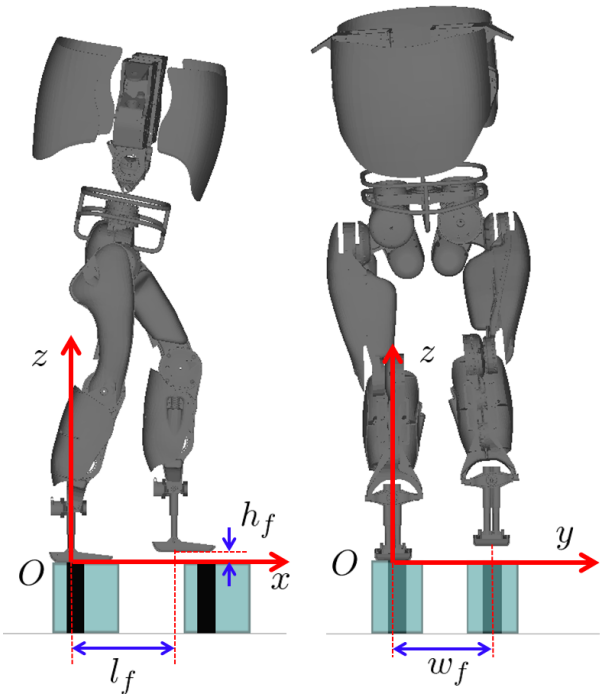


Figure 6.11: DURUS swing foot coordinates: The swing foot position in 3D is defined by the step length, step width, and step height coordinates (l_f, w_f, h_f) along the x, y, z axes respectively.

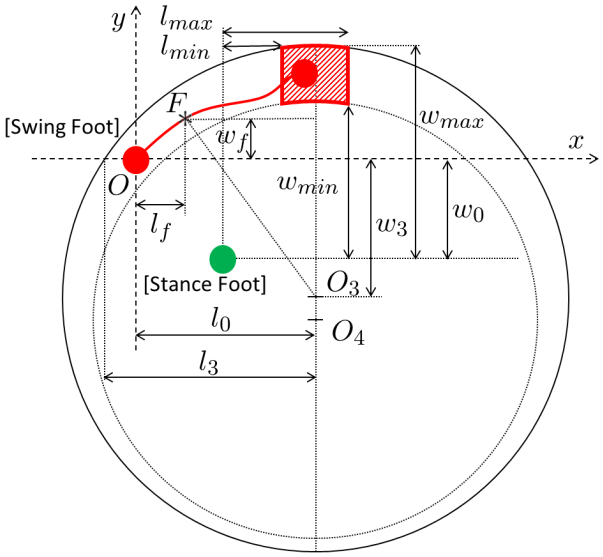


Figure 6.12: Geometric explanation of the stepping stone foot placement constraint with only step width constraint.

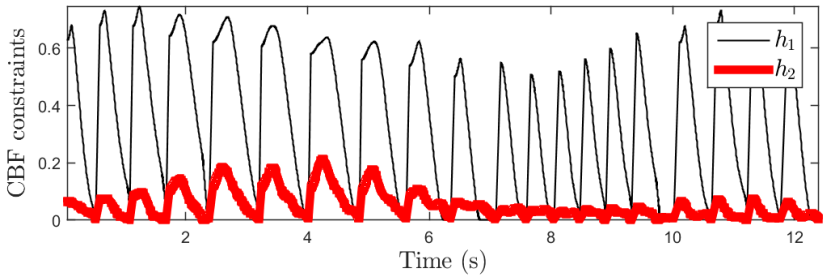


Figure 6.13: CBF constraints for Case 1 (changing step length only).

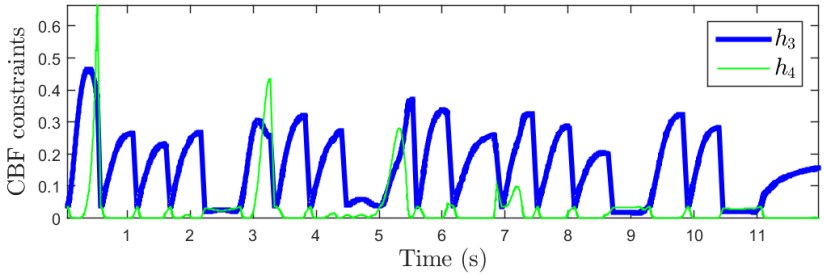


Figure 6.14: CBF constraints for Case 2 (changing step width only).

and w_{max} as follows:

$$\begin{aligned} R_3^2 = l_3^2 + w_3^2 = l_3^2 + (R_3 - w_{max} + w_0)^2, \\ \implies R_3 = \frac{l_3^2 + (w_{max} - w_0)^2}{2(w_{max} - w_0)}, \end{aligned} \tag{6.19}$$

where w_0 is the step width of the previous step. The same principle can be applied to enforce the minimum step width $w_f \geq w_{min}$ by requiring $O_4F \geq R_4$. These constraints can be enforced through a CBF controller from Section 5.1 by defining the following position constraints,

$$\begin{aligned} h_3(q) = R_3 - O_3F \geq 0, \\ h_4(q) = O_4F - R_4 \geq 0, \end{aligned} \tag{6.20}$$

along with Barriers B_3, B_4 to enforce them. Thus, the step width will satisfy the desired foothold $w_s \in [w_{min}, w_{max}]$.

6.3.3 Numerical Validation of 3D Dynamic Walking on Stepping Stones

Having developed the control framework based on control Lyapunov and control Barrier functions, we will now numerically validate the proposed controller on DURUS. We will consider three different simula-

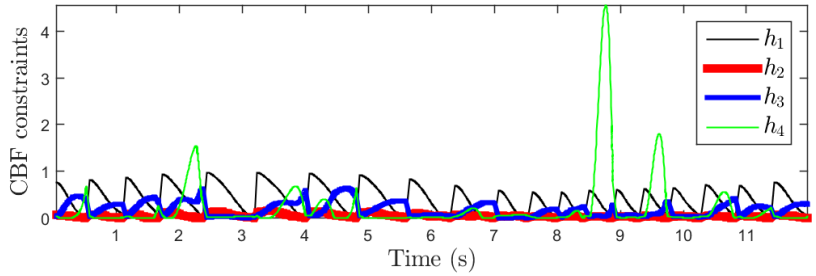


Figure 6.15: CBF constraints for Case 3 (changing step length and width)

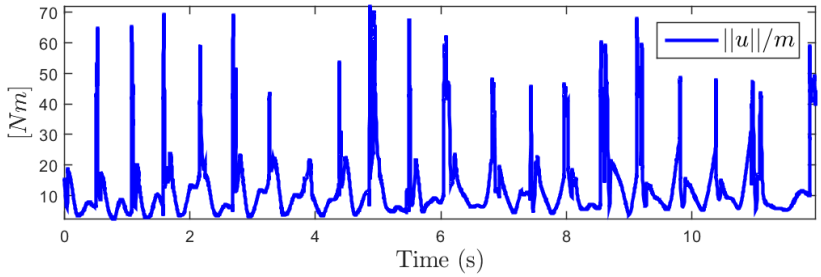


Figure 6.16: Norm of torques for Case 3 (changing step length and width).

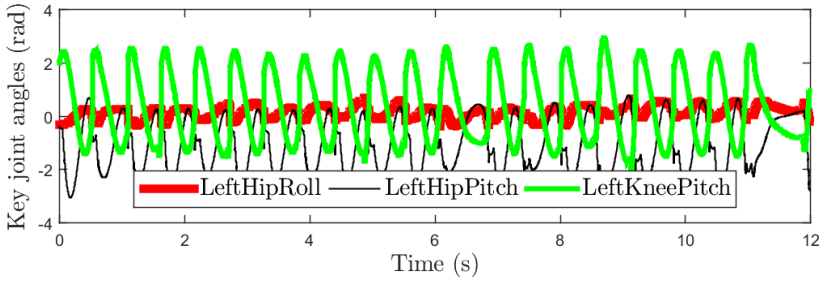


Figure 6.17: Key joints angles of left leg for Case 3 (changing step length and width).

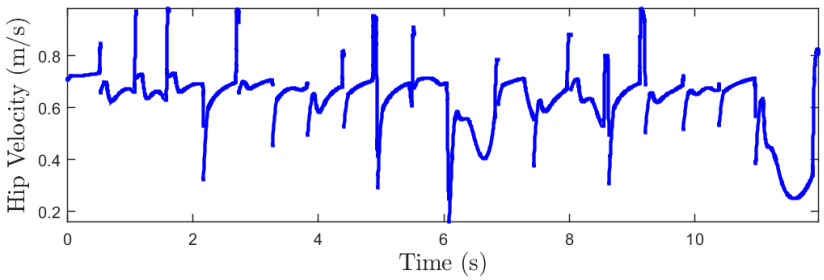


Figure 6.18: Hip Velocity for Case 3 (changing step length and width).

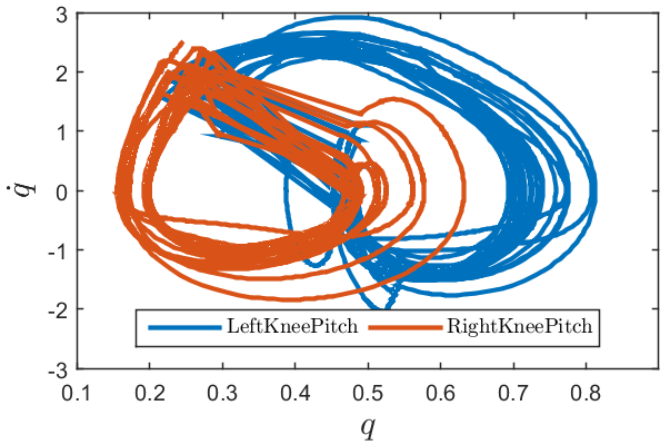


Figure 6.19: Phase portrait of left and right knee pitch for Case 3 (changing step length and width).

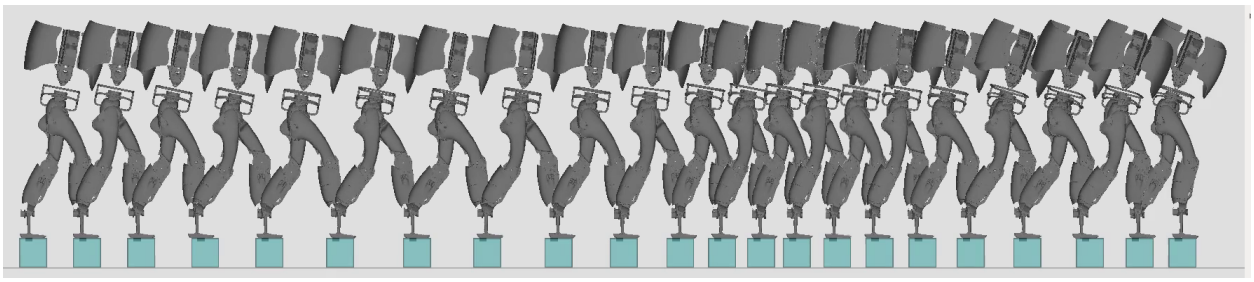


Figure 6.20: Snapshots of simulation results (Case 1: Changing step length only). Simulation video: `https://youtu.be/yUSTraDn9-U`.

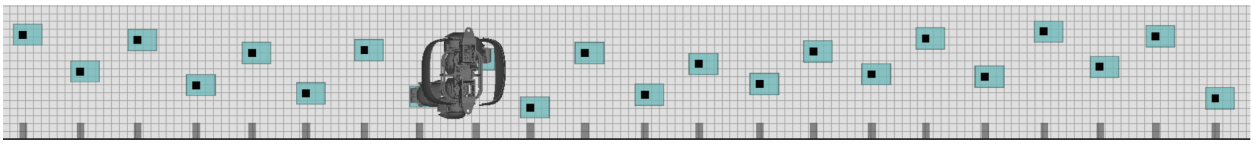


Figure 6.21: Snapshots of simulation results (Case 2: Changing step width only). The small vertical bars at the bottom of the figure illustrate that the step length doesn't change.

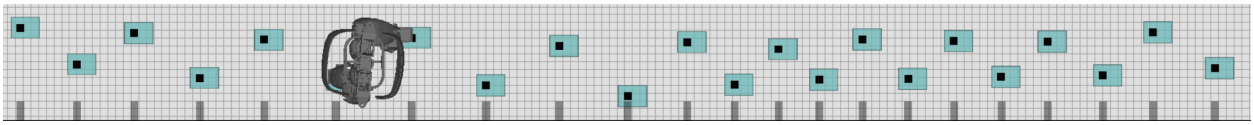


Figure 6.22: Snapshots of simulation results (Case 3: Changing step length and step width) From the small vertical bars at the bottom of the figure, it's clear that the step length also changes in addition to step width.

tion cases:

- Case 1: Changing step length only: $l_s \in [l_{min}, l_{max}]$,
- Case 2: Changing step width only: $w_s \in [w_{min}, w_{max}]$,
- Case 3: Changing both step length and width,

where l_{min}, l_{max} and w_{min}, w_{max} were picked with an offset of $\pm 2.5cm$ about the desired footstep location l_d, w_d. To be more specific, we have:

$$\begin{aligned} l_{min} &= l_d - 0.025,\ l_{max} = l_d + 0.025, \\ w_{min} &= w_d - 0.025,\ w_{max} = w_d + 0.025. \end{aligned} \tag{6.21}$$

The controller is simulated with l_d and w_d chosen randomly at each step. While the nominal walking gait has step length $36.6cm$ and step width $23.3cm$, our proposed controller was able to address a large range of desired footholds with the following specific values:

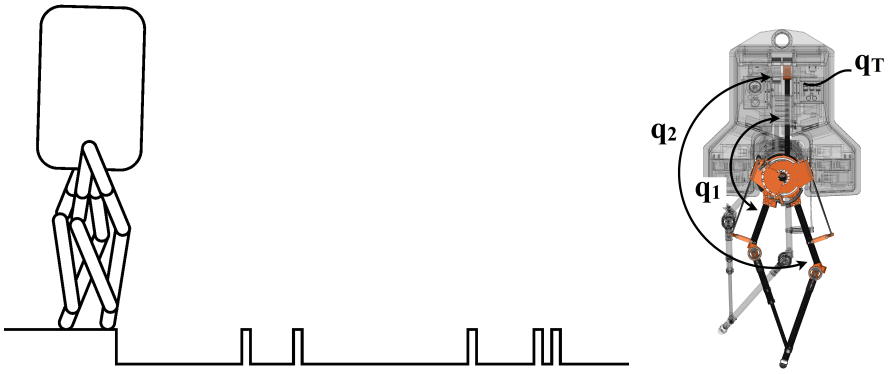
- Case 1: Changing step length only: $l_d \in [22cm, 50cm]$ ($\pm 39\%$ of the nominal step length.);
- Case 2: Changing step width only: $w_d \in [12cm, 33cm]$ (-48% to 43% of the nominal step width);
- Case 3: Changing both step length and width: $l_d \in [24cm, 47cm]$ (-33% to 31% of the nominal step length), $w_d \in [20cm, 33cm]$ (-13% to 43% of the nominal step width).

Beyond these step length and step width values, the initial condition for the subsequent step is too far from the nominal walking gait and the controller is unable to track.

Figures 6.13,6.14,6.15 clearly show that the CBF constraints on footstep placement are strictly satisfied for all three simulation cases. Figures 6.20,6.21,6.22 illustrate snapshots of the simulation with footstep constraints for the three cases respectively. For Case 3 (changing both step length and width), Figures 6.17, 6.18, 6.19, illustrate the key joint angles, hip velocity and phase portrait respectively.

6.4 Dynamic Walking on Stepping Stones with Control Barrier Functions and Gait Library

We have presented optimization approach to create an individual walking gait using Hybrid Zero Dynamic in Chapter. 3, we will next discuss the design of a finite set of gaits and a means to create from it a continuum of gaits, called the gait library.



(a) (b)

Figure 6.23: (a) The problem of dynamically walking over a randomly generated set of discrete footholds. Simulation video: https://youtu.be/udpxZUXBi_s. (b) The coordinates of the biped are illustrated with the torso world frame pitch angle denoted by q_T, and the body coordinates denoted by (q_1, q_2).

Gait Library and Interpolation

The optimization problem posed in the previous section is used to generate five gaits having step lengths $L_{step} = \{0.08, 0.24, 0.40, 0.56, 0.72\}$ meters[1]. For values of step length between the discrete values, $L_{step,i}$, $1 \leq i \leq 5$, define the Beziér coefficients α in (3.2) by linear interpolation of the coefficients α_i for the five nominal step lengths. In particular, define,

$$\zeta(L_{step}) = \frac{L_{step} - L_{step,i}}{L_{step,i+1} - L_{step,i}}, \;\; 1 \leq i \leq 4 \tag{6.22}$$

$$\alpha(L_{step}) = (1 - \zeta(L_{step}))\alpha_i + \zeta(L_{step})\alpha_{i+1}. \tag{6.23}$$

For step lengths longer than 0.72, linear extrapolation is used. As in [19, Eqn. (8,9)], this defines a continuum of gaits, called the gait library

$$\mathcal{A} = \{\alpha(L_{step}) \mid 0.08 \leq L_{step} \leq 0.72\}. \tag{6.24}$$

The update resets the periodic orbit to adapt the step length, while respecting the physical constraints and approximately optimizing the cost on the periodic orbit. During steady-state, there is no theoretical guarantee that the interpolated gait results in exactly the desired step length. However, due to continuity of each pre-defined gait in the library, the interpolated gait will result in a step length that is close enough to the desired step length. On the other hand, during a transient following a change in commanded step length, the footstep placement and optimization constraints shown in Table.3.1 are not guaranteed to be satisfied. In

[1]The number of gaits is arbitrary. A finer grid did not change the results. A coarser grid was not tried.

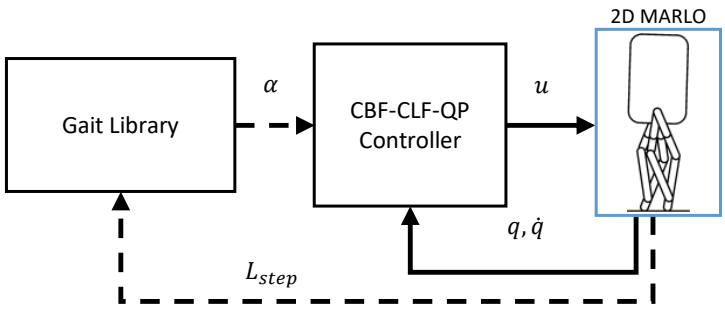


Figure 6.24: Diagram of the controller structure integrating the gait library and CBF based controller. Solid lines indicate signals in continuous time representing a within-stride controller; dashed lines indicate signals in discrete time representing a stride-to-stride controller.

the next Section, we will introduce the method of control barrier functions to handle transients in the form of real-time constraints on footstep placement, scuffing avoidance, friction cone, and input saturation.

6.4.1 Safety-Critical Control for Dynamical Bipedal Walking with Precise Footstep Placement

Constraints on Footstep Placement

If we want to force the robot to step onto a specific position (see Fig. 7.7), we need to guarantee that the step length when the robot swing foot hits the ground is bounded within a given range $[l_{min}; l_{max}]$. Let $h_f(q)$ be the height of the swing foot to the ground and $l_f(q)$ be the horizontal distance between the stance and swing feet. We then can apply the same approach to enforce desired footstep placement presented in Section. 6.1.3.

Remark 6.2. *The gait library provides gaits that enforce constraints during steady-state but not during transients that occur when the gait is switched. The CBF controller guarantees constraints during transients, thereby preventing a combinatorial explosion of pre-computed transient gaits. The combination of the gait library and the CBF controller is then able to handle constraints during both steady-state and transient phases. The typical failure mode of the gait library is foot scuffing of the swing foot during gait switches. Using a CBF to maintain the swing foot position to be outside the circle O_2, guarantees both the lower bound constraint on step length ($l_s \geq l_{min}$) and foot scuffing avoidance simultaneously. The choice of R_2 in Fig. 6.3 can be designed based on the desired mid-step swing foot clearance.*

Constraints on Friction Cone

In bipedal robotic walking, contact force constraints are very important for the problem of robotic walking. Any violation of these constraints will result in the leg slipping and the robot potentially falling. Although walking gait optimization is usually designed to respect these constraints, we cannot guarantee these constraints when switching between different walking gaits. In particular, we consider, $F^h_{st}(x,u)$ and $F^v_{st}(x,u)$, the horizontal and vertical contact force between the stance foot and the ground (or the friction force and the normal ground reaction force). Then, the constraints to avoid slipping during walking are,

$$\begin{aligned} F^v_{st}(x,u) &\geq \delta_N > 0, \\ \frac{|F^h_{st}(x,u)|}{|F^v_{st}(x,u)|} &\leq k_f. \end{aligned} \tag{6.25}$$

where δ_N is a positive threshold for the vertical contact force, and k_f is the friction coefficient. We enforce the above ground contact constraints with $\delta_N = 150(N), k_f = 0.6$.

Remark 6.3. *Note that since the gait optimization is performed offline, we enforce stricter constraints (ground reaction force $F^v_{st} \geq 200(N)$ and friction cone $\mu \leq 0.4$) (see Table 3.1), allowing for a margin of safety. These constraints hold only for the gaits in the gait library and not for the transient steps generated by the gait library controller. Our ECBF-CLF-QP controller enforces the constraints $F^v_{st} \geq 150(N)$ and friction cone $\mu \leq 0.6$ in real-time for the transient steps.*

We then have the following ECBF-CLF-QP based controller that can handle simultaneously footstep placement, scuffing avoidance, friction constraint and input saturation:

$$\begin{aligned} \mu^*(x) = \underset{\mu, d_1}{\text{argmin}} \quad & \mu^T \mu + p_1 \, d_1^2 & \\ \text{s.t.} \quad & \dot{V}(x,\mu) + \lambda V(\eta) \leq d_1 & \textbf{(CLF)} \\ & \dot{B}_1(x,\mu) + \gamma B_1(x) \geq 0 & \textbf{(ECBF on } l_s \leq l_{max}\textbf{)} \\ & \dot{B}_2(x,\mu) + \gamma B_2(x) \geq 0 & \textbf{(ECBF on } l_s \geq l_{min} \\ & & \textbf{\& Foot Scuffing)} \\ & F^v_{st}(x, u(x,\mu)) \geq \delta_N > 0 & \textbf{(Normal Force)} \\ & \frac{|F^h_{st}(x, u(x,\mu))|}{|F^v_{st}(x, u(x,\mu))|} \leq k_f & \textbf{(Friction Cone)} \\ & u_{min} \leq u(x,\mu) \leq u_{max} & \textbf{(Input Saturation)} \end{aligned} \tag{6.26}$$

Remark 6.4. *Note that all the constraints are affine in μ and thus the above optimization problem is still a quadratic program that can be solved in real-time.*

Remark 6.5. *The gait library approach offers a switching strategy under a wide range of step lengths. Based on the desired step length, the interpolation between different gaits in the library will result in a new walking gait for the next step. If the system state is on or close enough to the periodic orbit, it will converge to the desired step length while maintaining physical constraints mentioned in Table.3.1. However, in our problem, we want the robot to be able to switch between two gaits with very different step lengths, the initial condition is basically very far from the periodic orbit of the next step. Therefore, the transition to the new gait is not guaranteed to satisfy constraints such as friction constraints as well as scuffing avoidance. In the simulation, these two main reasons make the gait library approach fail almost all the time.*

Note that the CBF-CLF-QP controller in [58] is only based on one nominal gait and tries to adjust the control inputs so as to enforce the footstep placement constraint, friction constraints and input saturation while following the nominal gait. Due to the limitation of having only one walking gait, the working range of step length is therefore limited.

In this work, we attempt to combine the advantages of each method and develop the ECBF-CLF-QP controller with foot scuffing constraints and combine it with the gait library approach (see Fig.7.4). Given a desired step length, the gait library assigns an interpolated gait for the next walking step and the ECBF-CLF-QP controller tracks the outputs corresponding to this gait by solving a quadratic program in real-time to find the control input that follows this new gait while maintaining all above constraints (footstep placement, friction constraints, scuffing avoidance and input saturation).

In the next Section, we present numerical validation of our proposed controller on the dynamical model of the bipedal robot MARLO.

6.4.2 Numerical Validation of Dynamic Walking on Stepping Stones with CBF and Gait Library

In this Section, we will demonstrate the effectiveness of the proposed method by conducting numerical simulations on the model of MARLO. We validate the performance of our proposed approach through dynamic bipedal walking on MARLO, while simultaneously enforcing foot placement, scuffing avoidance, ground contact force constraints and input saturation. Furthermore, in order to demonstrate the effectiveness of the method, we compare three controllers on different ranges of desired step lengths:

$$\begin{cases} \text{I: Gait Library} \\ \text{II: CBF (with nominal step length of 56 cm)} \\ \text{III: CBF \& Gait Library} \end{cases} \tag{6.27}$$

Table 6.1: **(Main Result)** Percentage of successful tests of three controllers (see (6.27)) with different ranges of desired step length.

Step Length Range (cm)	Gait Library	CBF	Gait Library & CBF
[50:60]	6%	100%	100%
[40:70]	1%	44%	100%
[30:80]	1%	17%	100%
[25:85]	1%	12%	100%
[20:90]	1%	3%	97%
[15:95]	1%	0%	92%
[10:100]	0%	0%	78%

For each range of step length (see Table 6.1), we randomly generated 100 problem sets, where each set has 10 randomly placed "stepping stones" with a stone size of 5 (cm) (see Fig. 7.7). The controller is considered successful for a trial run if the bipedal robot is able to walk over this terrain without violation of foot placement, ground contact, friction, and input constraints. The percentage of successful tests for each of the three controllers is tabulated in Table 6.1 for various ranges of step lengths. The approach based on the combination of CBF and Gait Library outperforms the approaches that rely on only the CBF or only the Gait Library. For example, with the step length range of [20:90] (cm), the percentage of successful tests on controller III (CBF and Gait Library) is 97 %, while that of controller II (CBF only) and controller I (Gait Library only) are just 3% and 1% respectively. Thus the proposed controller not only achieves dynamic walking over discrete footholds, it also dramatically increases the range of step lengths that are handled compared to our prior work in [58].

We show here one simulation of MARLO walking over 20 stepping stones with desired step lengths randomly generated in the range of $[10 : 100]$ (cm), where the stone size is smaller, i.e., $l_{max} - l_{min} = 2$ (cm). Fig. 6.25 shows the satisfaction of foot step placement constraints as well as CBF constraints, without a violation of the friction cone or input saturation (see Fig. 6.26). Furthermore, in order to illustrate how aggressively our proposed method can traverse a set of stepping stones, Fig. 6.27 shows a simulation where the robot has to switch between very a large step length (95 cm) and a very small step length (15 cm).

Remark 6.6. *Potential reasons of the gait library and CBF controller not reaching 100% success in the last three cases include (a) the QP* (6.26) *becoming feasible; and (b) the gait library being limited to a pre-computed gait at 72 (cm) (see Section 6.4), which is extrapolated by 25-39% to reach step lengths of 90-100 (cm). Including a pre-computed gait with a step length of 100 (cm) could potentially improve performance.*

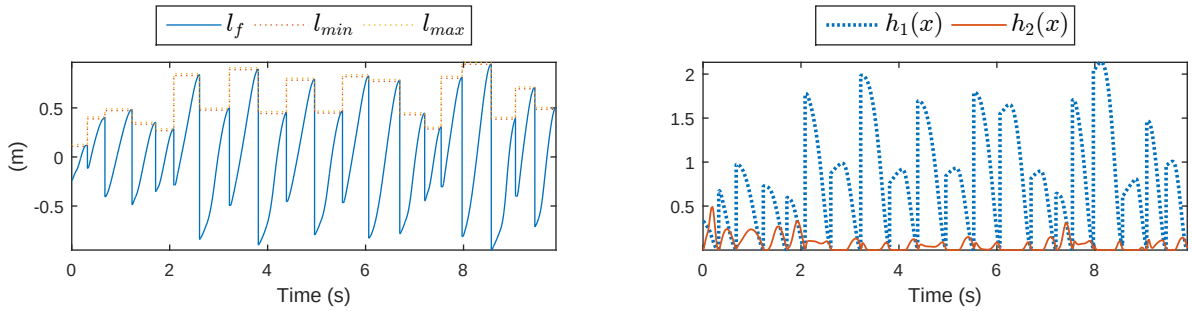


Figure 6.25: (a) Footstep placement constraint: $l_{min} \leq l_s \leq l_{max}$, where the step length l_s is the value of the distance between swing and stance feet l_f at impact (see (6.13)). (b) CBF constraints: $h_1(x) \geq 0, h_2(x) \geq 0$.

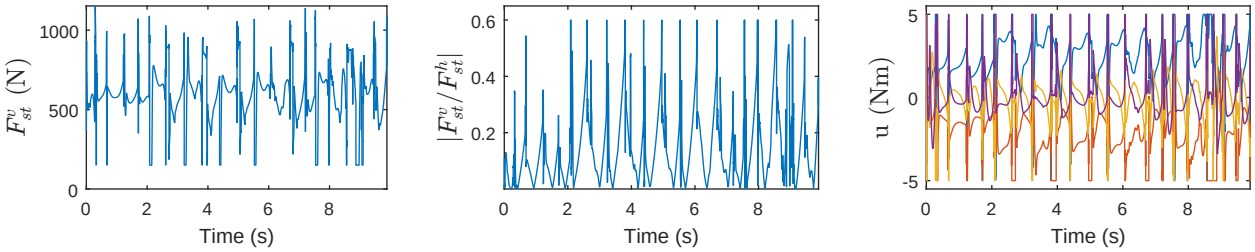


Figure 6.26: (a) Ground reaction force: $F^v_{st} \geq 150(N)$. (b) Friction cone: $|F^h_{st}/F^v_{st}| \leq 0.6$. (c) Control inputs are saturated at 5 (Nm) ($|u| \leq 5$); recall the 50:1 gear ratio from the motors to the links.

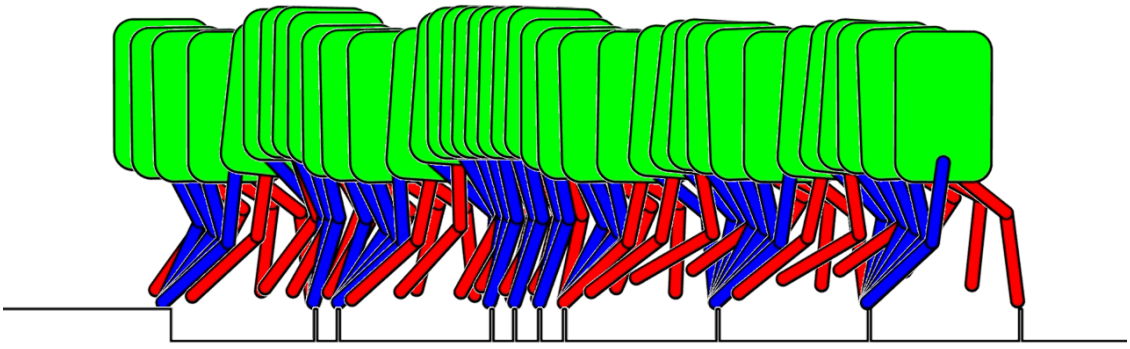


Figure 6.27: Simulation of MARLO walking over stepping stones with desired step lengths of $\{95, 15, 95, 15, 15, 15, 95, 95, 95, 15, 15, 15\}(cm)$ and stone size of 2 (cm). For clarity of visualization, the rear links of the 4-bar are suppressed. Simulation video: https://youtu.be/udpxZUXBi_s.

6.5 Summary

In this Chapter, we have presented our method to handle aperiodic walking with application of dynamic walking on stepping stones using control Barrier functions. We then extend the work to the problem of time-varying stepping stones and 3D dynamic walking on stepping stones with both step length and step width varied on the model of DURUS, a 23 DoFs humanoid robot. While these works all use only one nominal gait, we can improve the performance significantly by incorporating CBFs with a gait library. This novel combination is able to increase the step length range from ($[25 : 60](cm)$) without the gait library to ($[10 : 100](cm)$) with the gait library.

Chapter 7

Dynamic Walking on Stepping Stones with 2-Step Periodic Gait Library

While handling physical constraints and step transition via CBF presented in Section. 6 appears to work well, these constraints often become active at step switching. In order to resolve this issue, this chapter will introduce a new method of 2-step periodic gait optimization to address the problem of dynamic walking on stepping stones. The method takes into account the gait transition between different configurations of stepping stones, resulting in a smooth step transition at impact. It also opens the ability to handle the problem of dynamic walking on discrete terrain with varied step length and step height. Our proposed methods are successfully validated on ATRIAS, an underactuated, human-scale bipedal robot. In particular, experimental demonstrations illustrate our controller being able to dynamically walk at 0.6 m/s over terrain with step length variation of 23 to 78 cm, as well as simultaneous variation in step length and step height of 35 to 60cm and -22 to 22 cm respectively.

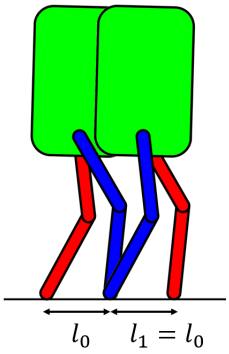


Figure 7.1: Periodic walking gait has the resulting step length (l_1) similar to the initial step length (l_0), or in other words $l_1 = l_0$.

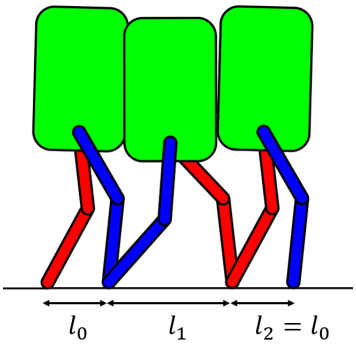


Figure 7.2: 2-Step periodic walking with changing step lengths only. The walking gait is 2-step periodic therefore the step length of the second step and that of the initial condition are the same ($l_2 = l_0$).

7.1 2-Step Periodic Gait Design Using Virtual Constraints

We presented the Hybrid Zero Dynamics framework in Chapter. 3 to derive a periodic gait with constant step length (see Fig. 7.1). In order to take into account the gait transition when we want to change the footstep location every walking step, we now introduce the 2-step periodic gait optimization. Inspired by the main issue of step transition on stepping stones [55], we develop an optimization framework to design 2-step periodic walking gaits, taking into account not only the desired footstep location of the next step but also the current configuration of the robot. The method combines virtual constraints, parameter optimization, and an interpolation strategy for creating a continuum of gaits from a finite library of gaits. The notion of a 2-step periodic gait means that the robot states are designed to be converge back to the initial condition after 2 walking steps. To be more specific, we will start off with the problem of changing step length only or walking on flat ground with varied step length.

7.1.1 Changing Only Step Lengths

In the nominal problem of periodic optimization presented in Section 3.2, we need to optimize for only one walking step with the constraint on the resulting step length (l_1) to be equal to the initial step length (l_0) (see Fig. 7.1). For this problem, we use the same optimization framework discussed in Section 3.2, but we will optimize for 2 walking steps while following additional constraints that allows us to have different step lengths during transition (see Fig. 7.2):

- The initial state at start of the first step is given by $x = x_0^+$ with corresponding (initial) step length l_0.
- The state at the end of the first step (before impact) is $x = x_1^-$ with (resulting) step length l_1.
- Impact constraints at the end of the first step are enforced as $x_1^+ = \Delta(x_1^-)$.

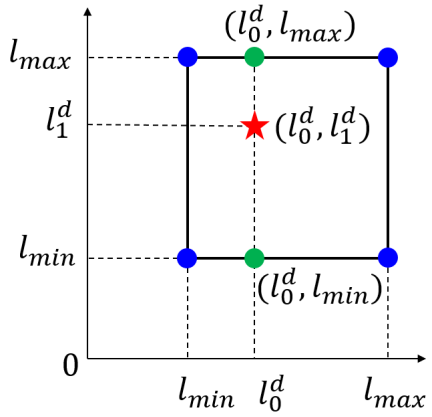


Figure 7.3: Gait interpolation for the problem of changing step length only.

- The initial state at start of the second step is given by $x = x_1^+$ with corresponding (initial) step length of l_1.
- The state at the end of the second step (before impact) is $x = x_2^-$ with (resulting) step length of l_2.
- Impact constraints at the end of the second step are enforced as $x_2^+ = \Delta(x_2^-)$.
- Periodic constraints are then enforced as $x_2^+ = x_0^+$, resulting in $l_2 = l_0$.

The optimization problem is then used to generate a gait library with different values of l_0 and l_1. In this work, we optimize four different gaits corresponding to:

$$\begin{cases} (l_0, l_1) = (0.3, 0.3)\ m \\ (l_0, l_1) = (0.3, 0.7)\ m \\ (l_0, l_1) = (0.7, 0.3)\ m \\ (l_0, l_1) = (0.7, 0.7)\ m. \end{cases} \tag{7.1}$$

This is similar to precomputing four gait primitives corresponding to walking with small steps ($(l_0, l_1) = (0.3, 0.3)\ m$), switching from a small step to a large step ($(l_0, l_1) = (0.3, 0.7)\ m$), switching from a large step to a small step ($(l_1, l_0) = (0.7, 0.3)\ m$) and walking with large steps ($(l_0, l_1) = (0.7, 0.7)\ m$). Having a gait library with different gaits representing a few general motion primitives, we then do gait interpolation to get the desired walking gait with an arbitrary set of (l_0^d, l_1^d).

Let $\alpha(l_0^d, l_1^d)$ be the Beziér coefficients (defined in (3.2)) of the desired walking gait that has the initial step length l_0^d and the resulting step length l_1^d. If $l_0^d \in [0.3 : 0.7]\ m$ and $l_1^d \in [0.3 : 0.7]\ m$, we will compute $\alpha(l_0^d, l_1^d)$ using bilinear interpolation of the coefficients from the four nominal gait parameters precomputed

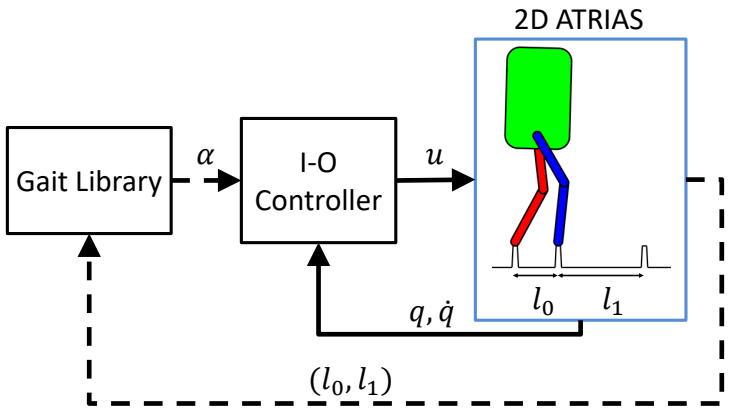


Figure 7.4: Diagram of the controller structure for the problem of changing step length only, integrating the gait library and I-O linearization controller. Solid lines represent signals in continuous time; dashed lines represent signals in discrete time.

using optimization. Therefore the new interpolated gait can be derived in real-time (1 kHz in our experiment) right after the impact of each walking step. A detailed explanation for bilinear interpolation can be found in [81], and summarized in Fig. 7.3. In this work, we use the MATLAB function "interp2" to implement the algorithm.

Remark 7.1. *If $l_0 \notin [0.3 : 0.7]$ m or $l_1 \notin [0.3 : 0.7]$ m, we can use extrapolation to compute the gait parameters for the desired gait.*

The gait library and gait interpolation are used to update the walking gait for every walking step based on the desired footstep placement of the next step (l_1) and the current configuration of the robot (l_0). They are then incorporated using input-output linearization to control the robot to follow the updated walking gait. The closed-loop control diagram is shown in Fig. 7.4.

Remark 7.2. *Note that the proposed method has a resemblance to MPC. While we use a 2-step periodic gait, we switch the gait at the end of each step, i.e., half-way into the 2-step periodic gait. For instance, with current step length being l_0, and subsequent step lengths being l_1, l_2, we use a gait with (l_0, l_1) and switch at the end of the first step to (l_1, l_2) so that there is an overlap of one step between the gaits. This easily addresses gait transitions that typically cause large violations in unilateral force constraints, friction constraints, and torque constraints.*

Remark 7.3. *Also note that in [55], we use control barrier functions to handle gait transitions. While this appears to work well, the feasibility of the quadratic program that enforces the control barrier constraint is not guaranteed. In this present work, as we will see, we achieve better results without using control barrier functions. We can easily add control barrier functions on top of the current method to further enforce these*

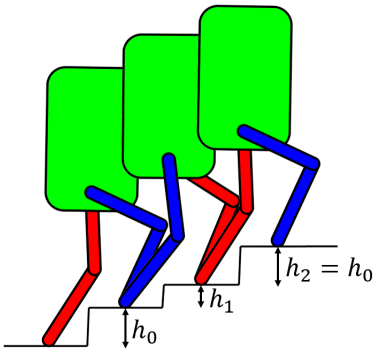


Figure 7.5: 2-Step periodic walking with changing step heights only. The walking gait is 2-step periodic therefore the step height of the second step and that of the initial condition are the same ($h_2 = h_0$). Note that step heights h_0, h_1 can be positive (stepping up) or negative (stepping down). In this figure, we illustrate the case of both h_0 and h_1 being positive for convenience.

safety-critical constraints. Since the underlying method achieves the foot placement without requiring the barriers, the barriers will remain inactive most of the time, leading to better feasibility of the quadratic program.

7.1.2 Changing Only Step Heights

For the problem of changing step heights only, the framework presented for changing step length on flat ground in Section 7.1.1 can be applied where (h_0, h_1) plays the role of (l_0, l_1) (see Fig. 7.5). For the problem of changing step length on flat ground, the step height is zero. In the problem of changing step height only in Fig. 7.5, we assume that the step length or the width of the stair is constant.

7.1.3 Changing Both Step Lengths and Step Heights

We now can combine the methods presented in Section 7.1.1 and 7.1.2 to handle the problem of walking on stepping stones with varied step length and step height for every walking step (see Fig. 7.6). Since the gait parameters now depends on 4 variables l_0, h_0, l_1, h_1, the gait interpolation needs to be extended for 4 variables and the number of gaits increases to $2^4 = 16$ gaits.

Remark 7.4. *Note that each of the nominal periodic walking gaits presented in Section 3.2 is locally exponentially stable [80]. The stability condition for switching policies between different locally exponentially stable periodic gaits can be found in [54], wherein it is assumed that one periodic gait switches into the domain of attraction of a subsequent periodic gait. With the 2-step periodic gaits, we can guarantee that when we switch to the next gait, the initial state of the robot is close to the periodic orbit of the next gait.*

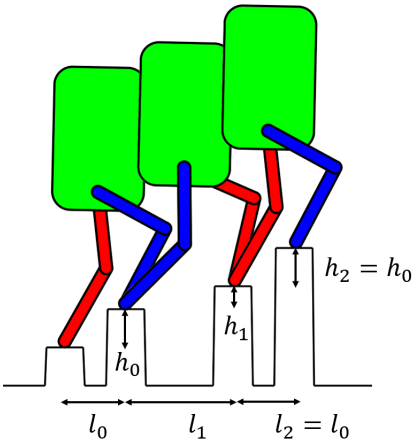


Figure 7.6: 2-Step periodic walking with changing step lengths and step heights. The walking gait is 2-step periodic therefore the step length and step height of the second step and that of the initial condition are the same ($l_2 = l_0, h_2 = h_0$). Note that step heights h_0, h_1 can be positive (stepping up) or negative (stepping down). In this figure, we illustrate the case of both h_0 and h_1 being positive for convenience.

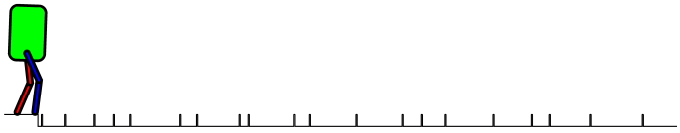
Remark 7.5. *Creating an interpolated gait between two nearby periodic gaits ensures that constraints enforced by the periodic gaits are enforced by the interpolated gait as well due to local continuity of the constraints. Further work needs to be carried out to establish stability results for switching between interpolated gaits.*

7.2 Numerical Validation of 2-Step Periodic Gait Library

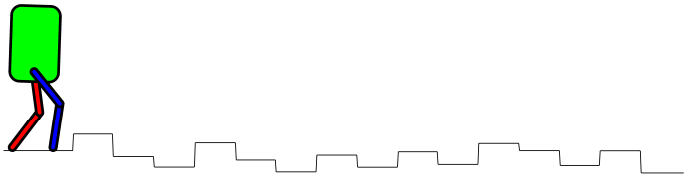
In this Section, we will demonstrate the effectiveness of the proposed method by conducting numerical simulations on the model of ATRIAS.

Using our method, we can control our robot to overcome different type of terrains:

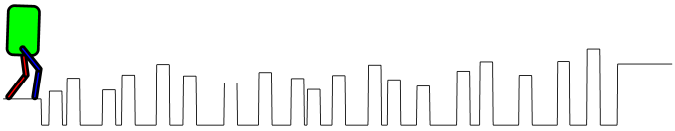
- Changing step length only within the range of [20:90] (cm) with the precision of only 2 (cm) (see Fig. 7.7a)
- Changing step height only within the range of [-30:30] (cm) where the step length is constant at 50 (cm) (see Fig. 7.7b)
- Changing step length and step height at the same time where the range of step length and step height are [30:80] (cm) and [-30:30] (cm) respectively (see Fig. 7.7c)
- Planar version of the W-Prize terrain [1] (see Fig. 7.7c)



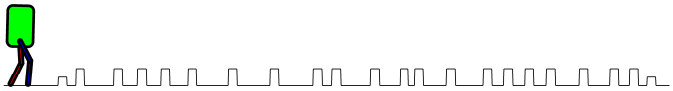
(a) Changing Step Length



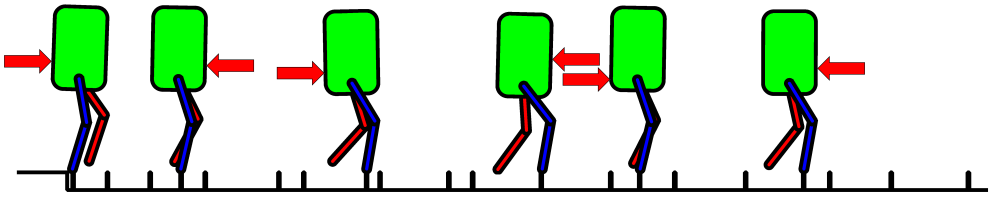
(b) Changing Step Height



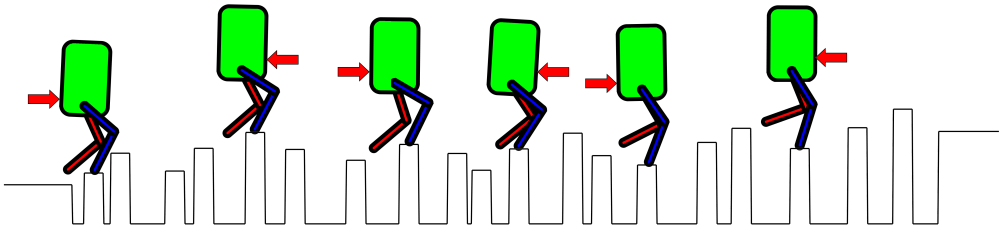
(c) Changing both Step Length and Step Height



(d) Planar Version of the W-Prize Terrain



(e) Changing Step Length with Perturbation



(f) Changing both Step Length an Step Height with Perturbation

Figure 7.7: The problem of dynamically walking over a randomly generated set of discrete footholds. Simulation video: https://youtu.be/Pxhb4_ojiC8.

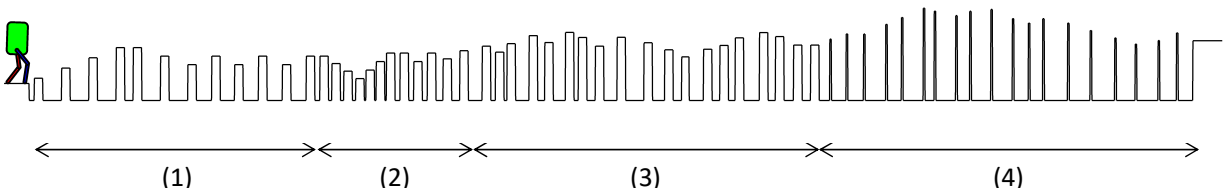


Figure 7.8: ATRIAS walking on different terrains. (1) worst case of walking up and down with large step length. (2) worst case of walking up and down with small step length. (3) 20 walking steps over randomly generated terrain with stone size of 25 (cm) and then. (4) 20 walking steps over randomly generated terrain with stone size of 5 (cm). Simulation video: https://youtu.be/Pxhb4_ojiC8.

- Changing step length with perturbation (see Fig. 7.7e), where the perturbation is generated by applying horizontal external force of $\pm 300N$ on the robot torso with the duration of $0.2s$ in the middle of some steps.
- Changing step length and step height with perturbation (see Fig. 7.7f), with similar type of perturbation mentioned above but the magnitude of the external force is now $200N$.

In all simulations, we check constraints on footstep placement, friction constraints and input saturation stated in Table 3.1. Note that friction constraints are checked for both impulse at impact and contact force during the continuous phase. Fig. 7.9 shows the satisfactions of those constraints in one example of ATRIAS walking on randomly generated discrete footholds shown in Fig. 7.7c, where step length and step height are varied in the range of [30:80] (cm) and [-30:30] (cm) respectively. In this simulation the absolute error between the desired step length and the real step length has the minimum of $0\ m$, the maximum of $0.0453\ m$ and the mean of $0.0113\ m$.

With the problem of changing step length only, we also compare the performance of (a) our prior work on Control Barrier Functions and gait library presented in [55] and (b) our proposed method of the 2-step periodic gait library. Both controllers are run on the same terrain illustrated in Fig. 7.7a. From Fig. 7.10, we can clearly see that our proposed method of 2-step periodic gait library (thick red line) has vertical ground reaction force (F^v_{st}) with smaller peak amplitudes, coefficient $k = |F^h_{vt}/F^v_{vt}|$ that stays more inside the friction cone, and control inputs with smaller norm. Note that although these two controllers are applied on the same terrain, the walking step times are different because the gait libraries and the low-level controllers are different.

In Fig. 7.8, we demonstrate the robot walking over multiple terrains including:

(1) Worst case of walking up and down with large step length,

(2) Worst case of walking up and down with small step length,

(3) 20 steps walking over randomly generated terrain with stone size of 25 (cm),

(4) 20 steps walking over randomly generated terrain with stone size of 5 (cm).

Note that we use the same controller and the same gait library for all these different terrains, thereby establishing that our single controller can handle different types of variation in the terrain, including step length, step height, as well as stone size. In particular, with Terrain (4), the random terrain with stone size of 5 cm, we show the accuracy of the precise footstep placements. With Terrain (3), the random terrain with stone size of 25 cm, we show that the robot has a good swing foot clearance to avoid the corners of the larger stepping stones.

Here, we also sucessfully applied our proposed control method for the planar version of the W-Prize terrain listed in [1], which is made from placing cinder blocks with distances varied in $[35 : 98](cm)$. There are also stepping-up and stepping-down stones at the start and the end of the terrain (see Fig. 7.7d). Note that, in the simulation, all the physical constraints are checked except the constraint on avoiding the cinder blocks from tipping over. This additional challenge of the blocks tipping over is not addressed in this work but it is an interesting problem to consider in future work.

7.3 Experimental Validation of 2-Step Periodic Gait Library

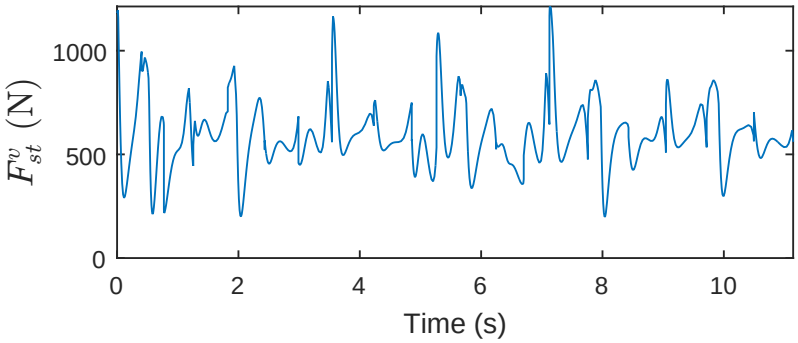
Having validated the controller in simulation, we now show the implementation of the proposed method on the robot hardware. We begin with a brief description of the robot hardware, followed by some implementation details. We then present the results for the following experiments:

- Changing step length within the range of $[23 : 78]$ cm.
- Changing both step length and step height in the range of $[35 : 60]$ cm and $[-22 : 22]$ cm respectively.

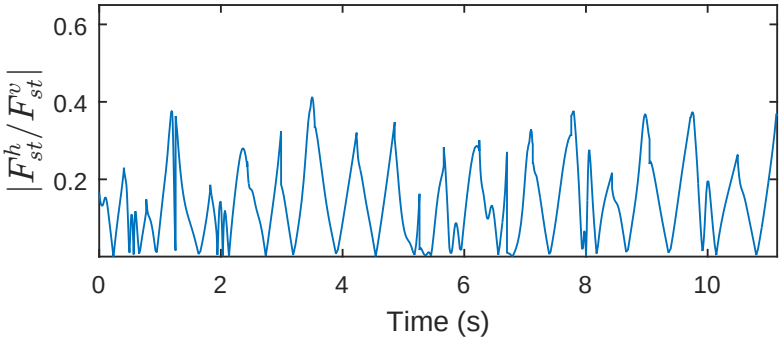
Remark 7.6. *To build the required terrain in experiments, we stack a number of cinder blocks to change the terrain height. In doing so, an additional complexity arises in that these cinder blocks can topple over.*

7.3.1 Hardware Description

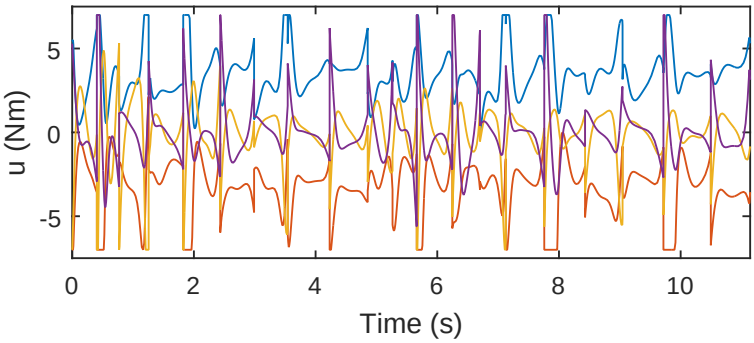
ATRIAS is a 3D capable bipedal robot, equipped with three actuators on each leg – two motors ("leg motors") controlling the motion in the sagittal plane (q_1 and q_2 in Fig. 3.2) and one motor ("hip motor") controlling the lateral angle in the frontal plane (see Fig. 3.2). Therefore, to restrict the motion in the saggital plane, ATRIAS is attached to a boom (see Fig. 7.11b). The two leg motors in the saggital plane, connected to the front and rear thigh links through a 50:1 harmonic drive transmission and fiberglass plate springs, can be controlled to cause an effective change in leg length and/or leg angle. Further, in addition to the encoders



(a) Ground Reaction Force

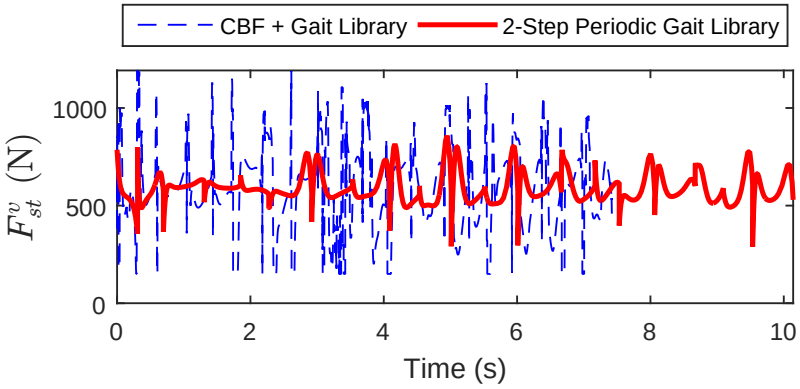


(b) Friction Cone

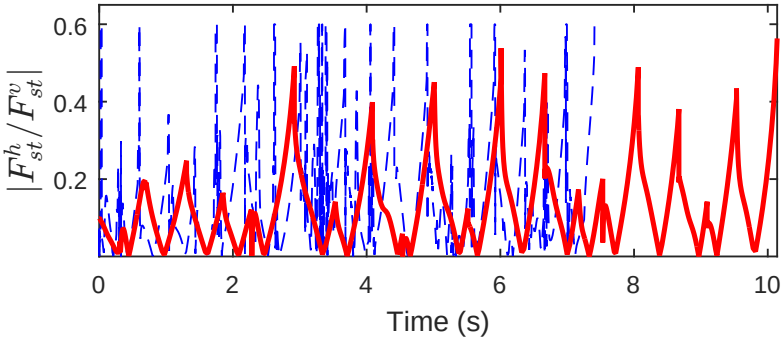


(c) Control Inputs

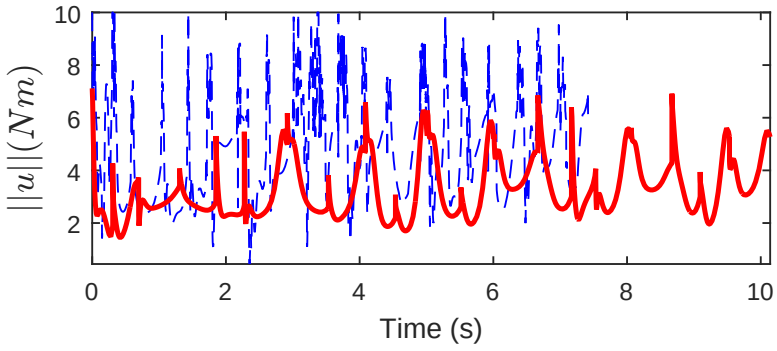
Figure 7.9: Simulation of ATRIAS walking on randomly generated stepping stones with step length and step height changing in the range of [30:80] (cm) and [-30:30] (cm) respectively. The terrain is illustrated in Fig. 7.7c. The following constraints are enforced: (a) Ground reaction force: $F_{st}^v \geq 150(N)$; (b) Friction cone: $|F_{st}^h/F_{st}^v| \leq 0.6$; and (c) Control motor inputs saturated at 7 (Nm) ($|u| \leq 7$). Note that there is a 50:1 gear ratio from the motors to the links.



(a) Ground Reaction Force



(b) Friction Cone



(c) Norm of Control Inputs

Figure 7.10: We compare the “CBF + Gait Library" controller from [55] with the proposed “2-Step Periodic Gait Library" controller (thick red line). As is seen, the proposed controller has better ground reaction force with smaller peak amplitudes, lower friction requirements, as well as smaller control inputs. The comparison is made by simulating both controllers for the same terrain illustrated in Fig. 7.7a.

(a) Stepping stones with step length variation.

(b) Cinder block terrain with step length variation.

(c) Stepping stones with step length and step height variation.

Figure 7.11: ATRIAS walking over stepping stones. Experimental video: https://youtu.be/JKkPWHm6H7k https://youtu.be/jQeC1OOmOmk. Simulation video: https://youtu.be/Pxhb4_ojiC8.

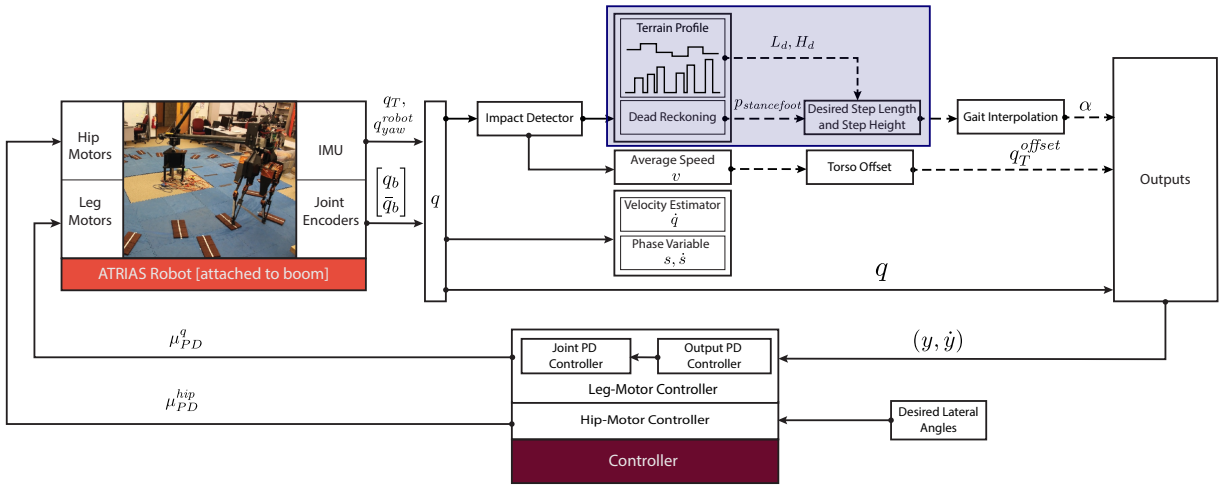


Figure 7.12: Control Diagram of the implemented controller in experiment. Dashed lines denote discrete-time signals sampled at the beginning of every walking step. Continuous lines denote signals sampled at 1 kHz. The shaded blue block illustrates the computation of the desired step length and step height. In particular, we compute the desired step length and height based on the next step location obtained from the terrain profile and the current stance foot position, obtained from the robot yaw and forward kinematics (7.2). This entire block can essentially be replaced by a visual sensor.

Table 7.1: Various measured quantities. The index $i \in \{st, sw\}$ represents stance/swing leg respectively.

Measured Quantity	Symbol	Sensor
Joint angles on motor side (i.e. at the output of the gears)	q_1^i, q_2^i	32-bit absolute optical encoders
Joint angles on leg side (i.e. at the output of the leaf-springs)	$\bar{q}_1^i$, $\bar{q}_2^i$	32-bit absolute optical encoders
Lateral joint angles (hip abduction)	q_3^i	13-bit absolute magnetic encoder
Torso Pitch	q_T	IMU
Robot Yaw	q_{yaw}^{robot}	IMU

mounted on the motor, ATRIAS is equipped with high-resolution (32-bit) absolute encoders for measuring joint angles and spring deflections. Position and orientation in the world frame are estimated using a high-precision inertial measurement unit (IMU). Table 7.1 lists the various quantities measured. Additionally, we define the body coordinates as $q_b := \left[q_1^{st}, q_2^{st}, q_1^{sw}, q_2^{sw}\right]^T$ and $\bar{q}_b := \left[\bar{q}_1^{st}, \bar{q}_2^{st}, \bar{q}_1^{sw}, \bar{q}_2^{sw}\right]^T$ corresponding to joint angles on the motor side and leg side respectively.

7.3.2 HZD Implementation

The Hybrid Zero Dynamics framework is an elegant and powerful means to design gaits and develop model-based feedback controllers for underactuated dynamic bipedal locomotion. However, implementing the controller developed in Section 3.3 involves overcoming certain challenges.

In this section, we discuss some of the implementation details of the HZD controller on the ATRIAS robot. In particular, the 2D robot model is only used to generate a library of 2-step-periodic gaits. The outputs are then regulated using a PD controller. Fig. 7.12 shows an overview of the implemented controller on ATRIAS.

Phase Variable Computation The phase variable s is computed using the joint angles measured at the output of the springs (i.e. on the leg side as opposed to the joint angles measured on the motor side). This is, in fact, the true phase variable with respect to the robot's base frame attached to the stance foot. Secondly, using joint measurements from the leg side leads to a "less noisy" phase variable. This is due to the fact that, during swing phase, the stance foot is assumed to be pinned to the ground [79], i.e. stance foot is attached to the ground through an ideal revolute joint. This causes the motor side to oscillate (due to the compliant elements) with respect to the leg side and the robot's stance foot. Thus, using the joint angles from the motor side to compute the phase variable causes it to be oscillatory, which leads to oscillations in the desired outputs. This could in turn de-stabilize the system. To remove further noise from the signal, the computed phase variable is passed through a low-pass filter.

Joint Velocity Computation Joint velocities are obtained by first computing the finite difference of the motor encoder readings and then passed through a low-pass filter to remove high-frequency noise.

Dead Reckoning using IMU The position of the robot in the world frame is estimated from a high-precision IMU mounted on the robot's torso. This information is then utilized to correct the relative distance between the robot's current stance foot and the next stepping stone (which is the desired step length input to the controller). In particular, we obtain an estimate of the robot's yaw with respect to an inertial frame. The

robot's yaw, multiplied by the boom length, gives us an estimate of the robot's position,

$$p_{stancefoot} = q_{yaw}^{robot} \times l_{boom} + d_{hip \to stancefoot}, \tag{7.2}$$

where q_{yaw}^{robot} is the yaw angle of the robot measured from the IMU signal, $l_{boom} = 2$ m is the boom length and $d_{hip \to stancefoot}$ is the distance from the hip to stance foot is computed based on the joint encoders. Note that we do not make use of any external sensors (such as a boom encoder or optical motion capture). The location of the stepping stones are precomputed and stored before the start of the experiment. However, only the distance between the current stepping stone and the next is provided to the controller one-step in advance. This distance is found based on where the robot is in the world (computed through IMU dead-reckoning) and corrected based on where the stance foot is on the current stone (computed through joint encoders).

Impact Detection ATRIAS is not equipped with any force sensors or contact switches at the foot. To detect swing foot impacts on the ground, we use the leg axial force estimated by the spring compression (obtained by computing the difference in the encoder readings from the motor and leg sides). In particular, an impact is detected if the axial force as well as the phase variable s defined in (3.4) cross certain thresholds. Specifically, we use a threshold value of 300 N for the axial force and 0.6 for the phase variable (see Fig. 7.13).

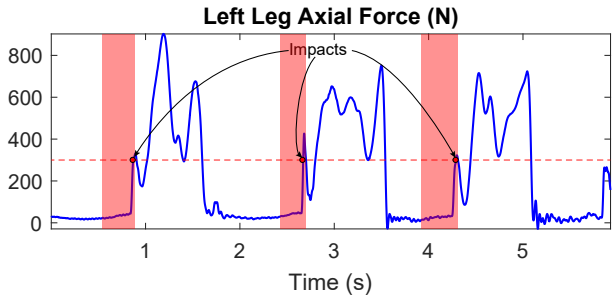


Figure 7.13: Axial Forces on the Left Leg for six walking steps. Shaded red regions denote the periods when the phase variable is greater than the threshold value of 0.6. Dotted red line denotes the threshold value of the axial force at 300 N. An impact is detected when both, the axial force as well as the phase variable, cross their respective thresholds.

Control of Hip Motors For simplicity, the desired lateral angles are kept constant, and enforced through a high gain position-derivative controller,

$$\mu_{PD}^{hip} = -K_p^{hip}\left(q_3 - q_{3,d}\right) - K_d^{hip}\left(\dot{q}_3 - \dot{q}_{3,d}\right), \tag{7.3}$$

where $K_p^{hip},\ K_d^{hip} > 0$.

Walking Speed Regulation We regulate the torso angle to maintain a desired average walking velocity – leaning the torso forward at the start of the swing phase causes the robot to accelerate and leaning the torso backwards causes it to decelerate. Torso angle regulation is achieved by changing the desired leg angles $h_d(s, \alpha)$ in (3.2) appropriately. In particular, at the start of swing phase, a desired torso offset is computed as,

$$q_T^{offset} = -K_p^T(v - v_d), \tag{7.4}$$

where v is the average forward velocity of the robot (estimated from the IMU) at the start of the swing phase, v_d is the desired average walking speed and $K_p^T > 0$. The offset computed above is upper bounded to prevent excessive torso oscillations.

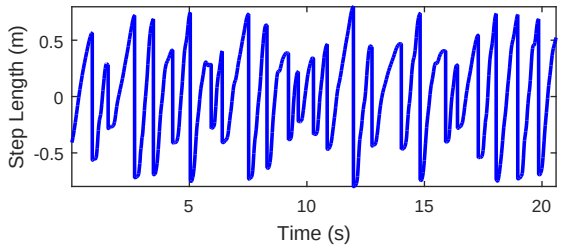
The outputs y are then redefined as,

$$y = h_0(q) - h_d(s, \alpha) - q_T^{offset}\begin{bmatrix}1\\1\\0\\0\end{bmatrix}. \tag{7.5}$$

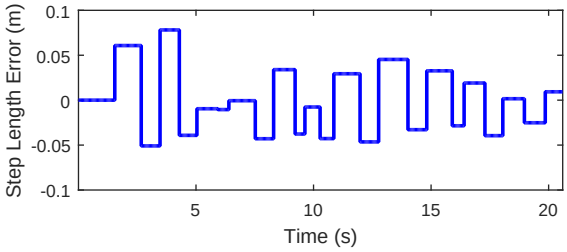
Control of Leg Motors A PD controller is utilized to regulate the outputs in (7.5). While the phase variable is computed using the joint angles on the leg side, the outputs y are computed using the joint angles on the motor side. The leg-motor torques are then computed using this output,

$$\mu_{PD}^{y} = -K_p y - K_d \dot{y} \tag{7.6}$$

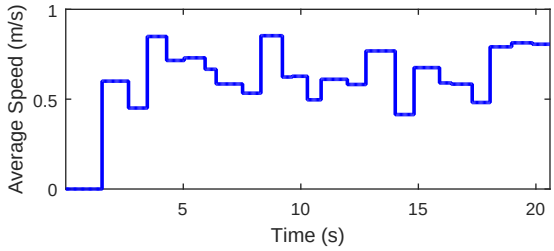
This is done so that the leg motors do not work against the springs. The output PD controller in (7.6) is then mapped to the leg-joint torques using the transformation,



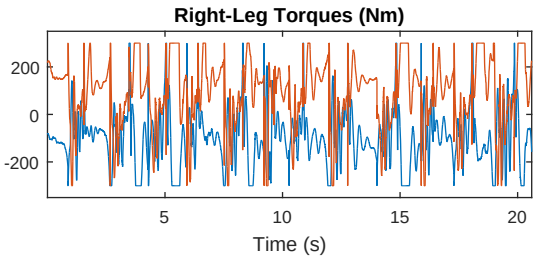
(a) Step Length



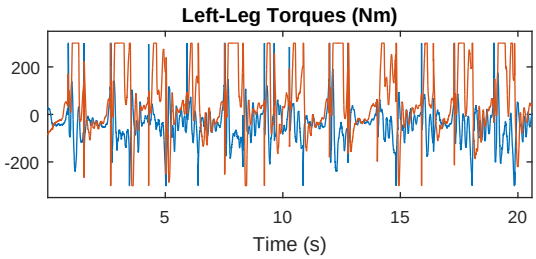
(b) Distance of the Foot from the Center of the Stone



(c) Distance of the Foot from the Center of the Stone



(d) Right-Leg Joint Torques



(e) Left-Leg Joint Torques

Figure 7.14: Results from experiment of ATRIAS walking over stepping stones with changing step length (Fig. 7.11a). Note that there is a 50:1 gear ratio from the motors to the links. We present here the joint torques. A video of the experiment can be found here: https://youtu.be/JKkPWHm6H7k.

$$\mu_{PD}^{q} = h_0^{-1}\left(\mu_{PD}^{y}\right), \tag{7.7}$$

where h_0 is defined in (3.3).

7.3.3 Stepping Stone Experiment with Changing Step Length

In this Section, we will present experimental results on ATRIAS walking over stepping stones (see Fig.7.11). The circle is covered by 24 stepping stones with the following distances or desired step lengths

of the robot:

$$L_d = [56, 31, 64, 78, 33, 75, 30, 40, 72, 67, 35, 23,$$
$$33, 52, 76, 50, 42, 78, 37, 31, 51, 76, 74, 69] \text{ cm}. \tag{7.8}$$

The controller is allowed knowledge of this information only for one step ahead.

Fig. 7.14 shows the experimental data including step length (the horizontal distance between swing and stance foot), the step length error (the error between the stance foot position and the center of the stone), the average speed of walking for each step and the joint torques of the right and left leg respectively. The step length errors have a mean of absolute of 3 cm and absolute max of 7.8 cm and thus are always within the stone size of $[-10 : 10]$ cm. The experiment was conducted with input saturation of 5 Nm on the motor torques. The average walking speed of the total 24 steps is 0.6 m/s. This experiment thus validated the effectiveness of our proposed method for the problem of dynamic walking on stepping stones with a wide range of step length (within [23:78] cm) and a fast average walking speed 0.6 m/s.

To better illustrate the problem, we also have the robot walking on cinder blocks (see Fig. 7.11b), where the height of the platform is 12 cm. Note that for this experiment, since the cinder blocks are not attached to the ground, we need to guarantee very accurate footstep placement and good friction constraint to ensure that the cinder blocks do not topple.

7.3.4 Stepping Stone Experiment with Changing Step Length and Step Height

For this experiment, the terrain was generated by placing 12 stepping stones (see Fig. 7.15) with different distances and heights. The following distances between these stones specify the desired step lengths:

$$L_d = [40, 35, 40, 60, 35, 60, 45, 35, 60, 35, 40, 50] \text{ cm}. \tag{7.9}$$

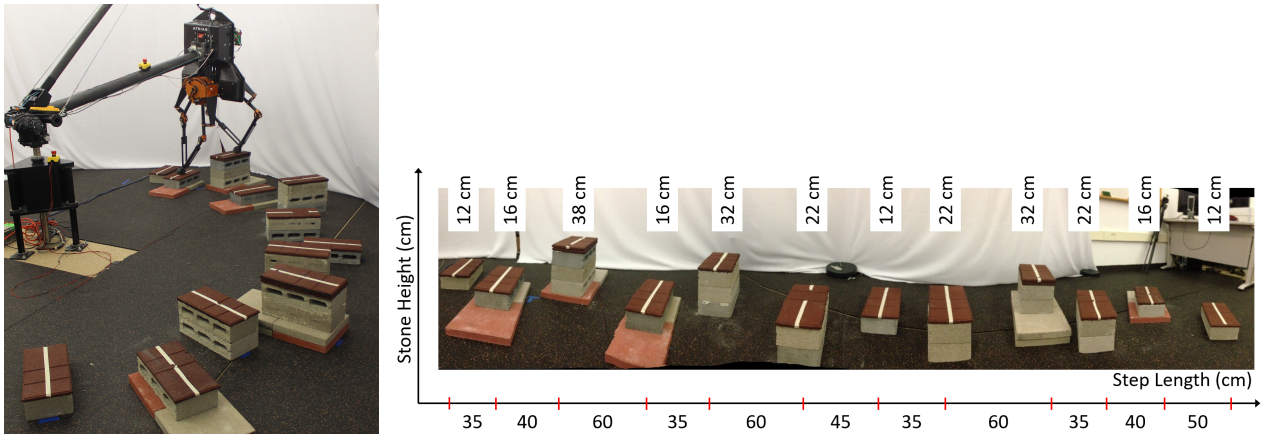
The heights of these stones from the ground are:

$$[12, 16, 38, 16, 32, 22, 12, 22, 32, 22, 16, 0] \text{ cm}, \tag{7.10}$$

that specifies the following step height changes between consecutive stepping stones:

$$H_d = [12, 4, 22, -22, 16, -10, -10, 10, -10, -6, -4, -12] \text{ cm}. \tag{7.11}$$

Similar to the previous problem of changing step length, we also report here in Fig. 7.16 the experiment data including step length (the horizontal distance between swing and stance foot), the step length error (the



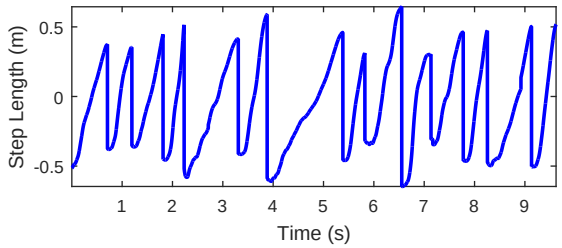
(a) (b) a panorama of the terrain

Figure 7.15: ATRIAS stepping stone experiment with changing step length and step height. (b) is a panorama of the terrain. Also shown are the step lengths between consecutive stones and the heights to the ground of each stones in the illustrated terrain. A video of the experiment can be found here: https://youtu.be/jQeC1OOmOmk

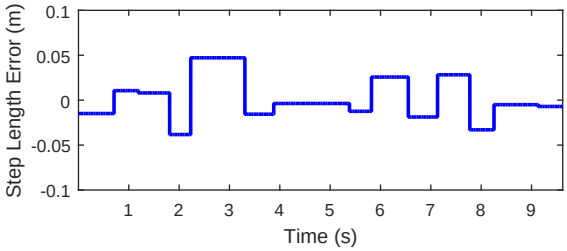
error between the stance foot position and the center of the stone), the average speed of walking for each step and the joint torques of the right and left leg respectively. The step length errors are just within $[-5 : 5]$ cm, therefore always staying inside stepping stones with the size of $[-10 : 10]$ cm. For this experiment, the change in step height is between -22 cm and $+22$ cm and the maximum height of the terrain to the ground is up to 38 cm. The average walking speed of the experiment is 0.6 m/s.

Note that cinder blocks used to create the terrain in Fig. 7.15 are not affixed to each other. Therefore, the accuracy of the footstep placement and good friction constraints are very critical to avoid both sliding and toppling issues in this experiment.

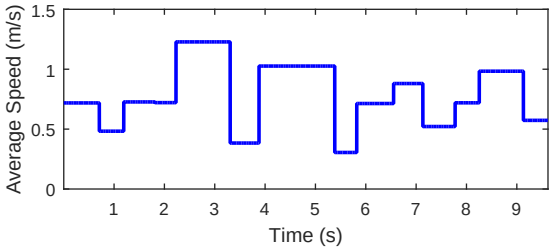
In the experiment shown in Fig.7.15, we tried to place the stepping stones so that we can test as much as possible different type of changes in the terrain. In addition to that, we also demonstrated our method on randomly generated terrains (see Fig. 7.17). For each test, the terrain is created by randomly placing 5 stepping stones, which corresponds to 6 walking steps with variation in both step length and step height. While the desired step lengths are randomly selected in increments of 1 cm, the stone heights are randomly selected from a limited discrete set due to the limitation of stone types used in the experiment. Furthermore, for the random height selection, we enforce the constraint that the starting and ending steps are at zero height so as to start and end the experiment on the ground level.



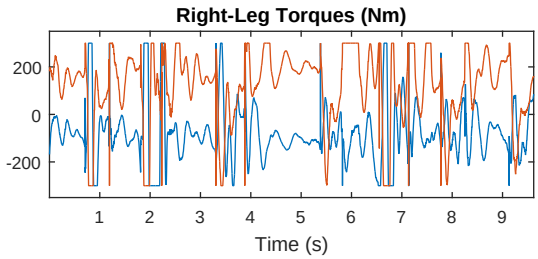
(a) Step Length



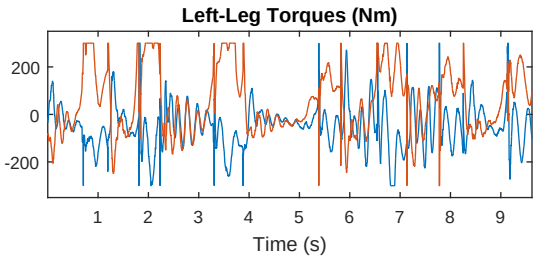
(b) Distance of the Foot from the Center of the Stone



(c) Distance of the Foot from the Center of the Stone



(d) Right-Leg Joint Torques

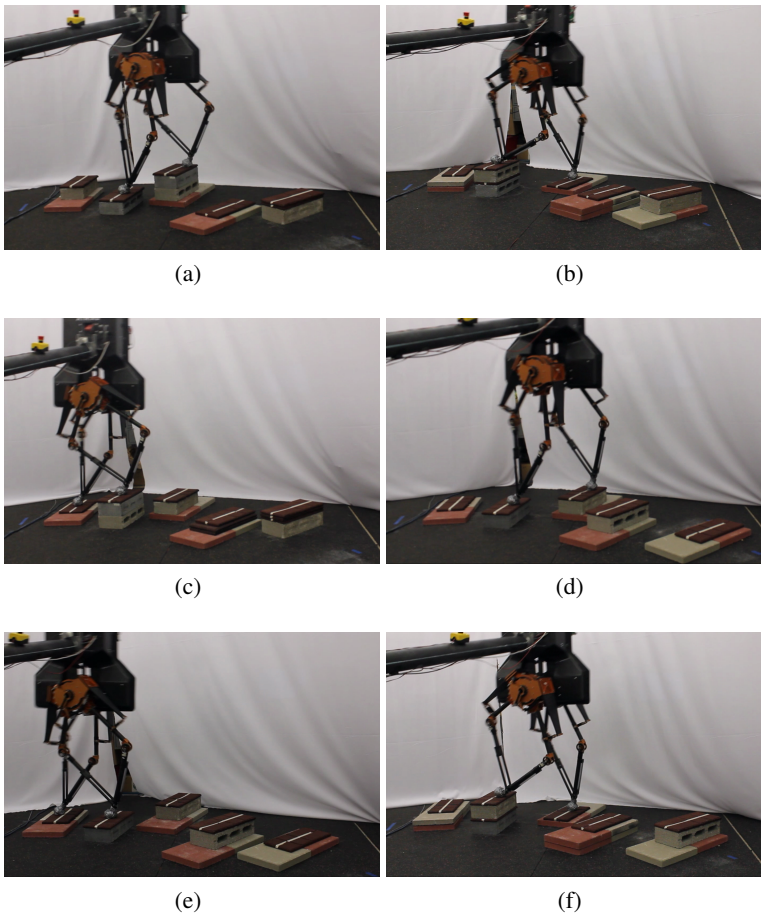


(e) Left-Leg Joint Torques

Figure 7.16: Results from experiment of ATRIAS walking over stepping stones with changing step length and step height (Fig. 7.11c).Note that there is a 50:1 gear ratio from the motors to the links. We present here the joint torques. Experimental video: https://youtu.be/jQeC1OOmOmk.

7.4 Summary

We have presented a novel approach called 2-step periodic gait optimization that allows us to handle a wide change in footstep placement of the robot (both step length and step height). Since our walking gait optimization takes into account not only the upcoming terrain but also the current configuration of the robot, the method can effectively address the transition when the controller switches between different gaits. The gait library is pre-computed with a small number of gaits (4 gaits for the problem of changing step length or step height only and 16 gaits for the problem of changing both), then gait interpolation is implemented in real-time to adapt with random changes in the terrain as well as the initial condition of the robot. We successfully validated the proposed approach on the model of ATRIAS, an underactuated bipedal robot, under different types of terrain, including changing step length in the range of [20:90] (cm) with the



(a) (b)

(c) (d)

(e) (f)

Figure 7.17: ATRIAS walking over different realizations of the stochastic terrain with simultaneous random variation in step length and step height.

precision of only 2 (cm); changing step height only in the range of [-30:30] (cm) or changing both step length and step height in the range of [30:80] (cm) and [-30:30] (cm) respectively. A 2D version of the W-Prize terrain was also attempted. In future work, this method will be extended to 3D walking.

Chapter 8

Optimal Robust Safety-Critical Control

In this chapter, we will introduce a unified control framework to address the problem of robust stability, robust safety and robust constraints under model uncertainty for dynamical systems in general and for bipedal robot in particular. Our proposed optimal robust control through quadratic program offers tracking stability while subject to input and state-based constraints as well as safety-critical constraints for nonlinear dynamical robotic systems under significant model uncertainty. The method formulates robust control Lyapunov and barrier functions to provide guarantees of stability and safety in the presence of model uncertainty. We evaluate our proposed control design on different applications ranging from a single-link pendulum to dynamic walking of bipedal robot subject to contact force constraints as well as safety-critical precise foot placements on stepping stones, all while subject to significant model uncertainty. We conduct preliminary experimental validation of the proposed controller on a rectilinear spring-cart system under different types of model uncertainty and perturbations.

8.1 Adverse Effects of Uncertainty in Dynamics on the CBF-CLF-QP controller

The controller presented in Section 5.1 provide means of an optimal control scheme that balances conflicting requirements between stability, state-based constraints and energy consumption. It is a powerful method that has been deployed successfully for different applications, for example Adaptive Cruise Control [49], quadrotor flight [82], and dynamic walking for bipedal robots [58], [34].

However, a primary disadvantage of the CBF-CLF-QP controller is that it requires the knowledge of an accurate dynamical model of the plant to make guarantees on stability as well as safety. In particular, enforcing the CLF and CBF conditions in the QP (5.7) requires the accurate knowledge of the nonlinear

dynamics $f(x), g(x)$ to compute the Lie derivatives in (3.8), (3.23) and (5.5). Therefore, model uncertainty that is usually present in physical systems can potentially cause poor quality of control leading to tracking errors, and potentially instability [57], as well as violation of the safety-critical constraints [84]. In this section, we will explore the effect of uncertainty on the CLF-QP controller, input and state constraints, and safety constraints enforced by the CBF-QP controller.

The effects of uncertainty on CLFs were presented on Section.4.1.1. We now will introduce the adverse effects of uncertainty on CBFs and constraints.

8.1.1 Effect of Uncertainty on Constraints

As developed in Section 5.1, the input and state constraints can be expressed as (3.32). These constraints depend on the model explicitly and are constraints on the real system dynamics. We can thus rewrite the constraints showing explicit model dependence as,

$$A_c(x, f, g)u \leq b_c(x, f, g). \tag{8.1}$$

If a controller naively enforces these constraints using the nominal model available to the controller, the controller will enforce the constraint

$$A_c(x, \tilde{f}, \tilde{g})u \leq b_c(x, \tilde{f}, \tilde{g}). \tag{8.2}$$

that is different from the desired constraint we want to enforce on the real model (8.1). For instance, to enforce a contact force constraint, if the controller computes and enforces the contact force constraint using the nominal model, there is absolutely no guarantee that the actual contact force on the real system satisfies the constraint.

Remark 8.1. *It must be noted that certain constraints do not depend on the model at all. In such cases, model uncertainty does not affect the constraint. One example of such a constraint is a pure input constraint, such as $u(x) \leq u_{max}$. Expressing this constraint in the form of* (8.1) *results in $A_c = I, b_c = u_{max}$, which clearly is not dependent on the model.*

8.1.2 Effects of Uncertainty on CBFs

Similar to what we have seen about the effect of uncertainty on CLFs and constraints, we will now see the effect of uncertainty on CBFs. We note that the time-derivative of the Barrier function in (5.6) depends

on the real model. Therefore we need to enforce the following constraint given by (5.3):

$$\dot{B}(x, f, g, u) = \frac{\partial B}{\partial x}(f(x) + g(x)u) \leq \frac{\gamma}{B(x)} \tag{8.3}$$

where $\dot{x} = f(x) + g(x)u$ is the real system dynamic. As seen in the case of control Lyapunov functions and constraints, naively enforcing this barrier constraint using the nominal model results in,

$$\frac{\partial B}{\partial x}(\tilde{f}(x) + \tilde{g}(x)u) \leq \frac{\gamma}{B(x)} \tag{8.4}$$

where $\dot{x} = \tilde{f}(x) + \tilde{g}(x)u$ is the nominal system dynamic known by the controller. Clearly due to model uncertainty, or the significant difference between $(f(x), g(x))$ and $(\tilde{f}(x), \tilde{g}(x))$, this constraint is different from the previous one. In fact, as analyzed in [84], this results in violation of the safety-critical constraint established by the Barrier function.

Remark 8.2. *With the CLF, the continuous-dynamics of* (3.13) *is a linear time-invariant system (F and G are constant matrices), and thus we can evaluate the effect of uncertainty based on the difference with the static nominal model of F and G to obtain* (4.5)*. However, in the case of constraints* (8.2) *and CBFs* (8.4)*, the relation with the control input is not static (nonlinearly dependent on the system state x), and thus the same approach to analyze and explore the effect of model uncertainty as on CLFs could encounter difficulties. In order to solve this issue, we will introduce an alternative design of Virtual Input-Output Linearization that will help us to develop a novel method of Robust CBF-CLF-QP in the next section.*

8.2 Proposed Robust Control based Quadratic Programs

We have presented Robust CLF-QP controller to address the problem of robust stability in Section. 4.1. We now will introduce the notion of virtual input-output linearizaion to help to extend the method to the problem of robust safety and robust constraints under model uncertainty.

8.2.1 Virtual Input-Ouput Linearization

As mentioned earlier, our development of the robust CLF-QP controller does not enable us to correctly enforce constraints in the presence of model uncertainty. This problem also exists for CBF constraints, preventing us from robustifying input and state constraints, as well as safety constraints in the form of barrier constraints. On the other hand, robustifying the CLF-QP condition was easier. The form of the Input-Output Linearization under the presence of model uncertainty (4.3) was essential for "robustifying" the CLF-QP condition (see Section 4.1). As mentioned in Remark 8.2, because the effect of uncertainty for

the CBF constraints with respect to control input u is nonlinear and state-dependent, the extension to robust CBF and robust constraints is not straightforward.

One idea to address this is to get the CBF time-derivative in (5.4) to the same form as the CLF time-derivative through another input-output linearization for the CBF time-derivative. This will robustify CBFs and enable creating a unified robust CBF-CLF-QP. However this idea is not directly feasible since $L_gB(x)$ in (5.4) is a row vector and obviously not invertible. One approach that could have worked for this is to use MIMO non-square input-output linearization [40]. However, here we solve this problem through optimization by introducing a new notion of Virtual Input-Output Linearization (VIOL), wherein an invertible decoupling matrix is not required.

We first explain our method for CBF constraints (a similar process can be carried out for input and state constraints). We begin with the CBF $B(x)$ and define a virtual control input $\mu_b \in \mathbb{R}$ such that the time derivative of the CBF is

$$\dot{B}(x,u) = L_fB(x) + L_gB(x)u =: \mu_b. \tag{8.5}$$

Then, we can write the time-derivative of the CBF in terms of μ_b as,

$$\dot{B}(x,\mu_b) = \mu_b,$$

such that the CBF condition (5.3) simply becomes

$$\dot{B}(x,\mu_b) - \frac{\gamma}{B(x)} \leq 0. \tag{8.6}$$

We can then reformulate the CBF-CLF-QP from (5.7) in terms of (8.5) and (8.6) as,

$$\begin{aligned}
u^*(x) = \underset{u,\mu,\mu_b,d}{\operatorname{argmin}} \quad & \mu^T\mu + p\, d^2 && (8.7)\\
\text{s.t.} \quad & \dot{V}_\varepsilon(\eta,\mu) + \frac{c_3}{\varepsilon}V_\varepsilon(\eta) \leq d, && \textbf{(CLF)}\\
& \dot{B}(x,\mu_b) - \frac{\gamma}{B(x)} \leq 0, && \textbf{(CBF)}\\
& L_fB(x) + L_gB(x)u = \mu_b, && \textbf{(VIOL)}\\
& u = u_{ff}(x) + (L_gL_fy(x))^{-1}\mu. && \textbf{(IO)}
\end{aligned}$$

Here, the equality constraint $L_fB(x) + L_gB(x)u = \mu_b$ of the Virtual Input-Ouput Linearization (VIOL) is presented in (8.5). Note that the solutions of the two CBF-CLF-QP controllers in (5.7) and (8.7) are identical. However, the VIOL in the CBF-CLF-QP opens up a very effective and systematic way to design

a robust version of the CBF-CLF-QP controller that will be introduced next.

Remark 8.3. *Note that the virtual input-output linearization of $\dot{B}$ does not actually give a least squares solution for u corresponding to $u = L_g B^\dagger(x)(\mu_b - L_f B(x))$, with $L_g B^\dagger(x)$ being the pseudo inverse of $L_g B(x)$. This is because the optimization solves for both u and μ_b simultaneously, and is subject to several other constraints.*

8.2.2 Robust CBF-CLF-QP

By using VIOL, the CBF now takes the form of a linear system, $\dot{B} = \mu_b$, and therefore the effect of uncertainty can be easily extended by using the same approach as with the robust CLF-QP. In particular, with model uncertainty we have,

$$\begin{aligned}\dot{B}(x, \Delta_1^b, \Delta_2^b, \mu_b) &= L_f B + L_g B u \\ &= \mu_b + \Delta_1^b + \Delta_2^b \mu_b, \end{aligned} \tag{8.8}$$

where the last equality follows from substituting for u from (4.1) with $\Delta_1^b, \Delta_2^b \in \mathbb{R}$. We next develop the exact expressions for Δ_1^b, Δ_2^b as was done for the case of CLFs in (4.4).

To be more specific, because we need to compute the VIOL based on the nominal model, we will have

$$L_{\tilde{f}} B + L_{\tilde{g}} B u = \mu_b. \tag{8.9}$$

From (8.8) and (8.9),

$$\begin{aligned} L_g B\, u &= \mu_b + \Delta_1^b + \Delta_2^b \mu_b - L_f B \\ L_{\tilde{g}} B\, u &= \mu_b - L_{\tilde{f}} B \end{aligned}$$

$$\begin{aligned} \implies u &= L_g B^\dagger(\mu_b + \Delta_1^b + \Delta_2^b \mu_b - L_f B) \\ &= L_{\tilde{g}} B^\dagger(\mu_b - L_{\tilde{f}} B), \end{aligned}$$

where $L_g B^\dagger(x), L_{\tilde{g}} B^\dagger(x)$ are the pseudo inverse of $L_g B(x)$ and $L_{\tilde{g}} B(x)$ respectively. We therefore have:

$$\mu_b(L_g B^\dagger \Delta_2^b + L_g B^\dagger - L_{\tilde{g}} B^\dagger) + (L_g B^\dagger \Delta_1^b - L_g B^\dagger L_f B + L_{\tilde{g}} B^\dagger L_{\tilde{f}} B) = 0\ (\forall \mu_b). \tag{8.10}$$

Since the equation (8.10) holds for all μ_b, we can then represent Δ_1^b, Δ_2^b as solutions of following equations:

$$\implies \begin{cases} L_g B^\dagger \Delta_2^b = L_{\tilde{g}} B^\dagger - L_g B^\dagger, \\ L_g B^\dagger \Delta_1^b = L_g B^\dagger L_f B - L_{\tilde{g}} B^\dagger L_{\tilde{f}} B. \end{cases} \tag{8.11}$$

Remark 8.4. *In the expressions for Δ_1^b, Δ_2^b, note that when there is no model uncertainty, i.e., $\tilde{f} = f, \tilde{g} = g$, then $\Delta_1^b = \Delta_2^b = 0$.*

The CBF condition (8.6) under model uncertainty becomes

$$\dot{B}(x, \Delta_1^b, \Delta_2^b, \mu_b) - \frac{\gamma}{B(x)} \leq 0. \tag{8.12}$$

Because we developed the CLF and CBF having the similar form of linearized system, we now have a systematic way to develop the Robust CBF-CLF-QP. Again, we will assume that our model uncertainty is bounded, i.e.,

$$||\Delta_1^b|| \leq \Delta_{1,max}^b, \qquad ||\Delta_2^b|| \leq \Delta_{2,max}^b. \tag{8.13}$$

Then, we will have the robust CBF condition based on this assumption as,

$$\max_{\substack{||\Delta_1^b|| \leq \Delta_{1,max}^b \\ ||\Delta_2^b|| \leq \Delta_{2,max}^b}} \dot{B}(x, \Delta_1^b, \Delta_2^b, \mu_b) - \frac{\gamma}{B(x)} \leq 0. \tag{8.14}$$

Note that choosing μ_b that satisfies (8.14) implies $x \in \mathcal{C}$ or $B(x) \geq 0$ for every Δ_1^b, Δ_2^b satisfying $||\Delta_1^b|| \leq \Delta_{1,max}^b, ||\Delta_2^b|| \leq \Delta_{2,max}^b$. Note that, $\mathcal{C}$ is as defined in (5.1) and $B(x)$ is as defined in (5.6).

We can then express (8.14) as the following QP:

Robust CBF-CLF-QP:

$$
\begin{aligned}
u^*(x) = \underset{u,\mu,\mu_b,\delta}{\operatorname{argmin}} \quad & \mu^T\mu + p\delta^2 && (8.15)\\
\text{s.t.} \quad & \max_{\substack{||\Delta_1||\le\Delta_1^{max}\\ ||\Delta_2||\le\Delta_2^{max}}} \dot{V}_\varepsilon(\eta,\Delta_1,\Delta_2,\mu) + \frac{c_3}{\varepsilon}V_\varepsilon \le \delta, \\
& && \textbf{(Robust CLF)}\\
& \max_{\substack{||\Delta_1^b||\le\Delta_{1,max}^b\\ ||\Delta_2^b||\le\Delta_{2,max}^b}} \dot{B}(x,\Delta_1^b,\Delta_2^b,\mu_b) - \frac{\gamma}{B(x)} \le 0, \\
& L_{\tilde{f}}B(x) + L_{\tilde{g}}B(x)u = \mu_b, && \textbf{(Robust CBF)}\\
& u = \tilde{u}_{ff}(x) + (L_{\tilde{g}}L_{\tilde{f}}y(x))^{-1}\mu. && \textbf{(IO)}
\end{aligned}
$$

Having robustified CLFs and CBFs, we will now apply this to obtain robust constraints.

Remark 8.5. *Note that, as illustrated in Appendix A.3.1, the max CBF inequality condition in the above Robust CBF-CLF-QP can be converted to two simple linear inequality constraints, thus ensuring the above min-max problem is actually a quadratic program.*

8.2.3 Robust CLF-QP with Robust Constraints

To robustify the "constraints", we can use a similar method as we did for the control barrier functions. We start by rewriting the constraints from (3.32) as $A_c(x)u - b_c \le 0$. We then reformulate the constraints by using Virtual Input-Output Linearization (VIOL) to obtain,

$$A_c(x)u - b_c(x) =: \mu_c, \tag{8.16}$$

and then enforce

$$\mu_c \le 0. \tag{8.17}$$

We can then reformulate the CLF-QP with constraints from (3.33) in terms of (8.16) and (8.17) as,

$$
\begin{aligned}
u^*(x) = & \underset{u,\mu,\mu_c,\delta}{\operatorname{argmin}} && \mu^T\mu + p\,\delta^2 && (8.18)\\
& \text{s.t.} && \dot{V}_\varepsilon(\eta,\mu) + \frac{c_3}{\varepsilon}V_\varepsilon(\eta) \le \delta, && (\textbf{CLF})\\
& && \mu_c \le 0, && (\textbf{Constraints})\\
& && A_c(x)u - b_c(x) = \mu_c, && (\textbf{VIOL})\\
& && u = \tilde{u}_{ff}(x) + (L_{\tilde{g}}L_{\tilde{f}}y(x))^{-1}\mu. && (\textbf{IO})
\end{aligned}
$$

The above QP is equivalent to (3.33), however, the reformulation of the constraints enables us to easily consider model uncertainty. With model uncertainty, we now have,

$$A_c(x)u - b_c(x) = \mu_c + \Delta_1^c + \Delta_2^c\mu_c =: \bar{\mu}_c(\mu_c, \Delta_1^c, \Delta_2^c), \tag{8.19}$$

with the constraint becoming,

$$\bar{\mu}_c(\mu_c, \Delta_1^c, \Delta_2^c) \le 0. \tag{8.20}$$

Once again, we make the assumption on bounded uncertainty,

$$||\Delta_1^c|| \le \Delta_{1,max}^c, \qquad ||\Delta_2^c|| \le \Delta_{2,max}^c, \tag{8.21}$$

such that the robust constraint condition becomes,

$$\max_{\substack{||\Delta_1^c|| \le \Delta_{1,max}^c \\ ||\Delta_2^c|| \le \Delta_{2,max}^c}} \bar{\mu}_c(\mu_c, \Delta_1^c, \Delta_2^c) \le 0. \tag{8.22}$$

Incorporating this with the robust CLF-QP we have the following robust CLF-QP with robust constraints:

Robust CLF-QP with Robust Constraints:

$$
\begin{aligned}
u^*(x) = & \underset{u,\mu,\mu_c,\delta}{\operatorname{argmin}} && \mu^T\mu + p\delta^2 && (8.23)\\
& \text{s.t.} && \max_{\substack{||\Delta_1||\le\Delta_1^{max}\\ ||\Delta_2||\le\Delta_2^{max}}} \dot{V}_\varepsilon(\eta,\Delta_1,\Delta_2,\mu) + \frac{c_3}{\varepsilon}V_\varepsilon \le \delta, \\
& && && \textbf{(Robust CLF)}\\
& && \max_{\substack{||\Delta_1^c||\le\Delta_{1,max}^c\\ ||\Delta_2^c||\le\Delta_{2,max}^c}} \bar{\mu}_c(\mu_c,\Delta_1^c,\Delta_2^c) \le 0, \\
& && A_c(x)u - b_c(x) = \mu_c, && \textbf{(Robust Constraints)}\\
& && u = \tilde{u}_{ff}(x) + (L_{\tilde{g}}L_{\tilde{f}}y(x))^{-1}\mu. && \textbf{(IO)}
\end{aligned}
$$

Remark 8.6. *As illustrated in Appendix A.4, the max constraints inequality condition in the above Robust CLF-QP with Robust Constraints can also be converted to two simple linear inequality constraints, thus ensuring the above min-max problem is actually a quadratic program.*

8.2.4 Robust CBF-CLF-QP with Robust Constraints

We finally unify the the robust CLF for stability under model uncertainty, robust CBF for safety enfrorcement under model uncertainty, and the robust constraints to obtain the following unified robust controller.

QP-based controller	Equation
CLF-QP	(3.27)
CLF-QP with constraints	(3.33)
CBF-CLF-QP	(5.7)
CBF-CLF-QP with constraints	(5.8)
Robust CLF-QP	(4.12)
Robust CLF-QP with constraints	(4.13)
Robust CBF-CLF-QP	(8.15)
Robust CLF-QP with robust constraints	(8.23)
Robust CBF-CLF-QP with robust constraints	(8.24)

Table 8.1: A list of all the various QP-based controllers presented in this Section.

Robust CBF-CLF-QP with Robust Constraints:

$$
\begin{aligned}
u^*(x) = \quad & && (8.24)\\
\underset{u,\mu,\mu_b,\mu_c,\delta}{\operatorname{argmin}} \quad & \mu^T\mu + p\delta^2 && \\
\text{s.t.} \quad & \max_{\substack{||\Delta_1||\leq\Delta_1^{max}\\ ||\Delta_2||\leq\Delta_2^{max}}} \dot{V}_\varepsilon(\eta,\Delta_1,\Delta_2,\mu) + \frac{c_3}{\varepsilon}V_\varepsilon \leq \delta, && \\
& && \textbf{(Robust CLF)}\\
& \max_{\substack{||\Delta_1^b||\leq\Delta_{1,max}^b\\ ||\Delta_2^b||\leq\Delta_{2,max}^b}} \dot{B}(x,\Delta_1^b,\Delta_2^b,\mu_b) - \frac{\gamma}{B(x)} \leq 0, && \\
& L_{\tilde{f}}B(x) + L_{\tilde{g}}B(x)u = \mu_b, && \textbf{(Robust CBF)}\\
& \max_{\substack{||\Delta_1^c||\leq\Delta_{1,max}^c\\ ||\Delta_2^c||\leq\Delta_{2,max}^c}} \bar{\mu}_c(\mu_c,\Delta_1^c,\Delta_2^c) \leq 0, && \\
& A_c(x)\mu - b_c(x) = \mu_c, && \textbf{(Robust Constraints)}\\
& u = \tilde{u}_{ff}(x) + (L_{\tilde{g}}L_{\tilde{f}}y(x))^{-1}\mu. && \textbf{(IO)}
\end{aligned}
$$

A summary of the various QP-based controllers presented in this Section are presented in Table 8.1.

Having presented our proposed optimal robust controller that can address stability and strictly enforce constraints under model uncertainty, we now validate our controller in simulations on a single pendulum and bipedal robot systems and experiments on spring cart system.

8.3 Simulation Validation of Robust CBF

8.3.1 Single Pendulum

We first evaluate our proposed controller on a classical engineering example of single pendulum. The control goal in to drive the pendulum angle from the initial condition $q_0 = \pi/4(rad)$ to the origin $q_f = 0$. Let's assume that the pendulum is connected to a vertical wall and therefore, the pendulum angle is limited by $q_{min} = -0.02(rad)$. We compare three different controllers:

$$\begin{cases} \text{I: CLF-QP} \\ \text{II: CBF-CLF-QP} \\ \text{III: Robust CBF-CLF-QP} \end{cases}$$

in which, the CLF is to drive $q \to q_f$ and the CBF is to guarantee the safety constraint $q \geq q_{min}$.

In order to generate uncertainty for the pendulum model, we will multiply the mass and the length of the pendulum by a certain scale:

$$\begin{cases} \text{Case 1: model scale =1 (NO Uncertainty)} \\ \text{Case 2: model scale =1.2} \\ \text{Case 3: model scale =2} \end{cases}$$

The simulation result is shown in the Fig.8.1. As seen in this figure, our proposed controller of robust CBF-CLF-QP is able to guarantee both good tracking while strictly enforcing the safety constraint ($q \geq q_{min}$) even with 100% of model uncertainty in Case 3.

8.3.2 RABBIT Bipedal Robot

To demonstrate the effectiveness of the proposed robust CBF-CLF-QP controller, we will conduct numerical simulations on the model of RABBIT that was described in Section. 4.1.4. We consider model uncertainty in bipedal robotic walking by adding an unknown heavy load to the torso of the RABBIT robot to validate the performance of our proposed robust controllers. We will also require enforcement of contact force constraints (state constraints) and foot-step location constraints (safety constraints) in the presence of the model uncertainty.

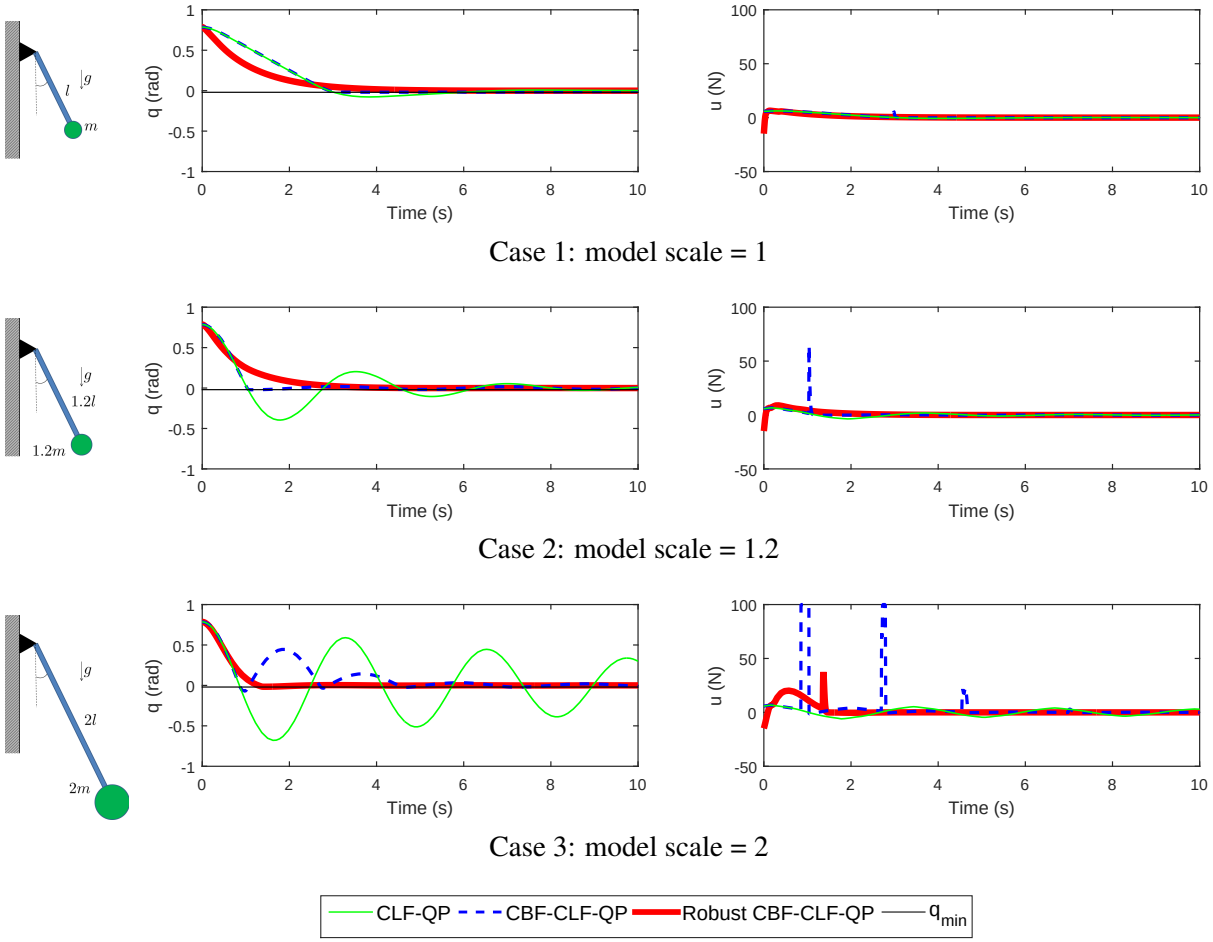


Figure 8.1: Single Pendulum: Control of single pendulum with safety constraint $q \geq q_{min}$. The CBF-CLF-QP can guarantee safety with a perfect model but fails with only 20% of model uncertainty. The Robust CBF-CLF-QP maintains both very good tracking performance and safety constraints under up to 100% of model uncertainty. Note that the non-robust CBF-CLF-QP controller has sudden jumps in the control signal due to the unknown model dynamics causing the state going fast to the boundary of the safe set. In contrast, the robust CBF-CLF-QP controller takes the uncertainty into account and only lets the state approach the boundary of the safe set slowly.

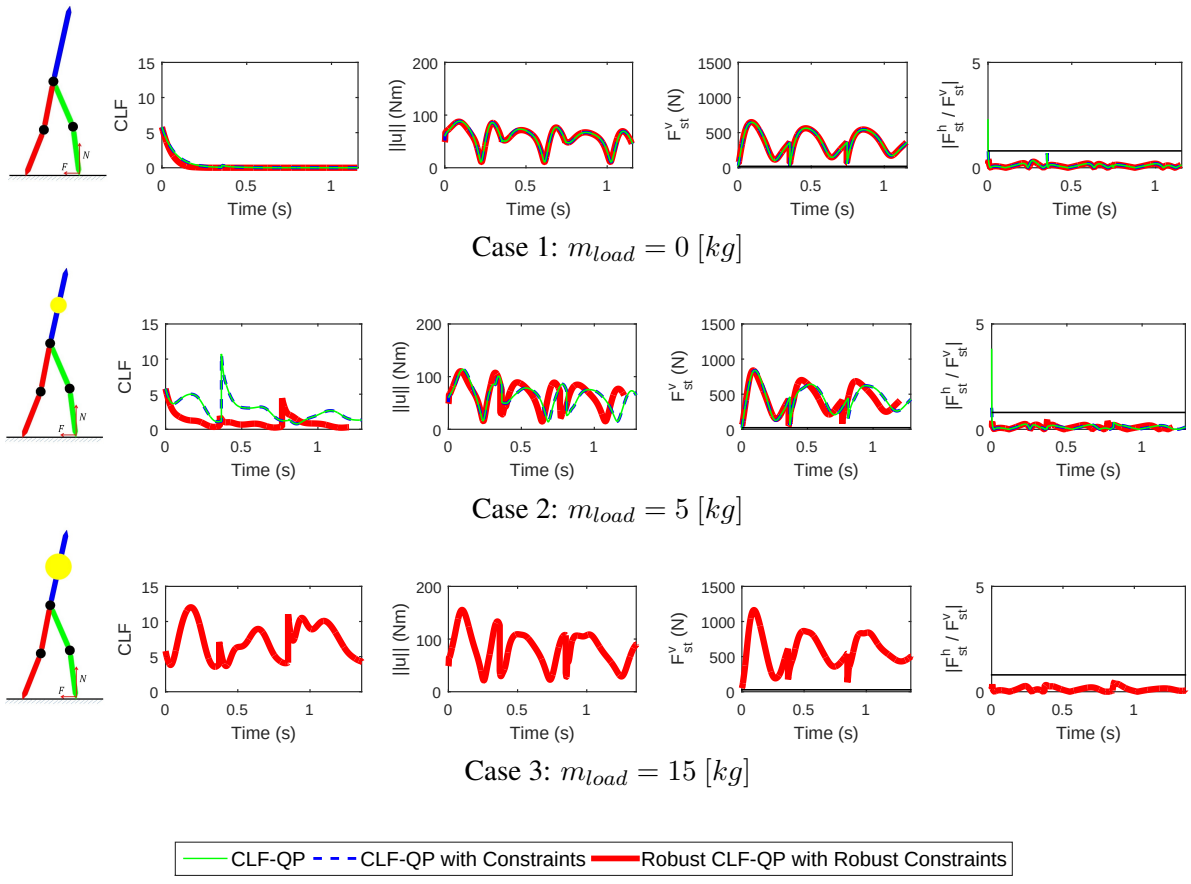


Figure 8.2: Bipedal robot with carrying unknown load and contact force constraints. Even in the nominal case of NO uncertainty, the CLF-QP controller fails due to lacking of contact force constraints. The CLF-QP with Constraints (Contact Force Constraints) works well with perfect model but fails with only 5 kg of load. The Robust CLF-QP with Robust Constraints maintains both good tracking performance and contact force constraints under up to 15 kg of load (47% of the robot weight). The other two controllers are unstable in this case.

8.3.3 Dynamic Walking of Bipedal Robot while Carrying Unknown Load, subject to Contact Force Constraints

The problem of contact force constraints is very important for robotic walking, as any violation of contact constraints would cause the robot to slip and fall. We design our nominal walking gait so as to satisfy contact constraints. However, we cannot guarantee this constraint once there is a perturbation from the nominal walking gait. Although the feedback controller (for example CLF-QP), will drive any error back to the periodic gait, however there is no way to enforce the contact force constraint. In our simulation, in addition to model uncertainty, we will introduce a small perturbation:

$$\begin{aligned} q_3(0) &= q_3^* - 2\pi/180, \\ q_4(0) &= q_4^* + 1\pi/180, \\ q_5(0) &= 1.2 q_5^* \\ \dot{q}(0) &= 1.2 \dot{q}^* \end{aligned}$$

where q^* is on the periodic orbit, resulting in an initial CLF $V_0 = \eta_0^T P \eta_0 = 5.8$ (see Fig.(8.2)). We will compare three different controllers,

$$\begin{cases} \text{I: CLF-QP} \\ \text{II: CLF-QP with Constraints (Contact Force Constraints)} \\ \text{III: Robust CLF-QP with Robust Constraints} \end{cases} \tag{8.25}$$

We also enforce input saturation constraints for all three controller in (8.25). However, since this constraint doesn't depend on the system model, a robust version for the constraint is not necessary.

Three simulation cases with different loads carried on the torso of the robot are conducted:

$$\begin{cases} \text{Case 1: } m_{load} = 0 \text{ [kg]} \\ \text{Case 2: } m_{load} = 5 \text{ [kg] (16\%)} \\ \text{Case 3: } m_{load} = 15 \text{ [kg] (47\%)} \end{cases} \tag{8.26}$$

We consider contact force constraints as follows. Let F_{st}^h and F_{st}^v be the horizontal and vertical contact force between the stance foot and the ground, in order to avoid slipping during walking, we will have to

guarantee:

$$F_{st}^{v}(x) \geq \delta_N > 0 \tag{8.27}$$

$$\left| \frac{F_{st}^{h}(x)}{F_{st}^{v}(x)} \right| \leq k_f \tag{8.28}$$

where δ_N is a positive threshold for the vertical contact force, and k_f is the friction coefficient. In our simulation, we picked $\delta_N = 0.1 m_{robot}$ and $k_f = 0.8$, with $m_{robot} = 32[kg]$ being the weight of the robot.

As we can see from Fig.8.2, although we just generate a small initial perturbation, the controller I (CLF-QP) without considering contact force constraints still violated the friction constraint with $|F/N|_{max} \simeq 2.4$, while the controller II (CLF-QP with constraints) can handle this case well. However, with a small model uncertainty (adding $m_{load} = 5[kg]$ to the torso), the controller B fails with the friction coefficient $|F/N|_{max} \simeq 1.1$. Interestingly, in this case the robust CLF-QP with robust contact force constraints controller not only guarantees a very good friction constraints with $|F/N|_{max} \simeq 0.3$, but also has better tracking performance. With $m_{load} = 15[kg]$, while the two controllers I (CLF-QP) and II (CLF-QP with constraints) become unstable and fail in the first walking step, the controller III (Robust CLF-QP with robust constraints) still works well with $|k|_{max} \simeq 0.4$. Especially, we can notice from the figures of $\|u\|$ that the proposed robust CLF-QP with robust constraints has a much better performance in both cases, its range of control inputs is nearly the same with those of the rest two controllers. In summary, we can conclude that the proposed robust QP offers a novel method that can significantly increase the robustness of both stability and constraints while using the same range of control inputs with prior controllers. These properties will be further strengthened in the next interesting application for bipedal robotic walking with safety-critical constraints.

8.3.4 Dynamic Walking of Bipedal Robot while Carrying Unknown Load, subject to Contact Force Constraints and Foot-Step Location Constraints

For validating our proposed controller, we will also test with the problem of footstep location constraints when the robot carries an unknown load on the torso. The control methodology for this problem with perfect model can be found in [58]. We will run 100 simulations. For each simulation, the unknown load was chosen randomly between 5-15 kg, the desired footstep locations for 10 steps were chosen randomly between 0.35-0.55 m (the nominal walking gait has a step length of 0.45 m). Because the CLF-QP cannot handle footstep location constraints, the four following controllers will be compared:

$$\begin{cases} \text{I :CBF-CLF-QP (Foot-Step Placement)} \\ \text{II: CBF-CLF-QP with Constraints (Friction Constraints)} \\ \text{III: Robust CBF-CLF-QP} \\ \text{IV: Robust CBF-CLF-QP with Robust Constraints} \end{cases} \tag{8.29}$$

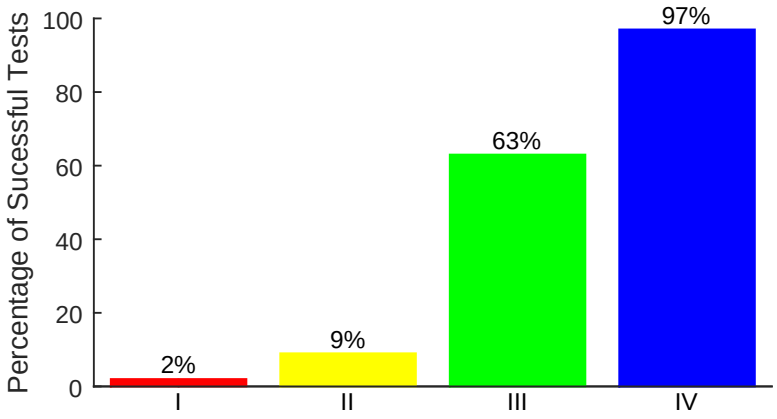


Figure 8.3: Dynamic Walking of Bipedal Robot while Carrying Unknown Load, subject to Contact Force Constraints and Foot-Step Location Constraints. 100 random simulations were tested. For each simulation, the unknown load was choose randomly between 5-15 kg, the desired footstep locations for 10 steps were choose randomly between 0.35-0.55 m. The same set of random parameters was tested on the four controllers, where the four controller was specified in (8.29).

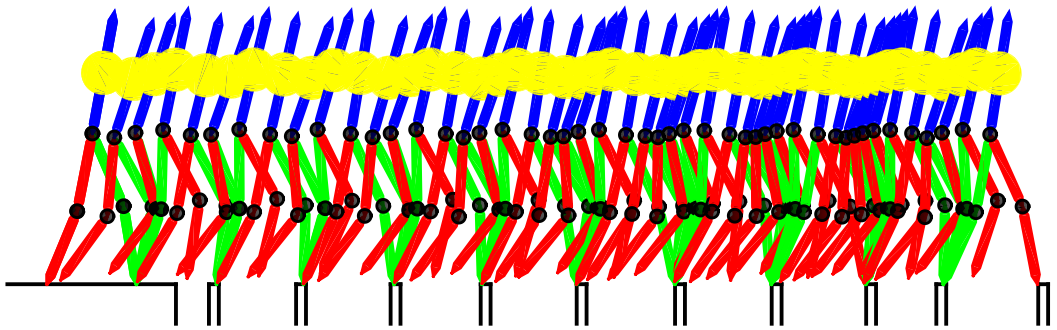


Figure 8.4: Dynamic bipedal walking while carrying unknown load, subject to torque saturation constraints (input constraints), contact force constraints (state constraints), and foot-step location constraints (safety constraints). Simulation of the Robust CBF-CLF-QP with Robust Constraints controller for walking over 10 discrete foot holds is shown, subject to model uncertainty of 15 Kg (47 %). An animation of the simulation is available at `http://youtu.be/tT0xE1XlyDI`.

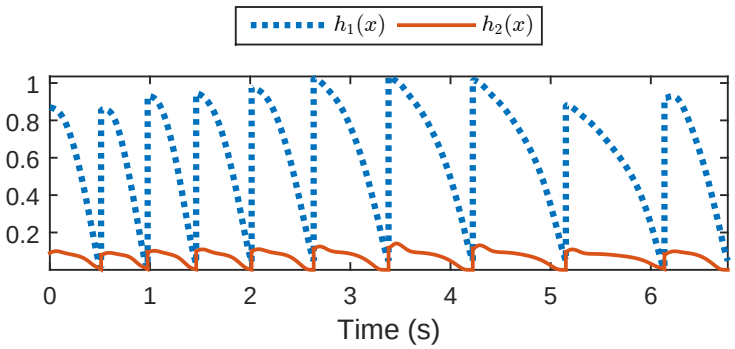
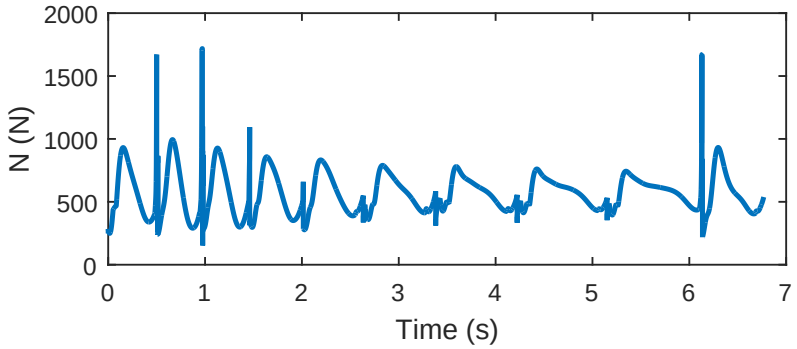
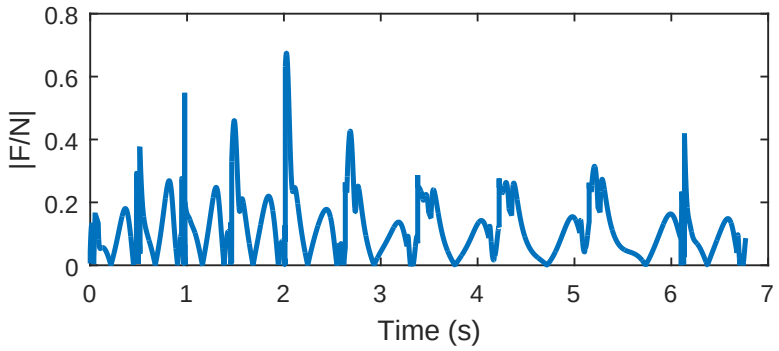


Figure 8.5: Dynamic walking of bipedal robot while carrying unknown load of 15 Kg (47 %). The CBF constraints, $h_1(x) \geq 0$ and $h_2(x) \geq 0$ defined in [58], guarantee precise foot-step locations. The figure depicts data for 10 steps of walking. As can be clearly seen, the constraints are strictly enforced despite the large model uncertainty.



(a) Vertical Contact Force: $N(x) > \delta_N, (\delta_N = 0.1mg)$.



(b) Friction Constraint: $|F/N| \leq k_f, (k_f = 0.8)$

Figure 8.6: Dynamic walking of bipedal robot while carrying unknown load of 15 Kg (47 %). (a) Vertical contact force constraint and (b) friction constraint are shown for 10 steps walking. As is evident, both constraints are strictly enforced despite the large model uncertainty.

As you can see from Fig.8.3, the performance of our proposed robust CBF-CLF-QP dominates that of the CBF-CLF-QP (97% of success in comparision with 2%). This result not only strengthens the effectiveness of the proposed controller, but it also emphasizes the importance of considering robust control for safety constraints because a small model uncertainty can cause violations of such constraints and thereby no longer guaranteeing safety.

Figures 8.8, 8.5, 8.6, illustrate one of the runs where the maximum load of 15 Kg (47% of robot mass) was considered. Stick figure plot, CBF constraints, vertical contact force, and friction constraint plots are shown. Note that, the simulations were artificially limited to 10 steps, to enable fast execution of 100 runs for each controller.

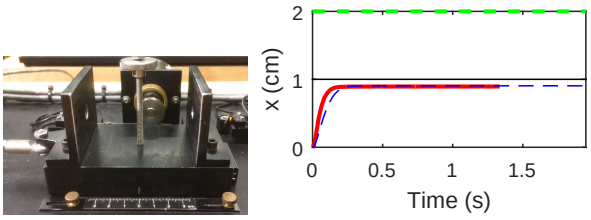
8.4 Experimental Validation of Robust CBF on Spring-Cart System

Experimental validation for the method was implemented on a rectilinear spring-cart system shown in Fig. 8.7. Our control problem is to track desired set-point ($x \rightarrow x_d = 2\ [cm]$) by using CLF, and guarantee state-dependent constraint ($x \leq x_{max} = 1\ [cm]$) by using CBF, where x is the position of cart 1.

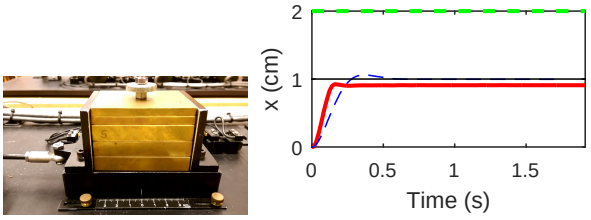
Fig.8.7 compares the performance of two controllers CBF-CLF-QP (dotted blue line) and Robust CBF-CLF-QP (red line) under six different cases. Experimental setup for each case and corresponding result are shown in Fig.8.7. Note that the two controllers were designed based on the nominal model indicated in Case 1 (a single cart) and we generated model uncertainty from Case 2-6 by adding masses, shaking table, adding spring and another cart, .etc.

From Fig.8.7, we can observe clearly that while in Case 1 (without model uncertainty), the two controllers have almost the same performance, from Cases 2-6, our proposed robust CBF-CLF-QP outperformed the nominal CBF-CLF-QP. To be more specific, the robust CBF-CLF-QP controller maintains the constraint ($x \leq 1(cm)$) very well, the nominal CBF-CLF-QP fails in all last 5 cases.

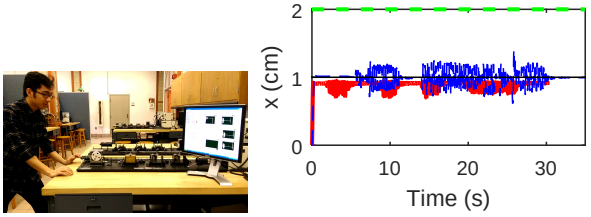
The proposed controller has a few shortcomings. Since we are solving for the control input under the worst-case (bounded) model uncertainty assumption, the control could be aggressive. This is a typical drawback of robust controllers. Moreover, as mentioned in Remark 4.3, we only have local feasibility of the QP. In particular, if we increase the bounds of the uncertainty significantly, i.e., large values of $\Delta_1^{max}, \Delta_2^{max}$, the optimization solves for the control input for the worst case, and could potentially lead to infeasibility of the QP. Thus, there is a trade-off between robustness and feasibility of the controller. If we choose a small bound on the model uncertainty, it could lead to poor tracking stability and potential violation of the safety constraint under mild model uncertainty that exceeds the bounds. In contrast, if we assume the bound of model uncertainty being too large the QP could become infeasible.



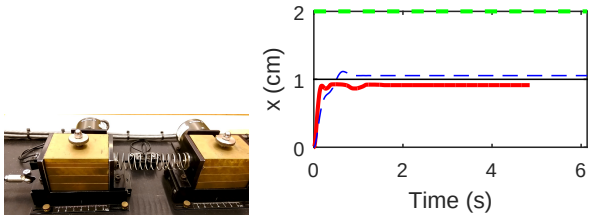
Case 1: No uncertainty



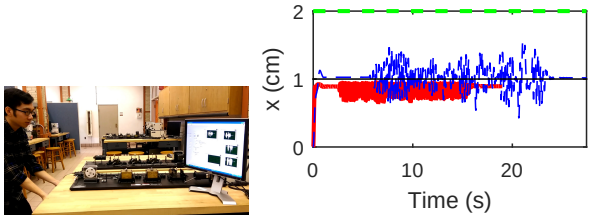
Case 2: Full-loaded cart 1



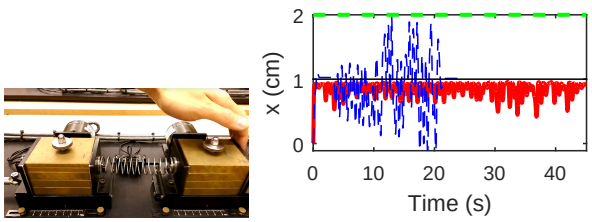
Case 3: Case 2 and shaking the table



Case 4: Full-loaded cart 1 + spring + full-loaded cart 2



Case 5: Case 4 and shaking the table



Case 6: Case 4 and perturbing the cart 2.

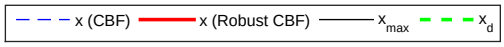


Figure 8.7: Experimental results on the spring-mass system. The goal is to drive the cart to the target location of 2 cm, while enforcing the safety constraint that the cart does not cross 1 cm. The controller just uses the nominal model as illustrated in Case 1 for all the 6 different cases. Model uncertainty is introduced for Cases 2 in the form of an added unknown mass. Additionally for Case 3, a perturbation is introduced in the form of shaking the table. For Case 4, in addition to the unknown mass, an unkown dynamics is introduced in the form of another cart that is connected through a spring. Additionally for Cases 5 and 6, perturbations are introduced in the form of manually shaking the table and shaking the second cart respectively. In all cases, the proposed robust controller still enforces the strict safety-critical constraint and maintains the cart position under 1 cm. A video of the experiment is available at `https://youtu.be/g1UewP4R8L4`.

8.4.1 Numerical Validation of Robust Dynamic Bipedal Walking over Time-Varying Stepping Stones

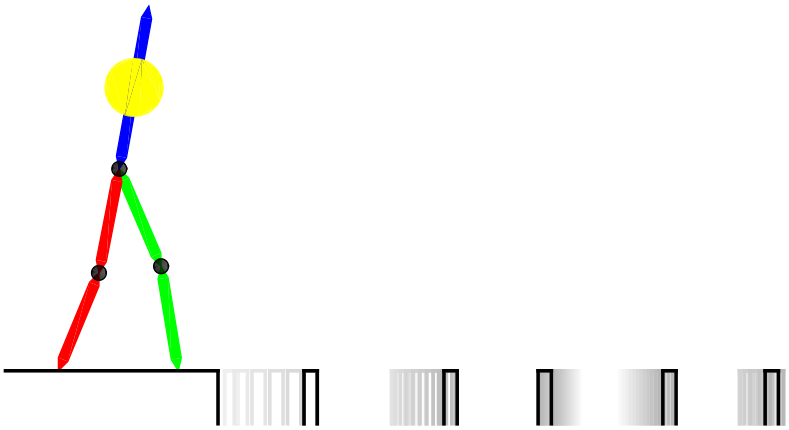


Figure 8.8: Dynamic bipedal walking while carrying unknown load, subject to torque saturation constraints (input constraints), contact force constraints (state constraints), and moving stepping stones (time-varying safety-critical constraints). Dynamical walking over 10 steps is carried out, subject to model uncertainty of carrying an unknown load of 15 Kg (47 %). Simulation video: `https://youtu.be/nBShDTn7IlI`.

To demonstrate the effectiveness of the proposed controller, we will conduct numerical simulations on the model of RABBIT, [15], a planar five-link bipedal robot with a torso and two legs with revolute knees that terminate in point feet. RABBIT weighs 32 kg, has four brushless DC actuators with harmonic drives to control the hip and knee angles, and is connected to a rotating boom which constrains the robot to walk in a circle, approximating planar motion in the sagittal plane. The dynamical model of RABBIT is nonlinear and hybrid, comprising of a continuous-time underactuated stance phase and a discrete-time impact map.

We validate the performance of our proposed approach through dynamic bipedal walking on RABBIT, subject to model uncertainty while simultaneously enforcing foot placement on time-varying stepping stones, ground contact force constraints, and input saturation. Model uncertainty appears in the form of an unknown heavy load added to the torso of RABBIT.

In bipedal robotic walking, contact force constraints are very important. Any violation of these constraints will result in the leg slipping and the robot potentially falling. Although a nominal walking gait is usually designed to respect these constraints, however, we cannot guarantee these constraints under transients or under model uncertainty. Our proposed controller addresses this through the robust constraints formulation in (8.24). In particular, we consider, $F(x,u)$ and $N(x,u)$, the friction force and vertical contact

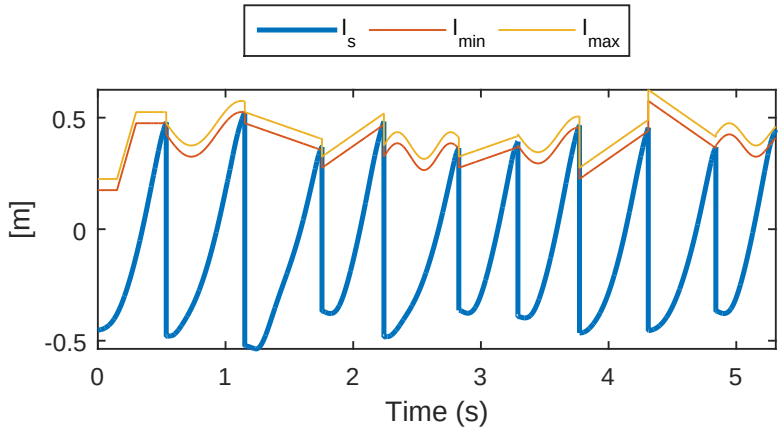


Figure 8.9: Dynamic walking of bipedal robot while carrying unknown load of 15 Kg (47 %). Ten steps of the horizontal position of the swing leg (thick blue) with respect to the stance leg, and desired stone locations varying either linearly or sinusoidally over time (thin orange / red) are shown. The proposed controller guarantees strict enforcement of the foot placement constraints on the moving stepping stones, even with large model uncertainty.

force between the stance foot and the ground. Then, the constraints to avoid slipping during walking are,

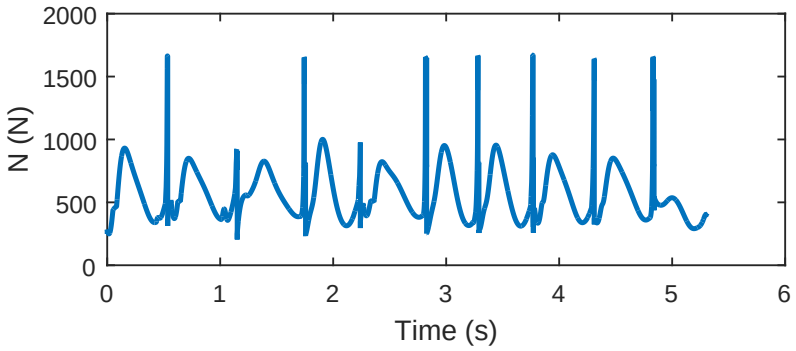
$$N(x,u) \geq \delta_N > 0, \tag{8.30}$$

$$\left|\frac{F(x,u)}{N(x,u)}\right| \leq k_f, \tag{8.31}$$

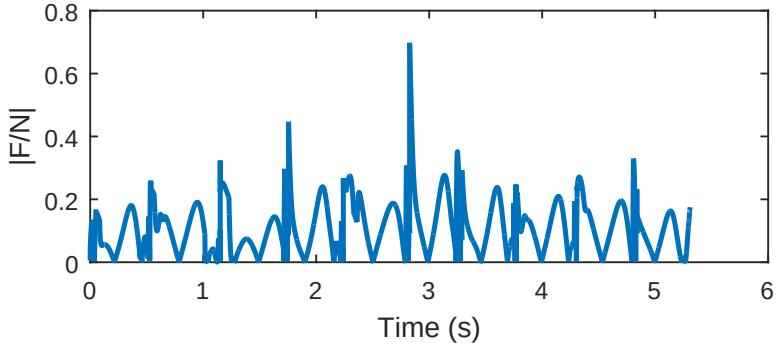
where δ_N is a positive threshold for the vertical contact force, and k_f is the friction coefficient. We enforce the above ground contact constraints with $\delta_N = 0.1mg$, $k_f = 0.8$.

Furthermore, we consider the problem of dynamic walking with precise foot placements on moving stepping stones, which translate to time-varying safety-critical constraints as detailed in Section 6.1. We then add torque constraints of 150 Nm as per the motor specifications. Achieving dynamic walking with the above constraints is fairly challenging. To make the problem even harder, we consider large model uncertainty corresponding to the addition of an unknown load of 15 Kg (47% of robot mass) on the torso of the robot. It must be noted that even a small model uncertainty can cause violation of these constraints.

Our proposed robust controller in (8.24) strictly enforces the state constraints corresponding to the ground contact and friction constraints, as seen in Figure 8.9, and the time-varying safety-critical constraints corresponding to stepping onto the moving stepping stones, as seen in Figure 8.10. These figures illustrate only 10 steps of walking since the time-varying profile of only ten stepping stones were selected as below,



(a) Vertical Contact Force: $N(x) > \delta_N, (\delta_N = 0.1mg)$.



(b) Friction Constraint: $|F/N| \leq k_f, (k_f = 0.8)$

Figure 8.10: Dynamic walking of bipedal robot while carrying unknown load of 15 Kg (47 %). (a) Vertical contact force constraint and (b) friction constraint are shown for 10 steps of walking. As is evident, both constraints are strictly enforced despite the large model uncertainty.

(1) $l_1^d(\tau) = \begin{cases} 0.2 & \tau < 0.15, \\ 0.2 + 2(\tau - 0.15) & 0.15 \leq \tau \leq 0.3, \\ 0.5 & \tau > 0.3, \end{cases}$

(2) $l_2^d(\tau) = 0.45 - 0.1\sin(8\tau)$,

(3) $l_3^d(\tau) = 0.5 - 0.2\tau$,

(4) $l_4^d(\tau) = 0.3 + 0.4\tau$,

(5) $l_5^d(\tau) = 0.35 + 0.06\sin(15\tau)$,

(6) $l_6^d(\tau) = 0.3 + 0.2\tau$,

(7) $l_7^d(\tau) = 0.4 - 0.08\sin(10\tau)$,

(8) $l_8^d(\tau) = 0.25 + 0.4\tau$,

(9) $l_9^d(\tau) = 0.6 - 0.4\tau$,

(10) $l_{10}^d(\tau) = 0.4 + 0.05\sin(15\tau)$,

where τ is the time reinitalized to zero at the beginning of each walking step. Then, for the i^{th} step, $l_{min}^i(\tau), l_{max}^i(\tau)$ were picked with an offset of $\pm 2.5\ cm$ about the desired footstep location l_i^d, i.e.,

$$l_{min}^i(\tau) = l_i^d(\tau) - 0.025, \quad l_{max}^i(\tau) = l_i^d(\tau) + 0.025, \tag{8.32}$$

where $i = 1, \cdots, 10$.

The controller can be easily shown to demonstrate longer durations of walking by specifying motion profiles for additional stepping stones.

It must be noted that the controller can fail in the following ways. Firstly, if the desired stone locations are too short or too long at the end of the step, although the controller still enforces the foot placement constraints, however on the subsequent step the robot state is too far away from the periodic orbit, causing a failure on the following step. Since our method relies on making changes to only one nominal walking gait, the variations in step lengths are limited. One potential solution would be to switch from one walking gait to another when the step lengths start increasing / decreasing. A second failure mode is, if the desired stepping stone location are varied too aggressively, beyond the mechanical bandwidth of the system, then the required torques to enforce the safety-constraint are beyond the torque saturation constraint, resulting in solution of the QP being infeasible.

Note that, as presented in [61], the non-robust version of the proposed controller can achieve precise footstep placements for 3D dynamic walking on a 28 degree-of-freedom humanoid, indicating that the proposed controller is scalable to higher dimension systems.

8.5 Summary

We have presented a novel Optimal Robust Control technique that uses control Lyapunov and barrier functions to successfully handle significantly high model uncertainty for both stability, input-based constraints, state-dependent constraints, and safety-critical constraints. We validated our proposed controller on different problems both numerically and experimentally, which show the same property: under model uncertainty, our Robust Control based QP, has much better tracking performance and guarantees desired constraints while other types of QP controllers using Lyapunov and barrier functions not only have large tracking errors but also violate the constraints with even a small model uncertainty. We show numerical validations on a suspended pendulum subject to a position constraint in the presence of model uncertainty, dynamically walking while subject to torque saturation and contact force constraints in the presence of model uncertainty, and dynamically walking with precise foot placements over a terrain of stepping stones while subject to model uncertainty. We also experimentally validate our controller on a spring-cart system subject to significant model uncertainty and perturbations.

Chapter 9

Adaptive Control Barrier Functions

This Section will present a novel approach called adaptive control Barrier functions (CBF) to guarantee safety-critical constraints of nonlinear systems under model uncertainty. Inspired by our work on using L_1 adaptive control with CLF-QP based closed-loop reference model presented in Section 4.2 that can be used to guarantee stability under model uncertainty, we now develop the method here to be able to address both stability and constraints under model uncertainty. Our approach works for both actuated and under-actuated systems. Using the reference model being a closed-loop control system controlled by control Barrier function and control Lyapunov function based quadratic program (CBF-CLF-QP), our adaptive control framework drives the performance of the real system including safety and stability close to those of the reference model.

9.1 Adaptive Control Barrier Functions for Fully Actuated Systems

In this Section, we will proposed a new method to enforce both stability and safety-critical constraints under model uncertainty. Our method offers the flexibility to design a closed-loop reference model based on our desired control goal. We therefore can design the reference model so that the desired system state will respect the additional properties (safety-critical constraints in this case) rather than just only stability or convergence to the desired trajectory. Note that for the conventional L_1 adaptive control which uses nominal linear reference model [9], it is not trivial to incorporate additional control goals to the system.

The control diagram of the proposed method on L_1 adaptive control using $CBF - CLF - QP$ based closed-loop reference model is illustrated in Fig. 9.1. As mentioned in Section. 4.2.2, the adaptation law will compare the difference between the real system output η and the reference system output $\hat{\eta}$ in order to estimate the model uncertainty $\hat{\theta}$. The adaptation law will play the role as a feedback controller that

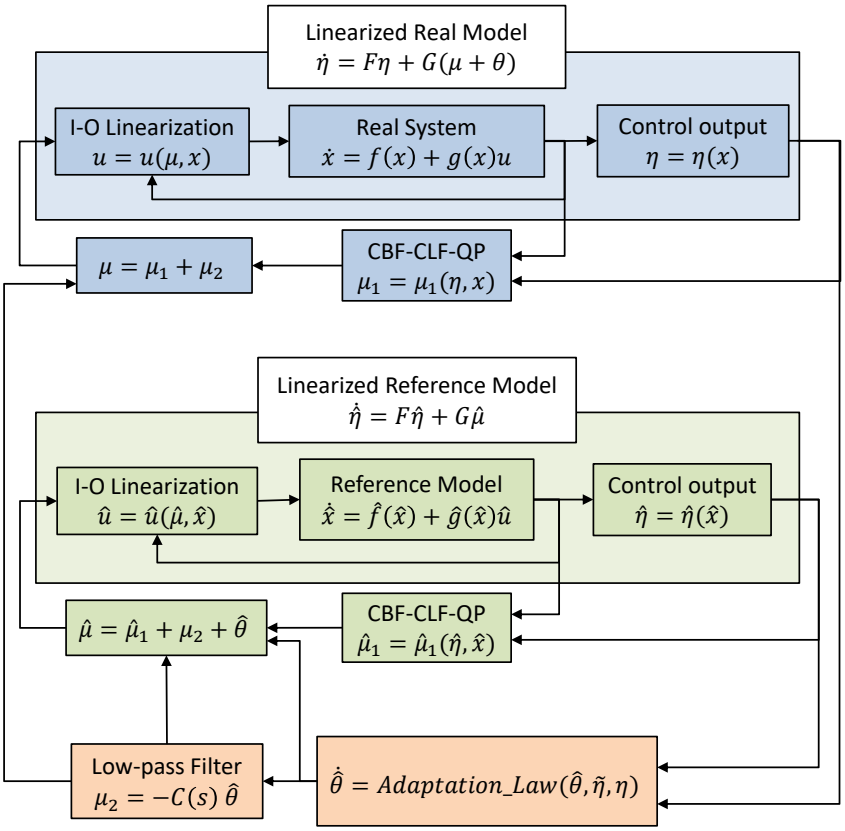


Figure 9.1: Control diagram illustrating L_1 adaptive control with a CBF-CLF-QP based closed-loop reference model for fully-actuated systems.

applies $\hat{\theta}$ to drive $\tilde{\eta} = \hat{\eta} - \eta \to 0$. With fully-actuated system, driving control output $eta \to \hat{\eta}$ is equivalent to driving system state $\hat{x} \to x_hat$. Furthermore, in our proposed method, we also control the reference model to enforce desired constraints or guarantee the reference system state to stay within the safety set. Therefore, as the adaptation law helps to drive the difference between the real state and the reference state $\tilde{x} = \hat{x} - x \to 0$, the real system state x will follow the same behavior of the reference system sate $\hat{x}$. As a result, the safety-critical constraints of the real system will be guaranteed under model uncertainty.

9.2 Adaptive Control Barrier Functions for Under-Actuated Systems

Since we are interested in applying our controller to a biped robot, an under-actuated system, we also want to extend the adaptive CBF framework for under-actuated systems. In this Section, we will introduce a modification of the approach presented in Section 9.1. For under-actuated system, the dimension of the control output η is less than that of the system state x. Therefore, while the adaptive control can drive η close to the reference state $\hat{\eta}$, the under-actuated coordinate of the real system may differ from that of the reference model. In other words, the major challenge of this problem is the presence of internal state ξ that are not a part of the control output η. The adaptive control can help to drive the real control output η to the reference control ouput $\hat{\eta}$. However, because the internal state ξ and $\hat{\xi}$ are not controlled or in some cases uncontrollable, the dynamics of ξ and $\hat{\xi}$ will differ significantly under high model uncertainty, resulting in unexpected differences between the performance of the real system and the reference model. The adaptation law therefore will become aggressive and the system will possibly be unstable. In order to overcome this issue, in Fig. 9.2, we propose a control diagram for a new adaptive CBF approach for under-actuated systems, where the internal state ξ from the real system will be combined with the reference output $\hat{\eta}$ to extract a new modified reference state $\hat{x}_m$. This modified reference state $\hat{x}_m$ will be used to design the reference controller as well as the dynamics of the reference model instead of using $\hat{x}$ as we did for the fully actuated adaptive CBF.

9.3 Simulation Result of Adaptive Control Barrier Functions

In this Section, we will present various numerical validation of the proposed method on adaptive control Barrier functions. Following are the systems that we evaluate our proposed controller on.

- Single cart system, a simple linear system
- Two-link pendulum, a non-linear and fully-actuated system
- Acrobot model, a non-linear and under-actuated system

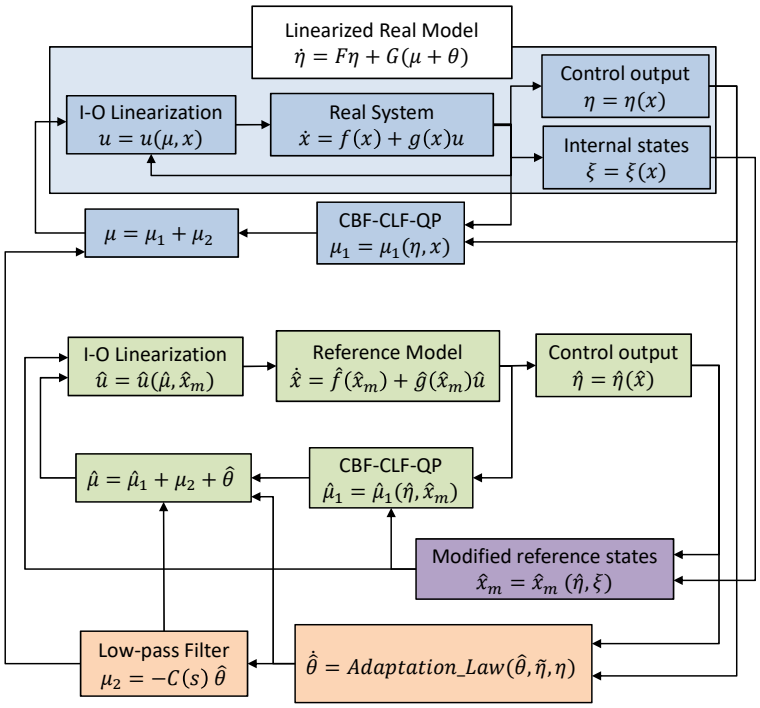
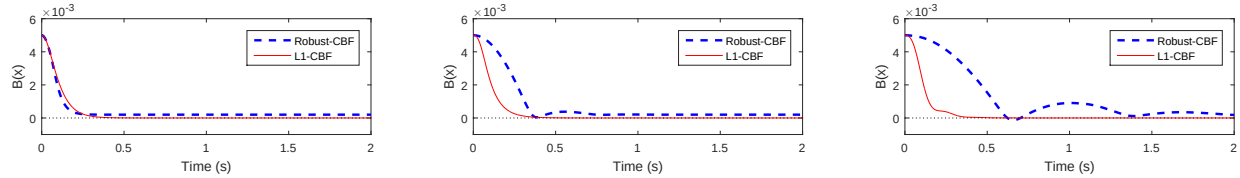
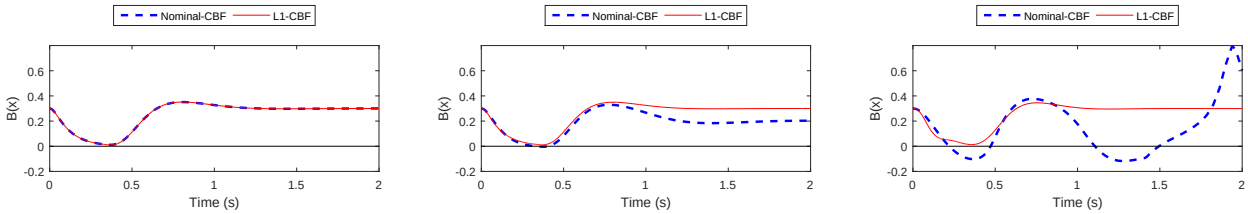


Figure 9.2: Control diagram illustrating L_1 adaptive control with a CBF-QP based closed-loop reference model for under-actuated systems.



(a) No Uncertainty(m=1 kg) (b) m=10 (kg) (c) m=30 (kg)

Figure 9.3: Simulation on a single cart system: Comparison of robust CBF and L1 adaptive CBF with different levels of model uncertainty. Our control goal is to enforce the state-dependent constraint $B(x) \geq 0$.



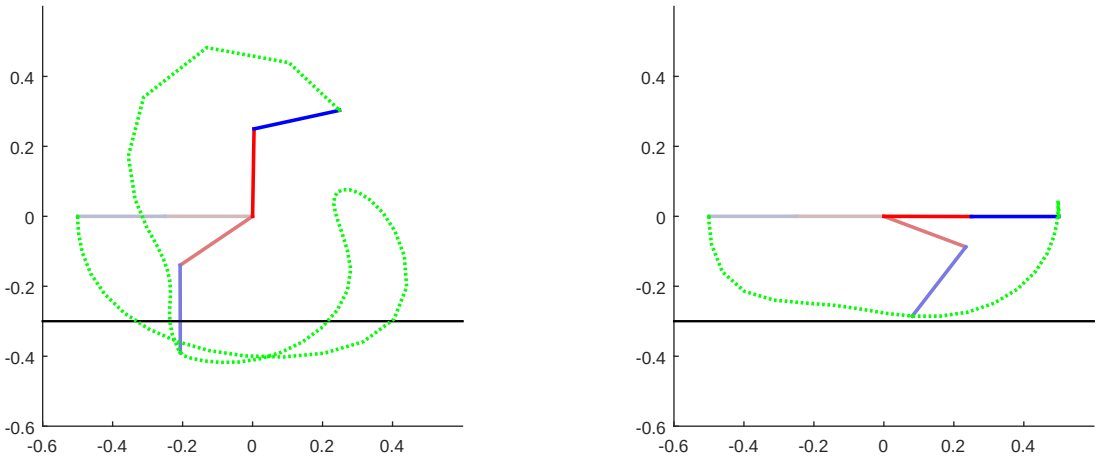
(a) No Uncertainty (b) mass scale = 1.5 (c) mass scale = 5

Figure 9.4: Simulation of a two-link pendulum: Comparison of nominal CBF and L1 adaptive CBF with different levels of model uncertainty. The model uncertainty is generated by scaling both the mass of two links of the pendulum by a certain value. Our control goal is to enforce the state-dependent constraint $B(x) \geq 0$.

- Bipedal robot walking on stepping stones while carrying an unknown load, a non-linear under-actuated and hybrid system
- Two-link pendulum with L1 adaptive CBF using linearized model

9.3.1 Single Cart System (a simple linear system)

In this problem, the control goal is to drive the cart position x_1 as close as possible to the set point of $x_1^d = 1(cm)$, but also guarantees that $x_1 \leq x_1^{max} = 0.5(cm)$. We will compare the performance of robust CBF and L1 adaptive CBF. The model uncertainty is introduced by scaling the mass of the cart. As seen in Fig. 9.3, the result shows that with the mass scale of 30, the L1 adaptive CBF still guarantees the safety constraint while the robust CBF controller fails to keep the system state in the safety set even with the mass scale of 10.



(a) Nominal CBF with mass scale of 5

(b) L1 CBF with mass scale of 5

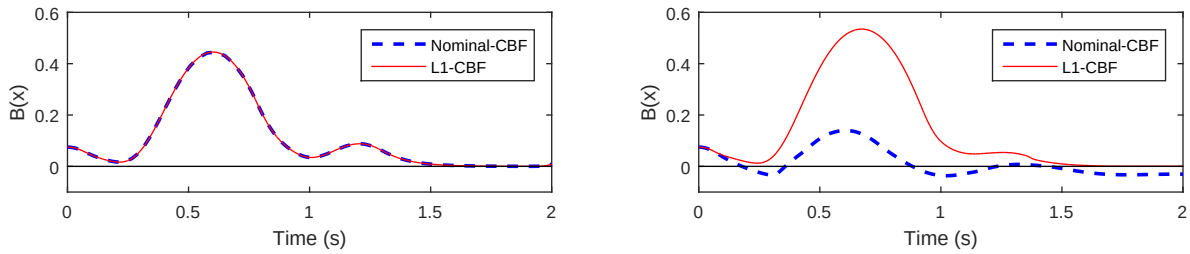
Figure 9.5: Simulation of a two-link pendulum: Comparison of nominal CBF and L1 adaptive CBF with model uncertainty scale of 5. The model uncertainty is generated by scaling both the mass of two links of the pendulum by a certain value.

9.3.2 Two-link Pendulum (a nonlinear and fully-actuated system)

For this problem, we use the model of two-link pendulum with the length and mass of each link are $0.25\ m$ and $0.5\ kg$ respectively. The control goal is to drive the absolute angles of the two links from $[-\pi/2; -\pi/2]$ to $[\pi/2; \pi/2]$ while maintaining the constraint of the vertical end-effector position (y_E) to be lower-bounded or $y_E \geq y_E^{min}$ (in this simulation we set $y_E^{min} = -0.3\ m$). We then generate model uncertainty by scaling both masses of the two links by a certain factor (the controller is still designed based on the assumption that the mass of each link has the nominal value of $0.5\ kg$). Fig. 9.4 compares the performance of nominal CBF and L1-CBF for three different cases: (a) no model uncertainty, (b) masses of two links are scaled by 1.5 and (c) masses of two links are scaled by 5. With no model uncertainty, the performances of nominal CBF and L1 adaptive CBF are identical. However, when we apply model uncertainty by scaling the masses of two links by a factor of 5, the nominal CBF violates the constraints as well as becomes unstable. The L1 adaptive CBF controller, in contrast, guarantees both stability and constraints well. The simulation snapshots of these two controllers in this case are shown in Fig. 9.5.

9.3.3 Acrobot model, a non-linear and under-actuated system

Acrobot is a two-link pendulum, but with an under-actuated joint at the base pivot. The Acrobot model used for this simulation has the mass and length of each link of $0.5\ kg$ and $0.25\ m$ respectively. Since this



(a) No Uncertainty

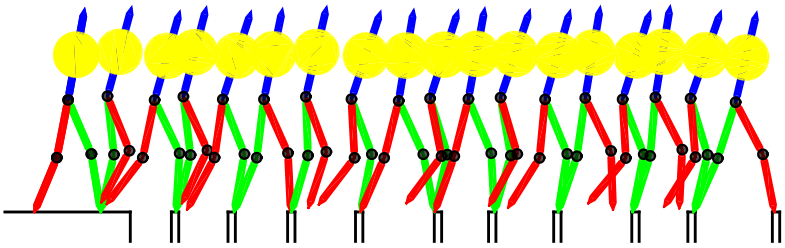
(b) With Model Uncertainty

Figure 9.6: Simulation of a Acrobot model: Comparison of nominal CBF and L1 adaptive CBF with and without model uncertainty. The model uncertainty is generated by scaling the mass of link 1 and link 2 by factors of 0.5 and 5 respectively. Our control goal is to enforce the state-dependent constraint $B(x) \geq 0$.

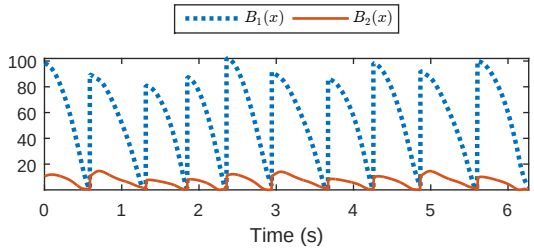
is an under-actuated system, we simplify the control goal by defining an one-dimensional control output $y(q) = q_1 - q_2 - q_{12}^d$, where q_1, q_2 are absolute angles of the two links, q_{12}^d is the desired relative angle between the two links (we set $q_{12}^d = 0\ rad$ for this simulation). In addition to that, we also enforce state-dependent constraint of the vertical end-effector position (y_E) to be lower-bounded or $y_E \geq y_E^{min}$ (in this simulation we set $y_E^{min} = -0.45\ m$). Fig. 9.6 shows the comparison between the nominal CBF and the L_1 adaptive CBF. Without model uncertainty, the two controllers have the same performance.However, when we apply model uncertainty by scaling the masses of link 1 and link 2 by factors of 0.5 and 5 respectively, the L_1 adaptive CBF is still able to strictly enforce the constraint while the nominal CBF violates it significantly. Note that we apply two different scaling factor to the masses of the two links in order to better perturb the under-actuated dynamic, that helps to illustrate more clearly the effectiveness of our proposed controller for under-actuated systems.

9.3.4 Bipedal robot walking on stepping stones while carrying an unknown load, a non-linear under-actuated and hybrid system

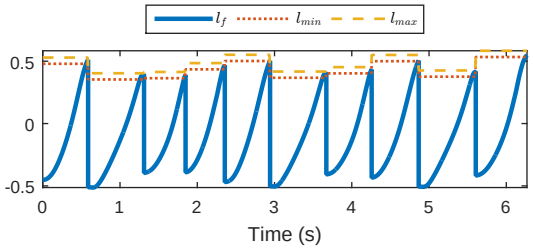
In this part, we will apply the proposed method on L_1 adaptive CBF for the problem on bipedal robotic walking on stepping stones while carrying an unknown load. For this problem, the robot needs to address model uncertainty cased by an unknown load carried on the torso, while enforcing safety-critical constraints to enforce precise footstep placement. A similar numerical study for this application was also conducted earlier in Section. 8.3.4 using our proposed method of Robust CBF, which was successfully validated with the unknown load of $15\ kg$ ($47\ \%$ of the robot weight). With this development of L_1 adaptive control, the robot can even carry $30\ kg$ of unknown load, which is $94\ \%$ of the robot weight while adjusting step length for every walking step within the range of $[35 : 55]\ cm$ (see Fig. 9.7). Note that the Robust CBF controller fails right at the first walking step with this level of model uncertainty.



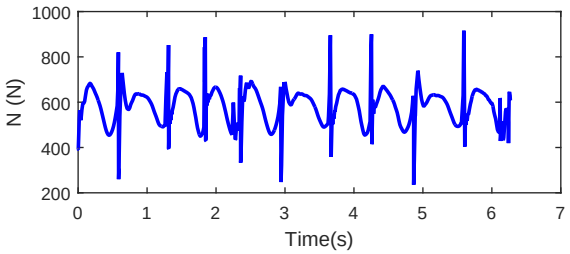
(a) Simulation Snapshots



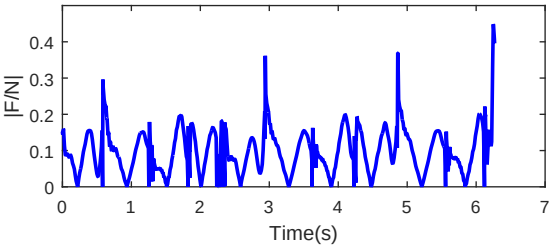
(b) CBF constraints



(c) Swing Foot Trajectory

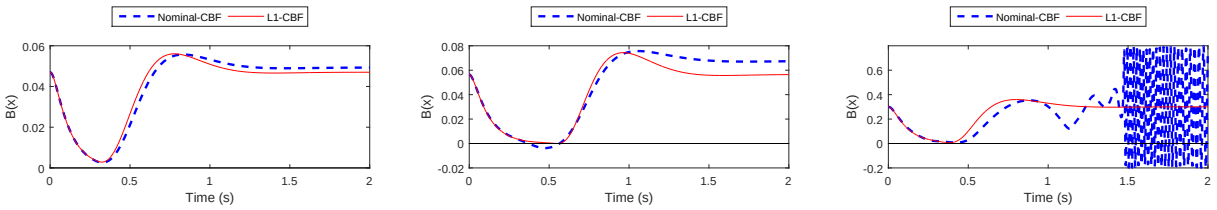


(d) GRF



(e) Friction Constraint

Figure 9.7: Simulation of L_1 adaptive CBF controller on a bipedal robot walking on stepping stones while carrying an unknown load of 30 kg (94 $\%$ of the robot weight).



(a) (b) (c)

Figure 9.8: Simulation of a two-link pendulum model: Comparison of nominal CBF and L1 adaptive CBF using linearized model with different working ranges: (a) $[q_1^0, q_2^0] = [-\pi/6, -\pi/6], [q_1^d, q_2^d] = [\pi/6, \pi/6], y_E^{min} = -0.48\ m$; (b) $[q_1^0, q_2^0] = [-\pi/4, -\pi/4], [q_1^d, q_2^d] = [\pi/4, \pi/4], , y_E^{min} = -0.41\ m$; (c) $[q_1^0, q_2^0] = [-\pi/2, -\pi/2], [q_1^d, q_2^d] = [\pi/2, \pi/2], , y_E^{min} = -0.3\ m$, where $[q_1^0, q_2^0]$ and $[q_1^d, q_2^d]$ are the initial condition and the desired set points of the two link angles, y_E^{min} is the minimum vertical position of the end-effector that we want to enforce. Our control goal is to enforce the state-dependent constraint $B(x) \geq 0$.

In Fig. 9.7, we include the snapshot of the simulation, CBF constraints, swing foot trajectory, ground reaction force and friction constraint, showing that the control goal and physical constraints are met.

9.3.5 Two-link pendulum with L1 adaptive CBF using linearized model

For this problem, we will use a linearized model of the system to design the controller. To be more specific, in this simulation, we do linearization at the down-right position of the pendulum, or at the absolute angle of the two links $[q_1 = q_2 = 0]$ and zero velocities $\dot{q}_1 = \dot{q}_2 = 0$. The differences between the real non-linear system and the linearized model used to design the controller will affect the performance. Note that, in Section. 9.3.2, we also validate our controller in the two-link pendulum model but we use the full non-linear model to design the controller. Fig. 9.8 shows the comparison of nominal CBF and L1 adaptive CBF using linearized model with different working ranges:

(a) $[q_1^0, q_2^0] = [-\pi/6, -\pi/6], [q_1^d, q_2^d] = [\pi/6, \pi/6], y_E^{min} = -0.48\ m$

(b) $[q_1^0, q_2^0] = [-\pi/4, -\pi/4], [q_1^d, q_2^d] = [\pi/4, \pi/4], , y_E^{min} = -0.41\ m$

(c) $[q_1^0, q_2^0] = [-\pi/2, -\pi/2], [q_1^d, q_2^d] = [\pi/2, \pi/2], , y_E^{min} = -0.3\ m$

where $[q_1^0, q_2^0]$ and $[q_1^d, q_2^d]$ are the initial condition and the desired set points of the two link angles, y_E^{min} is the minimum vertical position of the end-effector that we want to enforce. From Fig. 9.8, it can be seen that in Case (a), when the working range is close enough to the equilibrium point $[q_1, q_2, \dot{q}_1, \dot{q}_2] = [0, 0, 0, 0]$, where we linearized the system, the nominal CBF controller is still able to enforce the constraint $B(x) \geq 0$ (though it has tracking error to the desired set points). However, when we increase the working range, the

tracking error and the constraint violation also increase and then the system becomes unstable in case (c). In contrast, the L_1 adaptive CBF controller guarantee both stability and constraint in all three cases, offering a simple control design based on linearization model but can adapt well with a wide working range of the system states.

9.4 Summary

In this Section, we have presented a novel method of L_1 adaptive safety-critical control that enables the ability to enforce stability and safety-critical constraints under high level of model uncertainty. The proposed adaptive control framework uses the nonlinear closed-loop reference model that is controlled by CBF-CLF-QP. Our method can be applied for both fully-actuated and under-actuated systems. Numerical validation was successfully implemented in various dynamical systems including single cart system, two-link pendulum, Acrobot and bipedal robotic walking. In comparison with robust CBF presented in Section. 8, the L_1 adaptive CBF controller has two advantages. Firstly, since it can estimate and adapt to model uncertainty, it maintains consistent performance under different level of model uncertainty. The robust CBF controller instead tends to be aggressive even without model uncertainty. Secondly, in the same problem setup, the L_1 adaptive CBF controller can address larger model uncertainty. For example, for the problem of bipedal robot walking on stepping stones while carrying an unknown load, the adaptive controller works with a load up to $30\ kg$ but the maximum load that the robust one can handle is $15\ kg$.

Chapter 10

Conclusion and Future Work

10.1 Conclusion

In conclusion, this thesis has presented following main contributions:

- Robust and Adaptive Stability: This work introduced robust and adaptive control Lyapunov function (CLF) based controllers to guarantee stability under model uncertainty. Numerical simulation of the method was validated on the problem of dynamic walking while carrying an unknown load. Experimental validation of L_1 adaptive control on the ATRIAS robot shows that it can achieve very small tracking errors (about $0.01\ deg$ or $0.0017\ rad$ at all joints) while carrying an unknown load of $68\ lb$ ($31\ kg$).

- Safety-Critical Control via Control Barrier Functions: In this work, we presented a novel method of exponential control barrier function (E-CBF) to address safety-critical constraints in dynamical systems. The method offers a systematic design based on pole-placement control to address safety-critical constraints with high-relative degree. We numerically validate the method on various dynamical models including 2-link pendulum with elastic actuator, a relative-degree 4 nonlinear system; and a bipedal robot walking on stepping stones, a non-linear under-actuated hybrid system. Experimental validation of the method is also successfully implemented on the spring-cart system that has relative degree 6.

- Dynamic Walking on Stepping Stones using Control Barrier Functions: We then introduce a new approach to address dynamic walking of bipedal robots on stepping stones using control barrier functions. This method controls the footstep placement in real-time using a feedback controller. The proposed method can also handle time-varying stepping stones. We successfully validated the method

on the model of DURUS robot, a 3D humanoid robot with 23 DoFs, when the robot needs to adjust the step length and step width continuously for each walking step. In order to improve the performance of the method, we also incorporate the method with gait library so that we can increase range of step length variation from $[25 : 60]\ cm$ to $[10 : 100]\ cm$.

- Dynamic Walking on Discrete Terrain with 2-Step Periodic Gait Library: In order to overcome the problem of dynamic walking on discrete terrain with variation on both step length and step height, we introduced a novel method of 2-step periodic gait library. The proposed methods are successfully validated experimentally on the ATRIAS robot. In particular, experimental demonstrations illustrate our controller being able to dynamically walk at $0.6\ m/s$ over terrain with step length variation of $[23 : 78]\ cm$, as well as simultaneous variation in step length and step height of $[35 : 60]\ cm$and $[-22 : 22]\ cm$ respectively.

- Robust and Adaptive Safety-Critical Control: In this work, we presented solutions to address safety-critical constrains under high level of model uncertainty. Using the adaptive CBF-CLF controller, the robot can handle dynamic walking on stepping stones while carrying an unknown load up to $30\ kg$ which is $94\ \%$ of the robot weight.

10.2 Future Work

Having presented summary of the main contributions of the thesis, we now briefly discuss a few future research thrusts.

- Vision-based Dynamic Walking on Discrete Terrain: In the experiments presented in Section 7.3, we predefined the stone locations by measuring them at the start of the experiment and presenting this information to the controller one step ahead. For every walking step, the next stone location was then inputted to the controller right after an impact. As part of future work, we plan to use an on-board camera to determine the stone location rather than measuring them. It will help to show more clearly that our controller requires only one step ahead preview and can adapt to the change of stone location in real-time. This is also a step toward bringing the robot outdoor in the future.

  For this work, we attempted to conduct some preliminary experiments using an on-board camera. Fig. 10.1 shows the camera attachment on the robot. The comparison of real camera view and camera scene after filtering is shown in Fig. 10.2. We successfully demonstrated a short experiment with three walking steps. Fig. 10.3 illustrates three snapshots of this experiment, where the desired step length is accurately determined using an on-board stereo camera. Due to delay in camera processing, there are still lots of cases when the camera processing give a wrong estimation of the distance between the

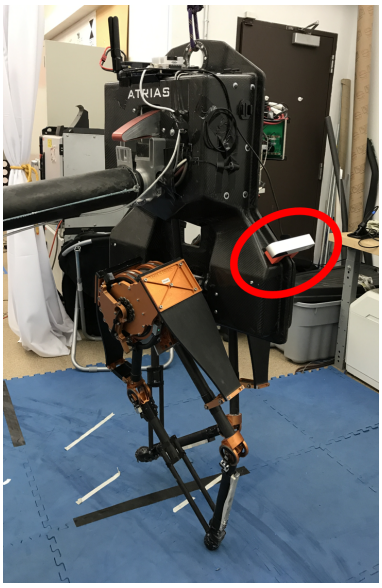
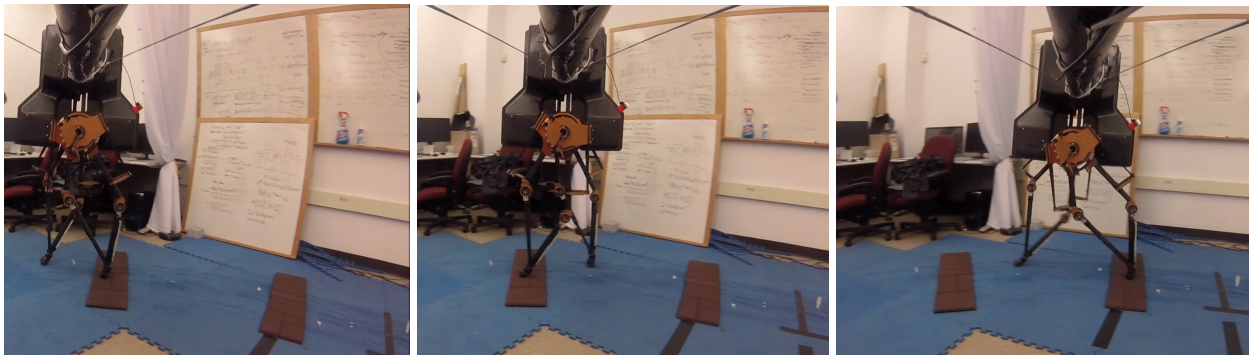


Figure 10.1: On-board camera attachment on the ATRIAS robot.

(a) (b)

Figure 10.2: Comparison of (a) real camera view and (b) camera scene after filtering.



(a) (b) (c)

Figure 10.3: Preliminary experiment on stepping stones using on-board camera. These are the snapshots of three consecutive walking steps.

camera to the stepping stones. A better synchronization solution between the robot and the camera could potentially compensate for the delay. Further results with a systematic development of the stereo algorithm will be tested and presented as part of future work. In an attempt to bring better perception for walking robots, we presented related result on synthetic vision for deep visual perception for dynamic walking on stepping stones in [67].

- 3D Dynamic Walking on Discrete Terrain: The work of dynamic walking on discrete terrain still needs to be extended to address 3D walking with simultaneous variation on step length, height and width, or even slope of the terrain. The dimension of the problem and the number of gaits therefore may need to be increased. However, with a fully-actuated 3D biped, we can possibly incorporate gait library with some other control goals like ZMP, speed control, etc., to reduce the dimension of the gait library. Furthermore, with our method of gait library, the impact force to the ground during walking is sensitive with the height change. In other words, a small error in step height input may cause bad impact to the robot. A potential solution could be optimizing a robust gait library, where the optimization process will also take into account the step height uncertainty so that the gait library can be more robust to the step height estimation.

- Systematic Design Procedures for Robust Safety-Critical Control: Our proposed method of robust CBF-CLF-QP controller can maintain stability and safety under a high level of model uncertainty. The optimal robust controller solves the minimal control efforts to guarantee robust CBF and CLF conditions under the assumption of a bounded model uncertainty. The controller therefore has several design parameters representing bounds on the uncertainty. However, in our current approach, these control parameters are hand-tuned. A better method should offer a systematic design procedure for these parameters. Because the effect of model uncertainty in the IO linearized system is nonlinear and time-varying, it is challenging to analytically quantify the bounds of model uncertainty for the robust controller design. However, one possible solution could be using nonlinear optimization to solve the optimal control parameters under a certain assumption or subset of the control problem.

- Other open research problems in legged locomotion: In recent years, there are lots of noticeable development in the field of dynamic robotics locomotion. However, many challenging problems are still unsolved. For example, for the problem of walking on rough terrain, lots of methods tend to focus on overcoming the terrain without taking into account energy efficiency. Due to the limitation of the robot's battery capacity, energy efficiency plays a crucial role to help the robot finishing the task before running out of power. Our approach on two-step periodic gait library offers an energy-efficient walking on discrete terrain. However, further development for different types of rough terrains should be deployed. In addition to that, some other open problems can be listed as follows: 3D dynamic walking with agile turning, 3D running and 3D running on rough terrains.

Bibliography

[1] "The W-Prize on stepping stones," *http://www.wprize.org/SteppingStones.html*.

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

[3] A. D. Ames, J. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *IEEE Conference on Decision and Control*, 2014, pp. 6271–6278.

[4] A. D. Ames, K. Galloway, J. W. Grizzle, and K. Sreenath, "Rapidly Exponentially Stabilizing Control Lyapunov Functions and Hybrid Zero Dynamics," *IEEE Trans. Automatic Control*, vol. 59, no. 4, pp. 876–891, 2014.

[5] D. Angeli, E. Sontag, and Y. Wang, "A characterization of integral input-to-state stability," *IEEE Transactions on Automatic Control*, vol. 45, pp. 1082–1097, 2000.

[6] M. Benallegue and J.-P. Laumond, "Metastability for high-dimensional walking systems on stochastically rough terrain," in *Robotics: Science and Systems*, 2013.

[7] K. Byl and R. Tedrake, "Metastable walking machines," *The International Journal of Robotics Research*, vol. 28, no. 8, pp. 1040–1064, August 2009.

[8] C. Cai and A. R. Teel, "Characterizations of input-to-state stability for hybrid systems," *Systems & Control Letters*, vol. 58, pp. 47–53, 2009.

[9] C. Cao and N. Hovakimyan, "Stability margins of l1 adaptive controller: Part ii," *Proceedings of American Control Conference*, pp. 3931–3936, 2007.

[10] C. Cao and N. Hovakimyan, "Design and analysis of a novel l1 adaptive controller with guaranteed transient performance," *IEEE Transactions on Automatic Control*, vol. 53, no.2, pp. 586–591, March, 2008.

[11] C.Cao and N. Hovakimyan, "L1 adaptive output-feedback controllers for non-strictly-positive-real reference systems: Missile longitudinal autopilot design." *Journal of Guidance, Control, and Dynamics*, vol. 32, pp. 717–726, 2009.

[12] C.Cao and N.Hovakimyan, "L1 adaptive controller for a class of systems with unknown nonlinearities: Part i," *American Control Conference, Seattle, WA*, pp. 4093–4098, 2008.

[13] J. Chestnutt, J. Kuffner, K. Nishiwaki, and S. Kagami, "Planning biped navigation strategies in complex environments," in *IEEE International Conference on Humanoid Robotics*, 2003, pp. 117–123.

[14] J. Chestnutt, M. Lau, G. Cheung, J. Kuffner, J. Hodgins, and T. Kanade, "Footstep planning for the honda asimo humanoid," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation.*, pp. 629 – 634, 2005.

[15] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. R. Westervelt, C. Canudas-de-Wit, and J. W. Grizzle, "RABBIT: A testbed for advanced control theory," vol. 23, no. 5, pp. 57–79, October 2003.

[16] M. J. D. R. A. S. A. A. Christian Hubicki, Jesse Grimes and J. Hurst, "Atrias: Design and validation of a tether-free 3d-capable spring-mass bipedal robot," *International Journal of Robotics Research (IJRR)*.

[17] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.

[18] S. H. Collins, M. Wisse, and A. Ruina, "A three-dimensional passive-dynamic walking robot with two legs and knees," *The International Journal of Robotics Research*, vol. 20, no. 7, pp. 607–615, 2001.

[19] X. Da, O. Harib, R. Hartley, B. Griffin, and J. Grizzle, "From 2d design of underactuated bipedal gaits to 3d implementation: Walking with speed tracking," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2016.

[20] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization." *Proceedings of the 2014 IEEE/RAS International Conference on Humanoid Robots*, pp. 279–286, 2014.

[21] R. L. H. Deits, "Convex segmentation and mixed-integer footstep planning for a walking robot," Master's thesis, Massachusetts Institute of Technology, 2014.

[22] S. V. Drakunov and V. I. Utkin, "Sliding mode control in dynamic systems," *International Journal of Control*, vol. 55, pp. 1029–1037, 1992.

[23] C. Edwards and S. Spurgeon, *Sliding Mode Control: Theory And Applications*. CRC Press, 1998.

[24] R. A. Freeman and P. V. Kokotović, *Robust Nonlinear Control Design*. Birkhäuser, 1996.

[25] K. Galloway, K. Sreenath, A. D. Ames, and J. W. Grizzle, "Torque saturation in bipedal robotic walking through control lyapunov function based quadratic programs," *IEEE Access*, vol. PP, no. 99, p. 1, April 2015.

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

[27] K. A. Hamed and J. W. Grizzle, "Robust event-based stabilization of periodic orbits for hybrid systems: Application to an underactuated 3d bipedal robot," in *American Control Conference*, 2013.

[28] K. A. Hamed and J. W. Grizzle, "Iterative robust stabilization algorithm for periodic orbits of hybrid dynamical systems: Application to bipedal running," in *IFAC Analaysis and Design of Hybrid Systems*, 2015.

[29] H. G. J. W. H. Hamid Reza Vejdani, Albert Wu, "Touch-down angle control for spring-mass walking," *IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA*, 2015.

[30] 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 *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[31] A. Hereid, C. M. Hubicki, E. A. Cousineau, J. W. Hurst, and A. D. Ames, "Hybrid zero dynamics based multiple shooting optimization with applications to robotic walking," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 5734–5740.

[32] J. P. Hespanhaa, D. Liberzon, and A. R. Teel, "Lyapunov conditions for input-to-state stability of impulsive systems," *Automatica*, vol. 44, pp. 2735–2744, 2008.

[33] J. K. Hodgins and M. Raibert, "Adjusting step length for rough terrain locomotion," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 289–298, 1991.

[34] S.-C. Hsu, X. Xu, and A. D. Ames, "Control barrier function based quadratic programs with application to bipedal robotic walking," in *American Control Conference*, 2015, pp. 4542–4548.

[35] M. S. Jones, "Optimal control of an underactuated bipedal robot," Master's thesis, Oregon State University, ScholarsArchive@OSU, 2014.

[36] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, pp. 1620 – 1626, 2003.

[37] K. Karydis, I. Poulakakis, J. Sun, and H. G. Tanner, "Probabilistically valid stochastic extensions of deterministic models for systems with uncertainty," *The International Journal of Robotics Research*, vol. 34, no. 10, pp. 1278–1295, 2015.

[38] M. Kelly and A. Ruina, "Non-linear robust control for inverted-pendulum 2d walking," *IEEE International Conference on Robotics and Automation*, 2015.

[39] H. Khalil, *Nonlinear Systems - 3rd Edition*, Upper Saddle River, NJ, 2002.

[40] S. Kolavennu, S. Palanki, and J. Cockburn, "Nonlinear control of nonsquare multivariable systems," *Chemical Engineering Science*, vol. 56, p. 2103âĂŞ2110, 2001.

[41] J. J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Footstep planning among obstacles for biped robots," *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems,*, vol. 1, pp. 500 – 505, 2001.

[42] E. Lavretsky, T. E. Gibson, and A. M. Annaswamy, "Projection operator in adaptive systems," *arXiv:1112.4232v6*, 2012.

[43] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*. John Wiley & Sons, 2012.

[44] Z. Li and S. S. Ge, "Adaptive robust controls of biped robots," *IET Control Theory and Applications*, vol. 7, no. 2, pp. 161–175, 2013.

[45] F. Lin and R. D. Brandt, "An optimal control approach to robust control of robot manipulators," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 1, pp. 69–77, February 1998.

[46] A. Majumdar and R. Tedrake, "Funnel libraries for real-time robust feedback motion planning," *Under Review*, 2016.

[47] J. Mattingley and S. Boyd, "CVXGEN: a code generator for embedded convex optimization," *Optimization and Engineering*, vol. 13, no. 1, pp. 1–27, March 2012.

[48] T. McGeer *et al.*, "Passive dynamic walking," *I. J. Robotic Res.*, vol. 9, no. 2, pp. 62–82, 1990.

[49] A. Mehra, W.-L. Ma, F. Berg, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Adaptive cruise control: Experimental validation of advanced controllers on scale-model cars," in *American Control Conference*, 2015, pp. 1411–1418.

[50] P. Michel, J. Chestnutt, J. Kuffner, and T. Kanade, "Vision-guided humanoid footstep planning for dynamic environments," in *Humanoids*, 2005, pp. 13–18.

[51] B. Morris and J. Grizzle, "A restricted Poincaré map for determining exponentially stable periodic orbits in systems with impulse effects: Application to bipedal robots," in *44th IEEE Conference on Decision and Control*, Seville, Spain, December 2005, pp. 4199–4206.

[52] B. Morris, E. Westervelt, C. Chevallereau, G. Buche, and J. Grizzle, *Fast Motions Symposium on Biomechanics and Robotics*, ser. Lecture Notes in Control and Information Sciences. Heidelberg, Germany: Springer-Verlag, 2006, ch. Achieving Bipedal Running with RABBIT: Six Steps toward Infinity, pp. 277–297.

[53] B. Morris, M. J. Powell, and A. D. Ames, "Sufficient conditions for the Lipschitz continuity of QP-based multi-objective control of humanoid robots," *Proc. 52nd IEEE Conf. Decision and Control*, pp. 2920–2926, 2013.

[54] M. S. Motahar, S. Veer, and I. Poulakakis, "Composing limit cycles for motion planning of 3d bipedal walkers," in *the IEEE Conference On Decision and Control (CDC)*, 2016.

[55] Q. Nguyen, X. Da, J. W. Grizzle, and K. Sreenath, "Dynamic walking on stepping stones with gait library and control barrier," *Workshop on Algorithimic Foundations of Robotics*, 2016.

[56] Q. Nguyen and K. Sreenath, "L1 adaptive control for bipedal robots withcontrol lyapunov function based quadratic programs," in *American Control Conference*, 2015, pp. 862–867.

[57] Q. Nguyen and K. Sreenath, "Optimal robust control for bipedal robots through control lyapunov function based quadratic programs," in *Robotics: Science and Systems*, 2015.

[58] Q. Nguyen and K. Sreenath, "Safety-critical control for dynamical bipedal walking with precise footstep placement," in *The IFAC Conference on Analysis and Design of Hybrid Systems*, 2015, pp. 147–154.

[59] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *American Control Conference*, 2016.

[60] Q. Nguyen and K. Sreenath, "Optimal robust control for constrained nonlinear hybrid systems with application to bipedal locomotion," in *American Control Conference*, 2016.

[61] Q. Nguyen and K. Sreenath, "3d dynamic walking on stepping stones with control barrier functions," in *2016 IEEE 55th Conference on Decision and Control*, Submitted, 2016.

[62] V. V. Patel, C. Cao, N. Hovakimyan, K. A. Wise, and E. Lavretsky, "L1 adaptive controller for tailless unstable aircraft in the presence of unknown actuator failures," *International Journal of Control*, vol. 82, pp. 705–720, 2009.

[63] M. H. Raibert, *Legged robots that balance*. MIT press, 1986.

[64] A. Ramezani, J. W. Hurst, K. Akbari Hamed, and J. W. Grizzle, "Performance Analysis and Feedback Control of ATRIAS, A Three-Dimensional Bipedal Robot," *Journal of Dynamic Systems, Measurement, and Control*, vol. 136, no. 2, 2014.

[65] C. O. Saglam and K. Byl, "Meshing hybrid zero dynamics for rough terrain walking," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 5718–5725.

[66] M. J. A. P. J. V. W. A. A. Siavash Rezazadeh, Christian Hubicki and J. W. Hurst, "Spring-mass walking with atrias in 3d: Robust gait control spanning zero to 4.3 kph on a heavily underactuated bipedal robot," *ASME Dynamic Systems and Control Conference (DSCC)*.

[67] A. Siravuru, A. Wang, Q. Nguyen, and K. Sreenath, "Deep visual perception for dynamic walking on discrete terrain," in *IEEE-RAS International Conference on Humanoid Robots*, 2017.

[68] E. D. Sontag, "A 'universal' construction of Artstein theorem on nonlinear stabilization," *Systems & Control Letters*, vol. 13, pp. 117–123, 1989.

[69] E. D. Sontag, "On the input-to-state stability property," *European Journal of Control*, vol. 1, pp. 24–36, 1995.

[70] E. D. Sontag and Y. Wang, "On characterizations of the input-to-state stability property," *Systems & Control Letters*, vol. 24, pp. 351–359, 1995.

[71] M. W. Spong, "On the robust control of robot manipulators," *IEEE Transactions on Automatic Control*, vol. 37, no. 11, pp. 1782–1786, November 1992.

[72] K. Sreenath, H. Park, I. Poulakakis, and J. Grizzle, "A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL," *IJRR*, vol. 30, pp. 1170–1193, 2011.

[73] R. Tedrake, T. W. Zhang, M.-f. Fong, and H. S. Seung, "Actuating a simple 3d passive dynamic walker," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 5. IEEE, 2004, pp. 4656–4661.

[74] V. A. Tucker, "The energetic cost of moving about: walking and running are extremely inefficient forms of locomotion. much greater efficiency is achieved by birds, fishâĂŤand bicyclists," *American Scientist*, vol. 63, no. 4, pp. 413–419, 1975.

[75] L. Vu, D. Chatterjee, and D. Liberzon, "Input-to-state stability of switched systems and switching adaptive control," *Automatica*, vol. 43, pp. 639–646, 2007.

[76] M. Vukobratović and B. Borovac, "Zero-moment pointâĂŤthirty five years of its life," *International journal of humanoid robotics*, vol. 1, no. 01, pp. 157–173, 2004.

[77] H. G. W Martin, A Wu, "Experimental evaluation of deadbeat running on the atrias biped," *Robotics and Automation Letters*, 2017.

[78] 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," vol. 24, no. 6, pp. 559–582, June 2004.

[79] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. Choi, and B. Morris, *Feedback Control of Dynamic Bipedal Robot Locomotion*, ser. Control and Automation. Boca Raton, FL: CRC, June 2007.

[80] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback control of dynamic bipedal robot locomotion*. CRC press Boca Raton, 2007.

[81] Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Bilinear_interpolation

[82] G. Wu and K. Sreenath, "Safety-critical and constrained geometric control synthesis using control lyapunov and control barrier functions for systems evolving on manifolds," in *American Control Conference*, 2015, pp. 2038–2044.

[83] G. Wu and K. Sreenath, "Safety-critical control of a planar quadrotor," in *American Control Conference*, 2016.

[84] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," in *The IFAC Conference on Analysis and Design of Hybrid Systems*, 2015, pp. 54–61.

[85] T. Yang, E. Westervelt, A. Serrani, and J. P. Schmiedeler, "A framework for the control of stable aperiodic walking in underactuated planar bipeds," *Autonomous Robots*, vol. 27, no. 3, pp. 277–290, 2009.

[86] M. Zhihong, A. Paplinski, and H. Wu, "A robust mimo terminal sliding mode control scheme for rigid robotic manipulators," *IEEE Transactions on Automatic Control*, vol. 39, pp. 2464–2469, 1994.

[87] K. Zhou, J. Doyle, and K. Glover, *Robust and Optimal Control*. Prentice Hall PTR, 1996.

Appendix A

Derivation of QP based controllers

In this Appendix, we will describe in details about how to formulate our proposed controller in a standard quadratic program formulation.

A.1 CLF-QP

The CLF-QP controller with constraints (3.33) can be written as the following standard form of quadratic programs:

CLF-QP with Constraints:

$$
\begin{aligned}
u^*(x) = \underset{u,\mu,\delta}{\operatorname{argmin}} \quad & \mu^T\mu + p\delta^2 && \text{(A.1)}\\
\text{s.t.} \quad & A_{CLF}\begin{bmatrix}\mu\\ \delta\end{bmatrix} \leq b_{CLF}, && \textbf{(CLF)}\\
& A_c(x)u \leq b_c(x), && \textbf{(Constraints)}\\
& u = u_{ff}(x) + (L_gL_fy(x))^{-1}\mu, && \textbf{(IO)}
\end{aligned}
$$

---

where

$$
A_{CLF} = \begin{bmatrix} L_{\bar{g}}V_\varepsilon & -1 \end{bmatrix}, \quad b_{CLF} = -L_{\bar{f}}V_\varepsilon - \frac{c_3}{\varepsilon}V_\varepsilon. \tag{A.2}
$$

A.2 CBF-CLF-QP

The QP form of CBF-CLF-QP controller in (5.8) is given by:

CBF-CLF-QP with Constraints:

$$
\begin{aligned}
u^*(x) = \underset{u,\mu,\delta}{\operatorname{argmin}} \quad & \mu^T\mu + p\,\delta^2 && \text{(A.3)}\\
\text{s.t.} \quad & A_{CLF}\begin{bmatrix}\mu\\ \delta\end{bmatrix} \le b_{CLF}, && \textbf{(CLF)}\\
& A_{CBF}\,u \le b_{CBF} && \textbf{(CBF)}\\
& A_c(x)u \le b_c(x). && \textbf{(Constraints)}\\
& u = u_{ff}(x) + (L_gL_fy(x))^{-1}\mu, && \textbf{(IO)}
\end{aligned}
$$

---

where A_{CLF}, b_{CLF} are as defined in (A.2), and

$$A_{CBF} = L_gB, \quad b_{CBF} = \frac{\gamma}{B} - L_fB. \tag{A.4}$$

A.3 Robust CLF-QP with constraints

Notice that the robust RES condition in (4.11) is affine in the control inputs μ and thus can be expressed as

$$\max_{\substack{||\Delta_1||\le\Delta_1^{max}\\ ||\Delta_2||\le\Delta_2^{max}}} \Psi_0 + \Psi_1\mu \le 0, \tag{A.5}$$

where

$$
\begin{aligned}
\Psi_0(\eta,\Delta_1) &:= L_{\bar{f}}V_\varepsilon(\eta) + \frac{c_3}{\varepsilon}V_\varepsilon,\\
\Psi_1(\eta,\Delta_2) &:= L_{\bar{g}}V_\varepsilon(\eta),
\end{aligned} \tag{A.6}
$$

with $L_{\bar{f}}V_\varepsilon(\eta), L_{\bar{g}}V_\varepsilon(\eta)$ as defined in (4.8).

The goal of the robust control design is then to find the control μ satisfying the RES condition (A.5), evaluated through the given bounds of uncertainty in (4.10). However, our controller can only use the

nominal model $\tilde{\bar{f}}, \tilde{\bar{g}}$ and not the true model $\bar{f}, \bar{g}$ (see Table 4.1 for different types of models). We thus define the Lie derivatives of V with respect to the nominal model as

$$L_{\tilde{\bar{f}}}V = \eta^T(F^T P_\varepsilon + P_\varepsilon F)\eta, \tag{A.7}$$

$$L_{\tilde{\bar{g}}}V = 2\eta^T P_\varepsilon G. \tag{A.8}$$

Then, from (4.8), we have

$$\dot{V} = L_{\bar{f}}V + L_{\bar{g}}V\mu \tag{A.9}$$

$$= L_{\tilde{\bar{f}}}V + \underbrace{2\eta^T P_\varepsilon \begin{bmatrix} \mathbf{0} \\ \Delta_1 \end{bmatrix}}_{\Delta_1^v} + \underbrace{L_{\tilde{\bar{g}}}V\mu}_{\mu_v} + \underbrace{L_{\tilde{\bar{g}}}\Delta_2\mu}_{\Delta_2^v\mu_v} \tag{A.10}$$

$$= L_{\tilde{\bar{f}}}V + \Delta_1^v + (1+\Delta_2^v)\mu_v, \tag{A.11}$$

where we have defined the following new scalar variables: uncertainty $\Delta_1^v \in \mathbb{R}$, control input $\mu_v \in \mathbb{R}$, and uncertainty $\Delta_2^v \in \mathbb{R}$. Since Δ_1^v is the model uncertainty on $L_{\bar{f}}V$ and Δ_2^v is the model uncertainty on the mapping of the control input μ_v, we then can simply design robust CLF condition using the assumption on the bounds of Δ_1^v and Δ_2^v. However, based on the model uncertainty analysis in (4.8), in order to consider the presence of the state η in the model uncertainty, we will evaluate Δ_1^v using the following assumption:

$$||\Delta_1^v|| \leq ||2\eta^T P_\varepsilon \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix} ||\Delta_1^{max}. \tag{A.12}$$

This approach is frequently used in adaptive control to improve the performance of the adaptation law (see [56], [10]). Since μ_v includes $L_{\bar{g}}V = 2\eta^T P_\varepsilon G$, and from (4.8), we therefore consider the following assumption for Δ_2^v:

$$||\Delta_2^v|| \leq \Delta_2^{max} \tag{A.13}$$

Therefore, the RES condition (A.5) will hold if the following inequalities hold

$$\begin{aligned} \Psi_0^{max} + \Psi_1^p \mu \leq 0, \\ \Psi_0^{max} + \Psi_1^n \mu \leq 0, \end{aligned} \tag{A.14}$$

where

$$\Psi_0^{max} = L_{\tilde{f}} V + \frac{c_3}{\varepsilon} V_\varepsilon + ||2\eta^T P_\varepsilon \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix} ||\Delta_1^{max},$$
$$\Psi_1^p = L_{\tilde{g}} V(1 + \Delta_2^{max}),$$
$$\Psi_1^n = L_{\tilde{g}} V(1 - \Delta_2^{max}). \tag{A.15}$$

Thus the robust CLF-QP in (4.12) is then equivalent to the following optimization problem, which can clearly be seen as a QP:

$$\begin{aligned} \mu^* = \underset{\mu}{\text{argmin}} \quad & \mu^T \mu \\ \text{s.t.} \quad & \Psi_0^{max}(\eta, \Delta_1^{max}) + \Psi_1^p(\eta, \Delta_2^{max})\, \mu \leq 0, \\ & \Psi_0^{max}(\eta, \Delta_1^{max}) + \Psi_1^n(\eta, \Delta_2^{max})\, \mu \leq 0. \end{aligned} \tag{A.16}$$

Incorporating the above robust CLF constraint we have the following Robust CLF-QP with constraints:

Robust CLF-QP with Constraints:

$$\begin{aligned} u^*(x) = \underset{u,\mu,\delta_1,\delta_2}{\text{argmin}} \quad & \mu^T \mu + p_1 \delta_1^2 + p_2 \delta_2^2 \\ \text{s.t.} \quad & \Psi_0^{max}(\eta, \Delta_1^{max}) + \Psi_1^p(\eta, \Delta_2^{max})\, \mu \leq \delta_1, \\ & \Psi_0^{max}(\eta, \Delta_1^{max}) + \Psi_1^n(\eta, \Delta_2^{max})\, \mu \leq \delta_2. && \textbf{(Robust CLF)} \\ & A_c(x) u \leq b_c(x), && \textbf{(Constraints)} \\ & u = u_{ff}(x) + (L_g L_f y(x))^{-1} \mu. && \textbf{(IO)} \end{aligned} \tag{A.17}$$

The above can be represented in a similar form to (A.1), thus representing a quadratic program.

A.3.1 Robust CBF-CLF-QP

Note that the robust CBF condition in (8.15) is affine in μ_b and can be expressed as

$$\max_{\substack{||\Delta_1^b|| \leq \Delta_{1,max}^b \\ ||\Delta_2^b|| \leq \Delta_{2,max}^b}} \Psi_0^b + \Psi_1^b \mu_b \leq 0 \tag{A.18}$$

where "b" refers to CBF constraint in the QP, and

$$\Psi_0^b(x, \Delta_1^b) := \Delta_1^b - \frac{\gamma}{B(x)},$$
$$\Psi_1^b(x, \Delta_2^b) := 1 + \Delta_2^b, \tag{A.19}$$

where the above arises due to the time-derivative of the CBF from (8.8).

Since Ψ_0^b, Ψ_1^b are affine with respect to Δ_1^b, Δ_2^b, and with the assumptions on the bounds on the uncertainty in (8.13), the robust CBF condition (A.18) will hold if the following two inequalities hold

$$\Psi_{0,max}^b(x) + \Psi_{1,p}^b(x)\mu_b \leq 0,$$
$$\Psi_{0,max}^b(x) + \Psi_{1,n}^b(x)\mu_b \leq 0. \tag{A.20}$$

where

$$\Psi_{0,max}^b := \Delta_{1,max}^b - \frac{\gamma}{B(x)}, \tag{A.21}$$
$$\Psi_{1,p}^b := 1 + \Delta_{2,max}^b, \tag{A.22}$$
$$\Psi_{1,n}^b := 1 - \Delta_{2,max}^b. \tag{A.23}$$

We then can incorporate the above robust CBF conditions into the robust CLF-QP (A.17) resulting in a quadratic program.

A.4 Robust Constraints

Note that the robust constraints condition in (8.23) is affine in μ_c and can be expressed as

$$\max_{\substack{||\Delta_1^c|| \leq \Delta_{1,max}^c \\ ||\Delta_2^c|| \leq \Delta_{2,max}^c}} \Psi_0^c + \Psi_1^c \mu_c \leq 0 \tag{A.24}$$

where "c" refers to constraints, and

$$\Psi_0^c(x, \Delta_1^c) := \Delta_1^c,$$
$$\Psi_1^c(x, \Delta_2^c) := 1 + \Delta_2^c, \tag{A.25}$$

where the above arises due to the form of the constraints in (8.19). The same procedure as what was done for the robust CBF in the previous section can be carried out for the robust constraints as well to show how

the max problem gets converted to a set of linear inequalities and thus the min-max problem then becomes a quadratic program even for the robust constraints as well.

Appendix B

Sufficient conditions for the stability of CLF with relaxed inequality

In this Appendix, we will present two theorems and their proof about the stability of CLF based controller with relaxed RES-CLF condition for both continuous-time and hybrid systems.

We begin with the standard RES-CLF that guarantees the following inequality,

$$\dot{V}_\varepsilon(\eta, \mu) + \frac{c_3}{\varepsilon} V_\varepsilon(\eta) \leq 0. \tag{B.1}$$

The CLF-QP with the relaxed inequality takes the form,

$$\dot{V}_\varepsilon(\eta, \mu) + \frac{c_3}{\varepsilon} V_\varepsilon(\eta) \leq d_\varepsilon, \tag{B.2}$$

where $d_\varepsilon(t) \geq 0$ represents the time-varying relaxation of the RES-CLF condition. We define

$$w_\varepsilon(t) = \int_0^t \frac{d_\varepsilon(\tau)}{V_\varepsilon} \, \mathrm{d}\tau. \tag{B.3}$$

to represent a scaled version of the total relaxation up to time t. In the following subsections, we make use of this quantity to establish exponential stability under certain conditions for both continuous-time and hybrid systems. In Appendix B.1 we will look at continuous-time systems where we will need exponential stability ($\varepsilon = 1$ in the above formulations), while in Appendix B.2 we will look at hybrid systems where we need rapid exponential stability ($\varepsilon < 1$).

B.1 Stability of the Relaxed CLF-QP Controller for Continuous-Time Systems

Consider a control affine system

$$\begin{aligned} \dot{\eta} &= \bar{f}(\eta) + \bar{g}(\eta)\mu, \\ \dot{z} &= p(\eta, z), \end{aligned} \tag{B.4}$$

where η is the controlled (or output) state, and z is the uncontrolled state. Let $\mathscr{O}_Z$ be an exponentially stable periodic orbit for the zero dynamics $\dot{z} = p(0, z)$. As stated in [4, Theorem 1], for all control inputs $\mu(\eta, z)$ that guarantee enforcing the ES-CLF condition (with $\varepsilon = 1$) (B.1), we have that $\mathscr{O} = \iota_0(\mathscr{O}_Z)$ is an exponentially stable periodic orbit of (B.4). Then, for the relaxed CLF-QP condition (B.2) ($d_\varepsilon \not\equiv 0$), the following theorem establishes sufficient conditions under which the exponential stability of the periodic orbit still holds.

Theorem B.1. *Consider a nonlinear control affine system* (B.4)*. Let $\mathscr{O}_Z$ be an exponentially stable periodic orbit for the zero dynamics $\dot{z} = p(0, z)$. Then $\mathscr{O} = \iota_0(\mathscr{O}_Z)$ is an exponentially stable periodic orbit of* (B.4)*, if*

$$\bar{w}_\varepsilon := \sup_{t \geq 0} \; w_\varepsilon(t) \tag{B.5}$$

is a finite number.

Proof. Note that if $d_\varepsilon(t) \equiv 0$, i.e., there is no violation on the RES-CLF condition, we will have conventional exponential stability as stated in [4]. Here, we will extend the proof of exponential stability to the case of relaxation of the control Lyapunov function condition when $\bar{w}_\varepsilon$ is finite.

We begin by noting that since $d_\varepsilon(t) \geq 0$, we have

$$w_\varepsilon(t) \leq \bar{w}_\varepsilon, \; \forall t \geq 0. \tag{B.6}$$

Next, from (B.2), we have

$$\begin{aligned} \frac{dV_\varepsilon}{dt} &\leq -\frac{c_3}{\varepsilon} V_\varepsilon + d_\varepsilon(t), \\ \Rightarrow \; \frac{dV_\varepsilon}{V_\varepsilon} &\leq -\frac{c_3}{\varepsilon} dt + \frac{d_\varepsilon(t)}{V_\varepsilon} dt, \\ \Rightarrow \; ln\left(\frac{V_\varepsilon(t)}{V_\varepsilon(0)}\right) &\leq -\frac{c_3}{\varepsilon} t + \int_0^t \frac{d_\varepsilon(\tau)}{V_\varepsilon} \, \mathrm{d}\tau, \\ \Rightarrow \; V_\varepsilon(t) &\leq e^{-\frac{c_3}{\varepsilon} t + w_\varepsilon(t)} V_\varepsilon(0). \end{aligned} \tag{B.7}$$

This, in combination with the inequality in (3.19), then implies that

$$\|\eta(t)\| \leq \sqrt{\frac{c_2}{c_1}}\frac{1}{\varepsilon}e^{-\frac{c_3}{2\varepsilon}t+\frac{1}{2}w_\varepsilon(t)}\|\eta(0)\|. \tag{B.8}$$

Moreover, from the inequality in (B.6), we have

$$\begin{aligned}\|\eta(t)\| &\leq \sqrt{\frac{c_2}{c_1}}\frac{1}{\varepsilon}e^{-\frac{c_3}{2\varepsilon}t+\frac{1}{2}\bar{w}_\varepsilon}\|\eta(0)\|, \\ &= \left(\sqrt{\frac{c_2}{c_1}}\frac{1}{\varepsilon}e^{\frac{1}{2}\bar{w}_\varepsilon}\right)e^{-\frac{c_3}{2\varepsilon}t}\|\eta(0)\|. \end{aligned} \tag{B.9}$$

Therefore, if $\bar{w}_\varepsilon$ is finite, the control output η will still be exponentially stable under the relaxed CLF condition (B.2). Therefore, from [4, Theorem 1], $\mathscr{O} = \iota_0(\mathscr{O}_Z)$ is an exponentially stable periodic orbit of (B.4).

Remark B.1. *It must be noted that the CLF V_ε no longer provides a guarantee of exponential stability due to the relaxation in* (B.2)*. However due to the inequality established in* (B.9)*, and by the converse Lyapunov function theorems [39], there exists another CLF $\tilde{V}_\varepsilon$ that satisfies $\tilde{c}_1||\eta||^2 \leq \tilde{V}_\varepsilon(\eta) \leq \frac{\tilde{c}_2}{\varepsilon^2}||\eta||^2$, and $\dot{\tilde{V}}_\varepsilon(\eta,\mu) + \frac{\tilde{c}_3}{\varepsilon}\tilde{V}_\varepsilon(\eta) \leq 0$, for some positive constants $\tilde{c}_1, \tilde{c}_2, \tilde{c}_3$, and guarantees exponential stability.*

□

In the next section, we consider the hybrid system with impulse effects, wherein we will need to take into account the impact time or the switching time that signifies the end of the continuous-time phase and involves a discrete-time jump in the state.

B.2 Stability of the Relaxed CLF-QP Controller for Hybrid Systems

Here we look at the stability of the relaxed CLF-QP controller for hybrid systems of the form as defined in (3.13) without the restriction on the vector fields as in (3.14). Similar to the stability analysis for continuous-time systems in the previous section, we also use here the notions of $d_\varepsilon(t)$ (B.2), the relaxation of the CLF condition, and $w_\varepsilon(t)$ (B.3), the scaled version of the total relaxation up to time t.

We also define $T_I^\varepsilon(\eta, z)$ to be the time-to-impact or time taken to go from the state (η, z) to the switching surface S, that signifies ending time of the current step for a bipedal walking robot modeled as a hybrid system. Then, intuitively, $w_\varepsilon(T_I^\varepsilon(\eta, z))$ indicates a scaled version of the total violation of the RES-CLF bound in (B.1) over one complete step. If $w_\varepsilon(T_I^\varepsilon(\eta, z)) \leq 0$, it implies that $V_\varepsilon(T_I^\varepsilon(\eta, z))$ is less than or equal to what would have resulted if the RES-CLF bound had not been violated at all. As we will see in the

following theorem, we will in fact only require $w_\varepsilon(T_I^\varepsilon(\eta, z))$ to be upper bounded by a positive constant for exponential stability.

We first define the hybrid zero dynamics as the hybrid dynamics (3.13) restricted to the surface Z in (3.11), i.e.,

$$\mathcal{H}_Z : \begin{cases} \dot{z} = p(0, z), & z \notin S \cap Z, \\ z^+ = \Delta_z(0, z^-), & z^- \in S \cap Z. \end{cases} \tag{B.10}$$

We then have the following theorem:

Theorem B.2. *Let $\mathcal{O}_Z$ be an exponentially stable hybrid periodic orbit of the hybrid zero dynamics $\mathscr{H}|_Z$* (B.10) *transverse to $S \cap Z$ and the continuous dynamics of $\mathscr{H}$* (3.13) *controlled by a CLF-QP with relaxed inequality* (3.33)*. Then there exists an $\bar{\varepsilon} > 0$ and an $\bar{w}_\varepsilon \geq 0$ such that for each $0 < \varepsilon < \bar{\varepsilon}$, if the solution $\mu_\varepsilon(\eta, z)$ of the CLF-QP* (3.33) *satisfies $w_\varepsilon(T_I^\varepsilon(\eta, z)) \leq \bar{w}_\varepsilon$, then $\mathcal{O} = \iota_0(\mathcal{O}_Z)$ is an exponentially stable hybrid periodic orbit of $\mathscr{H}$.*

B.3 Proof of Theorem B.2

A large part of this proof directly follows from results and proofs in [4], that is used to prove the stability of the hybrid system under the RES-CLF condition. In our case of relaxed CLF, we will state the additional condition under which the proof is still valid.

Let $\varepsilon > 0$ be fixed and select a Lipschitz continuous feedback u_ε of the relaxed CLF-QP controller (3.33). From [4, (56)], we have $T_I^\varepsilon(\eta, z)$ is continuous (since it is Lipschitz) and therefore there exists $\delta > 0$ and $\Delta T > 0$ such that for all $(\eta, z) \in B_\delta(0, 0) \cap S$,

$$T^* - \Delta T \leq T_I^\varepsilon(\eta, z) \leq T^* + \Delta T, \tag{B.11}$$

where T^* is the period of the orbit $\mathcal{O}_Z$.

In order to make use of the proof of the exponential stability for the standard RES-CLF controller in [4], we will present the condition for bounding the system states $\eta(t)$ at the time-to-impact $T_I^\varepsilon(\eta, z)$ in the following lemma.

Lemma B.1. *Let $\mathcal{O}_Z$ be an exponentially stable periodic orbit of the hybrid zero dynamics $\mathscr{H}|_Z$* (B.10) *transverse to $S \cap Z$ and the continuous dynamics of $\mathscr{H}$* (3.13) *controlled by a CLF-QP with relaxed inequality* (3.33)*. Then for each $\Delta T > 0$ and $\varepsilon > 0$, there exists an $\bar{w}_\varepsilon \geq 0$ such that, if the solution $u_\varepsilon(\eta, z)$*

of the CLF-QP with relaxed inequality (3.33) *satisfies* $w_\varepsilon(T_I^\varepsilon(\eta, z)) \leq \bar{w}_\varepsilon$*, then*

$$\|\eta(t)\|\Big|_{t=T_I^\varepsilon(\eta,z)} \leq \sqrt{\frac{c_2}{c_1}} \frac{2e^{-1}}{(T^* - \Delta T)c_3} \|\eta(0)\|. \tag{B.12}$$

Proof. From (B.8) and because $w_\varepsilon(T_I^\varepsilon(\eta, z)) \leq \bar{w}_\varepsilon$, it implies that

$$\begin{aligned} \|\eta(t)\|\Big|_{t=T_I^\varepsilon(\eta,z)} &\leq \sqrt{\frac{c_2}{c_1}} \frac{1}{\varepsilon} e^{-\frac{c_3}{2\varepsilon} T_I^\varepsilon(\eta,z) + \frac{1}{2} w_\varepsilon(T_I^\varepsilon(\eta,z))} \|\eta(0)\| \\ &\leq \sqrt{\frac{c_2}{c_1}} \frac{1}{\varepsilon} e^{-\frac{c_3}{2\varepsilon} T_I^\varepsilon(\eta,z) + \frac{1}{2} \bar{w}_\varepsilon} \|\eta(0)\| \end{aligned} \tag{B.13}$$

Then, from (B.11), we have,

$$\|\eta(t)\|\Big|_{t=T_I^\varepsilon(\eta,z)} \leq \sqrt{\frac{c_2}{c_1}} \frac{1}{\varepsilon} e^{-\frac{c_3}{2\varepsilon}(T^* - \Delta T) + \frac{1}{2} \bar{w}_\varepsilon} \|\eta(0)\|. \tag{B.14}$$

Furthermore, because $e^{-\alpha} \leq e^{-1}/\alpha, \forall \alpha \geq 0$, we have:

$$\frac{1}{\varepsilon} e^{-\frac{c_3}{2\varepsilon}(T^* - \Delta T)} \leq \frac{2e^{-1}}{(T^* - \Delta T)c_3}. \tag{B.15}$$

Then it implies that there exists a $\bar{c}_3 \leq c_3$ such that:

$$\frac{1}{\varepsilon} e^{-\frac{c_3}{2\varepsilon}(T^* - \Delta T)} \leq \frac{1}{\varepsilon} e^{-\frac{\bar{c}_3}{2\varepsilon}(T^* - \Delta T)} \leq \frac{2e^{-1}}{(T^* - \Delta T)c_3}. \tag{B.16}$$

The above inequality follows by the fact that $\frac{1}{\varepsilon} e^{-\frac{c_3}{2\varepsilon}(T^* - \Delta T)}$ is a monotonically decreasing function of c_3, and that for any real numbers $a \leq b$, there always exists a number $c \in [a : b]$ such that $a \leq c \leq b$. To be more specific, let $\bar{c}_3^a$ be the solution of:

$$\frac{1}{\varepsilon} e^{-\frac{\bar{c}_3^a}{2\varepsilon}(T^* - \Delta T)} = \frac{2e^{-1}}{(T^* - \Delta T)c_3}, \tag{B.17}$$

then any $\bar{c}_3 \in [\bar{c}_3^a : c_3]$ will satisfy (B.16).

We can then define

$$\bar{w}_\varepsilon := \frac{c_3 - \bar{c}_3}{\varepsilon}(T^* - \Delta T) \geq 0, \tag{B.18}$$

$$\Rightarrow -\frac{c_3}{2\varepsilon}(T^* - \Delta T) + \frac{1}{2}\bar{w}_\varepsilon = -\frac{\bar{c}_3}{2\varepsilon}(T^* - \Delta T). \tag{B.19}$$

Next, plugging (B.19) into (B.14), we have,

$$\|\eta(t)\|\Big|_{t=T_I^\varepsilon(\eta,z)} \leq \sqrt{\frac{c_2}{c_1}}\frac{1}{\varepsilon}e^{-\frac{\bar{c}_3}{2\varepsilon}(T^*-\Delta T)}\|\eta(0)\|, \tag{B.20}$$

$$V_\varepsilon(\eta(t))\big|_{t=T_I^\varepsilon(\eta,z)} \leq e^{-\frac{\bar{c}_3}{\varepsilon}(T^*-\Delta T)}V_\varepsilon(\eta(0)). \tag{B.21}$$

We now complete the proof of Lemma B.1 by substituting (B.16) into (B.20) to obtain (B.12). The solution of $\bar{w}_\varepsilon$ can be found in (B.18). □

Using Lemma B.1, we then can follow the same protocol of the proof in [4, Theorem 2] until equation [4, (64)].

We then define $\beta_1(\varepsilon) = \frac{c_2}{\varepsilon^2}L_{\Delta_X}^2 e^{-\frac{c_3}{\varepsilon}(T^*-\Delta T)}$ and $\bar{\beta}_1(\varepsilon) = \frac{c_2}{\varepsilon^2}L_{\Delta_X}^2 e^{-\frac{\bar{c}_3}{\varepsilon}(T^*-\Delta T)}$ where L_{Δ_X}, defined after [4, (59)], is the Lipschitz constant for Δ_X. Because $\beta_1(0^+) := \lim_{\varepsilon\searrow 0}\beta_1(\varepsilon) = 0$, then there exists an $\bar{\varepsilon}$ such that

$$\beta_1(\varepsilon) < c_1 \qquad \forall \qquad 0 < \varepsilon < \bar{\varepsilon}. \tag{B.22}$$

and for each ε, if we define $\bar{c}_3^b$ be the solution of

$$\frac{c_2}{\varepsilon^2}L_{\Delta_X}^2 e^{-\frac{\bar{c}_3^b}{\varepsilon}(T^*-\Delta T)} = c_1, \tag{B.23}$$

then for $\bar{c}_3 \in (\bar{c}_3^b : c_3]$, we have,

$$\beta_1(\varepsilon) \leq \bar{\beta}_1(\varepsilon) < c_1. \tag{B.24}$$

However, as presented in proof of Lemma B.1, $\bar{c}_3$ also needs to satisfy (B.16). Therefore, in order to guarantee the satisfaction of both (B.16) and (B.24), $\bar{c}_3$ needs to be chosen in the following set

$$\bar{c}_3 \in \{[\bar{c}_3^a : c_3] \cap (\bar{c}_3^b : c_3]\}, \tag{B.25}$$

where $\bar{c}_3^a$ and $\bar{c}_3^b$ are defined in (B.17) and (B.23) respectively.

The rest of the proof follows from the proof of [4, Theorem 2] using $\bar{\beta}_1$ instead of β_1. We finish our proof with the value of $\bar{w}_\varepsilon$ determined via (B.18), in which the feasible set of $\bar{c}_3$ is defined in (B.25).