

# Robust Safety-Critical Control for Dynamic Robotics

Quan Nguyen and Koushil Sreenath

**Abstract**—We present a novel method of optimal robust control through quadratic programs that offers tracking stability while subject to input and state-based constraints as well as safety-critical constraints for nonlinear dynamical robotic systems in the presence of 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 dynamic walking of a five-link planar bipedal robot subject to contact force constraints as well as safety-critical precise foot placements on stepping stones, all while subject to 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.

## I. INTRODUCTION

Designing controllers for complex robotic systems with nonlinear and hybrid dynamics for achieving stable high-speed tracking of reference trajectories while simultaneously guaranteeing input, state, and safety-critical constraints is challenging. Constraints on robotic systems arise as limits of the physical hardware (such as work-space constraints, joint position and velocity constraints, and motor torque constraints) as well as constraints imposed by controllers for safe operation of the system (such as collision constraints, range constraints, connectivity constraints, contact force constraints, etc.) Further, adding to the challenges of stable constrained control is the presence of high-levels of uncertainty in the dynamical model of the robot. The goal of this paper is to address the problem of designing stabilizing controllers for systems with strict constraints in the presence of model uncertainty.

### A. Background

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 [1]. Traditional CLF-based controllers involve closed-form control expressions such as the min-norm and the Sontag controllers [2]. Recently, a novel approach of expressing CLF-based controllers via online quadratic programs (QPs) in [3] opened an effective way for dealing with stability while also enabling the incorporation of

additional constraints, such as input constraints. In [3], 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 [4]. 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 [5]. This framework has also been extended to various interesting application domains, such as safety-critical geometric control for quadrotor systems [6] and safety-critical dynamic walking for bipedal robots [7], [8]. 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 [9], 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. Recent efforts to improve the robustness of CBFs using learning-based approaches such Gaussian process regression have been introduced in [? 10, 11, 12, 13] which use data to estimate the uncertainty. However, this typically requires large amounts of quality data for high-dimensional systems. Moreover, we note that in [? 11, 12], the authors assume uncertainty only in the drift vector field term of a control affine system and assume the input vector field to be known precisely. Nonetheless, with learning-based approaches, better and more accurate state-dependent bounds for the uncertainty can be learnt which could result in less conservative approaches than what is typically possible through robust control. However, on the other hand, such learning-based approaches need to assume various distributions to be Gaussian and do not computationally scale well with state-dimension. In this paper, we seek a method to simultaneously handle robust stability, robust input-based constraints and robust state-dependent constraint in the presence of model uncertainty. We will do this through robust control formulations of the CLF, CBF, and constraints. The framework can be easily achieved by extending the nominal CBF-CLF-QP controller while improv-

Q. Nguyen is with the Department of Aerospace and Mechanical Engineering, University of Southern California, Los Angeles, CA 90007, email: quann@usc.edu.

K. Sreenath is with the Department of Mechanical Engineering, University of California, Berkeley, CA 94720, email: koushils@berkeley.edu.

The work of Q. Nguyen is supported by USC Viterbi School of Engineering startup funds. The work of K. Sreenath was partially supported through National Science Foundation Grant IIS-1834557.

ing significantly the robustness of the entire approach with a provable stability and safety guarantee.

QP based controllers for robotic systems have been increasingly used in recent years. Especially, the DARPA Robotics Challenge inspired several new methods in this direction [14], [15], wherein QP controllers are used to formulate inverse kinematic and inverse dynamic control problems while minimizing tracking of desired accelerations. While these QP controllers enforce tracking performance and constraints at each time step, our method uses CLF and CBF condition to guarantee stability and safety constraints for future time.

For control of constrained systems, Model Predictive Control (MPC) has been widely used in many industrial applications [16]. However, the method is computationally expensive. Firstly, for nonlinear systems, MPC typically linearizes the model and the constraints. Next, even with a linear model and polyhedral constraints, the traditional approach can require a solving time from tens of milliseconds and higher even with recent advances [17]. For nonlinear control affine systems, CBF constraints are linear inequalities on the input and can be quickly enforced through quadratic programs that are solved point-wise in time in under 1 ms.

Robust control is an extensively studied topic. We have established methods, such as  $H_\infty$ -based robust control and linear quadratic Gaussian (LQG) based robust control [18, 19] 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 [20, 21, 22]) 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 [23, 24, 25]. 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 [26, 27, 28, 29]). 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 [30, 31], 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 [32] extends adaptive robust control of manipulator arms to bipedal robots, the work in [33] 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 [34] 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. Recent work in reachability-based analysis has been extended to enforce safety-constraints with model uncertainty by safe trajectory synthesis [35, 36]. Offline optimization for stabilization of walking and running with robustness to discrete-time uncertainties such as terrain perturbations has been carried out in [37, 38]. 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 [39, 40, 41, 42].

## B. Contributions

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

- Introduction of a new technique that optimally introduces robust control via quadratic programs to handle stability, input-based constraints and state-dependent constraints under high levels of model uncertainty.
- Robust stability and robust safety-critical constraints achieved through a min-max inequality constraint on the time-derivative of a control Lyapunov function and control barrier function respectively.
- Theoretical stability analysis for the QP controllers with relaxed CLF inequality.
- Numerical validation of the proposed controller on different problems:
  - Dynamic walking of a bipedal robot while carrying an unknown load and subject to contact force constraints; and
  - Dynamic walking of a bipedal robot while carrying an unknown load, subject to contact force constraints and precise foot-step location constraints.
- Experimental validation of the proposed control method on a rectilinear spring-cart system.

Note that preliminary results of this work were presented in [43, 44]. In contrast to the preliminary results, this paper presents (a) an entirely new min-max formulation of the proposed controller, (b) additional numerical examples for control of a bipedal robot, (c) experimental validations of the proposed controller on a rectilinear spring-cart system and (d) detailed stability and safety analysis for QP controllers with relaxed CLF inequality and the proposed robust CBF-CLF-QP controller.

## C. Organization

The rest of the paper is organized as follows. Section II revisits control barrier functions and control Lyapunov functions based quadratic programs (CBF-CLF-QPs). Section III presents the stability analysis for the QP controllers with relaxed CLF inequality. Section IV discusses the adverse effects of uncertainty and then presents the proposed optimal robust control and formulates as a QP. Section V presents

numerical and experimental validation on different dynamical robotic systems. Section VI provides concluding remarks.

## II. CONTROL LYAPUNOV FUNCTIONS AND CONTROL BARRIER FUNCTION BASED QUADRATIC PROGRAMS REVISITED

### A. Model and Input-Output Linearizing Control

Consider a nonlinear control affine hybrid model

$$\mathcal{H} : \begin{cases} \dot{x} = f(x) + g(x)u, & x \notin S \\ x^+ = \Delta(x^-), & x \in S \\ y = y(x), \end{cases} \quad (1)$$

where  $x \in \mathbb{R}^n$  is the system state,  $u \in \mathbb{R}^m$  is the control input,  $S$  is the switching surface, and  $y \in \mathbb{R}^m$  is a set of outputs.

While our approach is can be applied for a general high relative-degree system, we focus on the most common case to mechanical systems with relative-degree 2 control output. 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:

$$\ddot{y} = \frac{\partial \dot{y}}{\partial x} \dot{x} = L_f^2 y(x) + L_g L_f y(x)u, \quad (2)$$

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

$$L_f^2 y(x) \triangleq \frac{\partial \dot{y}}{\partial x} f(x), \quad L_g L_f y(x) \triangleq \frac{\partial \dot{y}}{\partial x} g(x). \quad (3)$$

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

$$u(x, \mu) = u_{ff}(x) + (L_g L_f y(x))^{-1} \mu, \quad (4)$$

with the feed-forward control input

$$u_{ff}(x) = -(L_g L_f y(x))^{-1} L_f^2 y(x), \quad (5)$$

input-output linearizes the system. The dynamics of the system (1) 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

$$\mathcal{Z} = \{x \in \mathbb{R}^n \mid \eta(x) \equiv 0\}. \quad (6)$$

One choice for the transverse variables is,

$$\eta = \begin{bmatrix} y(x) \\ \dot{y}(x) \end{bmatrix}. \quad (7)$$

The input-output linearized hybrid system then is,

$$\mathcal{H}^{IO} : \begin{cases} \dot{\eta} = \bar{f}(\eta) + \bar{g}(\eta)\mu, \\ \dot{z} = p(\eta, z), & (\eta, z) \notin S, \\ \eta^+ = \Delta_\eta(\eta^-, z^-), \\ z^+ = \Delta_z(\eta^-, z^-), & (\eta^-, z^-) \in S. \\ y = y(\eta), \end{cases} \quad (8)$$

where  $z$  represents uncontrolled states [45], and

$$\bar{f}(\eta) = F\eta, \quad \bar{g}(\eta) = G, \quad (9)$$

with,

$$F = \begin{bmatrix} O & I \\ O & O \end{bmatrix} \text{ and } G = \begin{bmatrix} O \\ I \end{bmatrix}. \quad (10)$$

The linear system in (9) is in controllable canonical form, and a linear controller such as  $\mu = -K\eta$  can be designed such that the following closed-loop system is stable

$$\dot{\eta} = (F - GK)\eta = A_{cl}\eta. \quad (11)$$

A corresponding quadratic Lyapunov function  $V(\eta) = \eta^T P \eta$ , with  $P$  being a symmetric positive definite matrix, can then be established through the solution of the Lyapunov equation:

$$A_{cl}^T P + P A_{cl} + Q = 0, \quad (12)$$

for  $Q$  being any symmetric positive definite matrix.

### B. Control Lyapunov Function based Quadratic Programs

1) *CLF-QP*: Instead of a linear control design  $\mu = -K\eta$  in (4), an alternative control design is through a control Lyapunov function  $V(\eta)$ , wherein a control is chosen pointwise in time such that the time derivative 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 [45]. RES-CLFs provide guarantees of *rapid exponential stability* for the transverse variables  $\eta$ . In particular, a function  $V_\varepsilon(\eta)$  is a RES-CLF for the system (1) if there exist positive constants  $c_1, c_2, c_3 > 0$  such that for all  $0 < \varepsilon < 1$  and all states  $(\eta, z)$  it holds that

$$c_1 \|\eta\|^2 \leq V_\varepsilon(\eta) \leq \frac{c_2}{\varepsilon^2} \|\eta\|^2, \quad (13)$$

$$\dot{V}_\varepsilon(\eta, \mu) + \frac{c_3}{\varepsilon} V_\varepsilon(\eta) \leq 0. \quad (14)$$

The RES-CLF will take the form:

$$V_\varepsilon(\eta) = \eta^T \begin{bmatrix} \frac{1}{\varepsilon} I & 0 \\ 0 & I \end{bmatrix} P \begin{bmatrix} \frac{1}{\varepsilon} I & 0 \\ 0 & I \end{bmatrix} \eta =: \eta^T P_\varepsilon \eta, \quad (15)$$

where  $P$  satisfies the Lyapunov equation (12) and the time derivative of the RES-CLF (15) is computed as

$$\dot{V}_\varepsilon(\eta, \mu) = \frac{\partial V_\varepsilon}{\partial \eta} \dot{\eta} = L_{\bar{f}} V_\varepsilon(\eta) + L_{\bar{g}} V_\varepsilon(\eta)\mu, \quad (16)$$

where

$$\begin{aligned} L_{\bar{f}} V_\varepsilon(\eta) &= \frac{\partial V_\varepsilon}{\partial \eta} \bar{f}(\eta) = \eta^T (F^T P_\varepsilon + P_\varepsilon F) \eta, \\ L_{\bar{g}} V_\varepsilon(\eta) &= \frac{\partial V_\varepsilon}{\partial \eta} \bar{g}(\eta) = 2\eta^T P_\varepsilon G. \end{aligned} \quad (17)$$

It can be show that for any Lipschitz continuous feedback control law  $\mu$  that satisfies (14), it holds that

$$V(\eta) \leq e^{-\frac{c_3}{\varepsilon} t} V(\eta(0)), \quad \|\eta(t)\| \leq \frac{1}{\varepsilon} \sqrt{\frac{c_2}{c_1}} e^{-\frac{c_3}{2\varepsilon} t} \|\eta(0)\|, \quad (18)$$

i.e. the rate of exponential convergence can be directly controlled with the constant  $\varepsilon$  through  $\frac{c_3}{\varepsilon}$ . One such controller is

the CLF-based quadratic program (CLF-QP)-based controller, introduced in [3], where  $\mu$  is directly selected through an online quadratic program to satisfy (14):

---

**CLF-QP:**

$$\begin{aligned} u^*(x) = \underset{u, \mu}{\operatorname{argmin}} \quad & \mu^T \mu & (19) \\ \text{s.t.} \quad & \dot{V}_\varepsilon(\eta, \mu) + \frac{c_3}{\varepsilon} V_\varepsilon(\eta) \leq 0, & \text{(CLF)} \\ & u = u_{ff}(x) + (L_g L_f y(x))^{-1} \mu. & \text{(IO)} \end{aligned}$$


---

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

$$A_{clf} \mu \leq b_{clf}, \quad (20)$$

where

$$A_{clf} = L_{\bar{g}} V_\varepsilon(\eta); \quad b_{clf} = -L_f V_\varepsilon(\eta) - \frac{c_3}{\varepsilon} V_\varepsilon(\eta). \quad (21)$$

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

$$A_{IO} \begin{bmatrix} u \\ \mu \end{bmatrix} = b_{IO}, \quad (22)$$

where

$$A_{IO} = [I \quad -(L_g L_f y(x))^{-1}], \quad b_{IO} = u_{ff}(x). \quad (23)$$

This optimization is solved pointwise in time using efficient quadratic program solvers, such as CVXGEN [46], at real-time speeds over 1 kHz.

**Remark 1:** Note that the solution of the above QP has been shown to be Lipschitz continuous in [47].

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

$$A_c(x)u \leq b_c(x). \quad (24)$$

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

---

**CLF-QP with Constraints:**

$$\begin{aligned} u^*(x) = \underset{u, \mu, \delta}{\operatorname{argmin}} \quad & \mu^T \mu + p\delta^2 & (25) \\ \text{s.t.} \quad & \dot{V}_\varepsilon(\eta, \mu) + \frac{c_3}{\varepsilon} V_\varepsilon(\eta) \leq \delta, & \text{(CLF)} \\ & A_c(x)u \leq b_c(x), & \text{(Constraints)} \\ & u = u_{ff}(x) + (L_g L_f y(x))^{-1} \mu, & \text{(IO)} \end{aligned}$$


---

where  $p$  is a large positive number that represents the penalty of relaxing the CLF inequality, which is necessary to keep the QP feasible when the constraints conflict with each other.

The constraints above could be input saturation constraints expressed as,

$$\begin{aligned} u^*(x) = \underset{u, \mu, \delta}{\operatorname{argmin}} \quad & \mu^T \mu + p\delta^2 & (26) \\ \text{s.t.} \quad & \dot{V}_\varepsilon(\eta, \mu) + \frac{c_3}{\varepsilon} V_\varepsilon(\eta) \leq \delta, & \text{(CLF)} \\ & u_{min} \leq u \leq u_{max}, & \text{(Input Saturation)} \\ & u = u_{ff}(x) + (L_g L_f y(x))^{-1} \mu. & \text{(IO)} \end{aligned}$$

Note that, similar to (20), the CLF inequality condition in the above CLF-QPs is affine in  $\mu$ , 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 [3], wherein experimental demonstration of bipedal walking with strict input constraints was demonstrated, and in enabling the application of L1 adaptive control for bipedal robots [48].

**Remark 2: (Stability of CLF-QP with relaxed CLF inequality.)** In subsequent sections of this paper, we develop different types of controllers based on the relaxed CLF-QP (25). Allowing the violation of the RES-CLF condition (14) enables us to incorporate various other input and state constraints as we saw in (25). Additionally, the relaxed RES-CLF condition also enables incorporating safety constraints through barriers (see Section II-C), as well as enabling modification of the controller to increase the robustness of the closed-loop system (see Section IV). However, it must be noted that the relaxation of the RES-CLF condition could lead to potential instability. In Section III, we establish sufficient conditions under which the relaxed CLF-QP controller can still retain the exponential stability of the hybrid periodic orbit.

Having revisited control Lyapunov function based quadratic programs, we will next revisit control Barrier functions.

### C. Control Barrier Function

We begin with the control affine system (1) with the goal to design a controller to keep the state  $x$  in the safe set

$$\mathcal{C} = \{x \in \mathbb{R}^n : h(x) \geq 0\}, \quad (27)$$

where  $h : \mathbb{R}^n \rightarrow \mathbb{R}$  is a continuously differentiable function. Then a function  $B : \mathcal{C} \rightarrow \mathbb{R}$  is a Control Barrier Function (CBF) [4] if there exists class  $\mathcal{K}$  function  $\alpha_1$  and  $\alpha_2$  such that, for all  $x \in \operatorname{Int}(\mathcal{C}) = \{x \in \mathbb{R}^n : h(x) > 0\}$ ,

$$\frac{1}{\alpha_1(h(x))} \leq B(x) \leq \frac{1}{\alpha_2(h(x))}, \quad (28)$$

$$\dot{B}(x, u) \leq \frac{\gamma}{B(x)}, \quad (29)$$

where  $\gamma > 0$  and

$$\dot{B}(x, u) = \frac{\partial B}{\partial x} \dot{x} = L_f B(x) + L_g B(x)u, \quad (30)$$

with the Lie derivatives computed as,

$$L_f B(x) = \frac{\partial B}{\partial x} f(x), \quad L_g B(x) = \frac{\partial B}{\partial x} g(x). \quad (31)$$

Thus, if there exists a Barrier function  $B(x)$  that satisfies the CBF condition in (29), 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 [4], this notion of a CBF is stricter than standard notions of CBFs in prior literature that only require  $\dot{B} \leq 0$ .

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

$$B(x) = \frac{1}{h(x)}. \quad (32)$$

Incorporating the CBF condition (29) into the CLF-QP, we have the following CBF-CLF-QP based controllers:

---

#### CBF-CLF-QP:

$$u^*(x) = \underset{u, \mu, \delta}{\operatorname{argmin}} \quad \mu^T \mu + p \delta^2 \quad (33)$$

$$\text{s.t.} \quad \dot{V}_\varepsilon(\eta, \mu) + \frac{c_3}{\varepsilon} V_\varepsilon(\eta) \leq \delta, \quad (\text{CLF})$$

$$\dot{B}(x, u) - \frac{\gamma}{B(x)} \leq 0, \quad (\text{CBF})$$

$$A_c(x)u \leq b_c(x), \quad (\text{Constraints})$$

$$u = u_{ff}(x) + (L_g L_f y(x))^{-1} \mu. \quad (\text{IO})$$


---

**Remark 3:** Note that, similar to (20), for any nonlinear affine system and nonlinear state-dependent constraint, the CBF condition (29) is scalar and affine in  $u$ , ensuring a compact quadratic program that can be solved in 1 kHz. More importantly, the satisfaction of this instantaneous condition at each time guarantees the state dependent constraint  $h(x) \geq 0$  for all  $t \geq t_0$ .

Having presented control Lyapunov functions, control Barrier functions, and their incorporation into a quadratic program with constraints, we now prove the stability of the QP controller with relaxed CLF inequality.

### III. SUFFICIENT CONDITIONS FOR THE STABILITY OF CLF WITH RELAXED INEQUALITY

In this Section, 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 recall that as discussed in Section II-B2 and Remark 2, relaxing the CLF inequality constraint is necessary to keep the QP feasible when additional potentially conflicting constraints are added to the QP. However, in this case, we could have a potential loss in stability guarantees if a certain sufficient condition, that is established next, is not satisfied.

We begin with the standard RES-CLF that guarantees the following inequality,

$$\dot{V}_\varepsilon(\eta, \mu) + \frac{c_3}{\varepsilon} V_\varepsilon(\eta) \leq 0. \quad (34)$$

The CLF-QP with the relaxed inequality takes the form,

$$\dot{V}_\varepsilon(\eta, \mu) + \frac{c_3}{\varepsilon} V_\varepsilon(\eta) \leq d_\varepsilon, \quad (35)$$

where  $d_\varepsilon(t) \geq 0$  represents the time-varying relaxation of the RES-CLF condition. We define

$$w_\varepsilon(t) = \int_0^t \frac{d_\varepsilon(\tau)}{V_\varepsilon} d\tau. \quad (36)$$

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 Section III-A we will look at continuous-time systems where we will need exponential stability ( $\varepsilon = 1$  in the above formulations), while in Section III-B we will look at hybrid systems where we need rapid exponential stability ( $\varepsilon < 1$ ).

#### A. Stability of the Relaxed CLF-QP Controller for Continuous-Time Systems

Consider a control affine system

$$\dot{\eta} = \bar{f}(\eta) + \bar{g}(\eta)\mu, \quad (37)$$

$$\dot{z} = p(\eta, z),$$

where  $\eta$  is the controlled (or output) state, and  $z$  is the uncontrolled state. Let  $\mathcal{O}_Z$  be an exponentially stable periodic orbit for the zero dynamics  $\dot{z} = p(0, z)$ . Let  $\mathcal{O} = \iota_0(\mathcal{O}_Z)$  be a periodic orbit for the full-order dynamics corresponding to the periodic orbit of the zero dynamics,  $\mathcal{O}_Z$ , through the canonical embedding  $\iota_0 : Z \rightarrow X \times Z$  given by  $\iota_0(z) = (0, z)$ . As stated in [45, Theorem 1], for all control inputs  $\mu(\eta, z)$  that guarantee enforcing the ES-CLF condition (with  $\varepsilon = 1$ ) (34), we have that  $\mathcal{O} = \iota_0(\mathcal{O}_Z)$  is an exponentially stable periodic orbit of (37). Then, for the relaxed CLF-QP condition (35) ( $d_\varepsilon \neq 0$ ), the following theorem establishes sufficient conditions under which the exponential stability of the periodic orbit still holds.

**Theorem 1:** Consider a nonlinear control affine system (37). Let  $\mathcal{O}_Z$  be an exponentially stable periodic orbit for the zero dynamics  $\dot{z} = p(0, z)$ . Then  $\mathcal{O} = \iota_0(\mathcal{O}_Z)$  is an exponentially stable periodic orbit of (37), if

$$\bar{w}_\varepsilon := \sup_{t \geq 0} w_\varepsilon(t) \quad (38)$$

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 [45]. 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

$$w_\varepsilon(t) \leq \bar{w}_\varepsilon, \quad \forall t \geq 0. \quad (39)$$

Next, from (35), we have

$$\begin{aligned} \frac{dV_\varepsilon}{dt} &\leq -\frac{c_3}{\varepsilon} V_\varepsilon + d_\varepsilon(t), \\ \Rightarrow \frac{dV_\varepsilon}{V_\varepsilon} &\leq -\frac{c_3}{\varepsilon} dt + \frac{d_\varepsilon(t)}{V_\varepsilon} dt, \\ \Rightarrow \ln \left( \frac{V_\varepsilon(t)}{V_\varepsilon(0)} \right) &\leq -\frac{c_3}{\varepsilon} t + \int_0^t \frac{d_\varepsilon(\tau)}{V_\varepsilon} d\tau, \\ \Rightarrow V_\varepsilon(t) &\leq e^{-\frac{c_3}{\varepsilon} t + w_\varepsilon(t)} V_\varepsilon(0). \end{aligned} \quad (40)$$

This, in combination with the inequality in (13), then implies that

$$\|\eta(t)\| \leq \sqrt{\frac{c_2}{c_1}} \frac{1}{\varepsilon} e^{-\frac{c_3}{2\varepsilon}t + \frac{1}{2}w_\varepsilon(t)} \|\eta(0)\|. \quad (41)$$

Moreover, from the inequality in (39), we have

$$\begin{aligned} \|\eta(t)\| &\leq \sqrt{\frac{c_2}{c_1}} \frac{1}{\varepsilon} e^{-\frac{c_3}{2\varepsilon}t + \frac{1}{2}\bar{w}_\varepsilon} \|\eta(0)\|, \\ &= \left( \sqrt{\frac{c_2}{c_1}} \frac{1}{\varepsilon} e^{\frac{1}{2}\bar{w}_\varepsilon} \right) e^{-\frac{c_3}{2\varepsilon}t} \|\eta(0)\|. \end{aligned} \quad (42)$$

Therefore, if  $\bar{w}_\varepsilon$  is finite, the control output  $\eta = 0$  will still be exponentially stable under the relaxed CLF condition (35). Therefore, from [45, Theorem 1],  $\mathcal{O} = \iota_0(\mathcal{O}_Z)$  is an exponentially stable periodic orbit of (37).

**Remark 4:** It must be noted that the CLF  $V_\varepsilon$  no longer provides a guarantee of exponential stability due to the relaxation in (35). However due to the inequality established in (42), and by the converse Lyapunov function theorems [49], 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. 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 (8) without the restriction on the vector fields as in (9). Similar to the stability analysis for continuous-time systems in the previous section, we also use here the notions of  $d_\varepsilon(t)$  (35), the relaxation of the CLF condition, and  $w_\varepsilon(t)$  (36), 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  $(\eta, z)$  to the switching surface  $S$ . Then, intuitively,  $w_\varepsilon(T_I^\varepsilon(\eta, z))$  indicates a scaled version of the total violation of the RES-CLF bound in (34) 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 (8) restricted to the surface  $Z$  in (6), i.e.,

$$\mathcal{H}_Z : \begin{cases} \dot{z} = p(0, z), & z \notin S \cap Z, \\ z^+ = \Delta_z(0, z^-), & z^- \in S \cap Z. \end{cases} \quad (43)$$

We then have the following theorem:

**Theorem 2:** Let  $\mathcal{O}_Z$  be an exponentially stable hybrid periodic orbit of the hybrid zero dynamics  $\mathcal{H}|_Z$  (43) transverse to  $S \cap Z$  and the continuous dynamics of  $\mathcal{H}$  (8) controlled by

Notations	Model types
$f, g$	true ( <i>unknown</i> ) nonlinear model
$\tilde{f}, \tilde{g}$	nominal nonlinear model
$f_\varepsilon, g_\varepsilon$	true ( <i>unknown</i> ) I-O linearized model
$\tilde{f}_\varepsilon, \tilde{g}_\varepsilon$	nominal I-O linearized model

TABLE I: A list of notations for different models used in this paper. 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 controllers in this paper use the nominal models and offer robustness guarantees to the uncertainty between the two models.

a CLF-QP with relaxed inequality (25). 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 (25) 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  $\mathcal{H}$ .

*Proof:* See Appendix A. ■

Having presented the stability analysis for the QP controller with relaxed CLF inequality, in the next Section, we will explore the effects of model uncertainty and propose robust QP controllers to guarantee stability, safety and constraints under model uncertainty.

## IV. PROPOSED ROBUST CONTROL BASED QUADRATIC PROGRAMS

The controllers presented in Section II 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 [5], quadrotor flight [6], and dynamic walking for bipedal robots [7], [8].

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 (33) requires the accurate knowledge of the nonlinear dynamics  $f(x), g(x)$  to compute the Lie derivatives in (3), (17) and (31). Therefore, model uncertainty that is usually present in physical systems can potentially cause poor quality of control leading to tracking errors, and lead to instability [43], as well as violation of the safety-critical constraints [9]. 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 and then proposed robust controllers based quadratic programs to address those effects. Firstly, we will analyze the effects of model uncertainty on the CLF condition.

### A. Adverse Effects of Uncertainty on the CLF-QP controller

In order to analyze the effects of model uncertainty in our controllers, we assume that the vector fields,  $f(x), g(x)$  of the

real dynamics (1), are unknown. We therefore have to design our controller based on the nominal vector fields  $\tilde{f}(x), \tilde{g}(x)$ . Then, the pre-control law (4) get's reformulated as

$$u(x) = \tilde{u}_{ff}(x) + (L_{\tilde{g}}L_{\tilde{f}}y(x))^{-1}\mu, \quad (44)$$

with

$$\tilde{u}_{ff}(x) := -(L_{\tilde{g}}L_{\tilde{f}}y(x))^{-1}L_{\tilde{f}}^2y(x), \quad (45)$$

where we have used the nominal model rather than the unknown real dynamics.

Substituting  $u(x)$  from (44) into (2), the input-output linearized system then becomes

$$\ddot{y} = \mu + \Delta_1 + \Delta_2\mu, \quad (46)$$

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

$$\begin{aligned} \Delta_1 &:= L_{\tilde{f}}^2y(x) - L_gL_fy(x)(L_{\tilde{g}}L_{\tilde{f}}y(x))^{-1}L_{\tilde{f}}^2y(x), \\ \Delta_2 &:= L_gL_fy(x)(L_{\tilde{g}}L_{\tilde{f}}y(x))^{-1} - I. \end{aligned} \quad (47)$$

**Remark 5:** In the definitions of  $\Delta_1, \Delta_2$ , 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 (10), the closed-loop system now takes the form

$$\dot{\eta} = \tilde{f}(\eta) + \tilde{g}(\eta)\mu, \quad (48)$$

where

$$\tilde{f}(\eta) = F\eta + \begin{bmatrix} O \\ \Delta_1 \end{bmatrix}, \tilde{g}(\eta) = G + \begin{bmatrix} O \\ \Delta_2 \end{bmatrix}. \quad (49)$$

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 (see [50]). 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 (16) becomes more complex and dependent on  $\Delta_1, \Delta_2$ .

Again, because our controller can only use the nominal model  $\tilde{f}, \tilde{g}$  in (48) and not the true model  $f, g$  as defined in (9) (see Table I for different types of models), the Lie derivatives of  $V_\varepsilon$  with respect to the nominal model  $\tilde{f}, \tilde{g}$  will be as follows:

$$L_{\tilde{f}}V_\varepsilon = \eta^T(F^T P_\varepsilon + P_\varepsilon F)\eta, \quad (50)$$

$$L_{\tilde{g}}V_\varepsilon = 2\eta^T P_\varepsilon G. \quad (51)$$

Then, the true time-derivative of the CLF defined in (15) is:

$$\begin{aligned} \dot{V}_\varepsilon &= L_{\tilde{f}}V_\varepsilon + L_{\tilde{g}}V_\varepsilon\mu \\ &= L_{\tilde{f}}V_\varepsilon + 2\eta^T P_\varepsilon \underbrace{\begin{bmatrix} \mathbf{0} \\ \Delta_1 \end{bmatrix}}_{\Delta_1^v} + \underbrace{L_{\tilde{g}}V_\varepsilon\mu}_{\mu_v} + \underbrace{L_{\tilde{g}}V_\varepsilon\Delta_2\mu}_{\Delta_2^v\mu_v} \\ &= L_{\tilde{f}}V_\varepsilon + \Delta_1^v + (1 + \Delta_2^v)\mu_v, \end{aligned} \quad (52)$$

where we have defined the following new scalar variables: uncertainty  $\Delta_1^v \in \mathbb{R}$ , virtual control input  $\mu_v \in \mathbb{R}$ , and uncertainty  $\Delta_2^v \in \mathbb{R}$ .

## B. Robust CLF-QP

Having discussed the effect of model uncertainty on the control Lyapunov function based controllers, 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.

With the presence of model uncertainty, the CLF condition (14) now becomes:

$$\dot{V}_\varepsilon(\eta, \Delta_1^v, \Delta_2^v, \mu_v) + \frac{c_3}{\varepsilon}V_\varepsilon \leq 0. \quad (53)$$

In general, we can not satisfy this inequality for all possible unknown  $\Delta_1^v, \Delta_2^v$ . To address this, we assume the uncertainty is bounded as follows

$$|\Delta_1^v| \leq \Delta_{1,max}^v, \quad |\Delta_2^v| \leq \Delta_{2,max}^v. \quad (54)$$

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

$$\begin{aligned} &\dot{V}_\varepsilon(\eta, \Delta_1^v, \Delta_2^v, \mu_v) + \frac{c_3}{\varepsilon}V_\varepsilon \leq 0, \\ &\quad \forall |\Delta_1^v| \leq \Delta_{1,max}^v, \quad \forall |\Delta_2^v| \leq \Delta_{2,max}^v \\ \Leftrightarrow &\max_{\substack{|\Delta_1^v| \leq \Delta_{1,max}^v \\ |\Delta_2^v| \leq \Delta_{2,max}^v}} \dot{V}_\varepsilon(\eta, \Delta_1^v, \Delta_2^v, \mu_v) + \frac{c_3}{\varepsilon}V_\varepsilon \leq 0. \end{aligned} \quad (55)$$

It means that choosing  $\mu$  that satisfies (55) implies (18) for every  $\Delta_1^v, \Delta_2^v$  satisfying  $|\Delta_1^v| \leq \Delta_{1,max}^v, |\Delta_2^v| \leq \Delta_{2,max}^v$ . Then, our optimal robust control can be expressed as the following min-max problem:

### Robust CLF:

$$\begin{aligned} u^*(x) &= \underset{u, \mu, \mu_v}{\operatorname{argmin}} \quad \mu^T \mu \quad (56) \\ \text{s.t.} &\quad \max_{\substack{|\Delta_1^v| \leq \Delta_{1,max}^v \\ |\Delta_2^v| \leq \Delta_{2,max}^v}} \dot{V}_\varepsilon(\eta, \Delta_1^v, \Delta_2^v, \mu_v) + \frac{c_3}{\varepsilon}V_\varepsilon \leq 0 \\ &\quad \mu_v = L_{\tilde{g}}V_\varepsilon\mu \quad \text{(Robust CLF)} \\ &\quad u = \tilde{u}_{ff}(x) + (L_{\tilde{g}}L_{\tilde{f}}y(x))^{-1}\mu. \quad \text{(IO)} \end{aligned}$$

Note that, one advantage of CLF based control is to convert the problem of driving a vector output  $\eta \rightarrow 0$  to the problem of driving a scalar  $V_\varepsilon(\eta) \rightarrow 0$ . Therefore, since  $V_\varepsilon$  is a scalar, we always have  $\mu_v, \Delta_1^v, \Delta_2^v$  in (52) being scalars. Then the min-max optimization problem (56) can be expressed as a quadratic program as follows:

$$\begin{aligned} &\max_{\substack{|\Delta_1^v| \leq \Delta_{1,max}^v \\ |\Delta_2^v| \leq \Delta_{2,max}^v}} \dot{V}_\varepsilon(\eta, \Delta_1^v, \Delta_2^v, \mu_v) + \frac{c_3}{\varepsilon}V_\varepsilon \leq 0 \\ \Leftrightarrow &\max_{\substack{|\Delta_1^v| \leq \Delta_{1,max}^v \\ |\Delta_2^v| \leq \Delta_{2,max}^v}} L_{\tilde{f}}V_\varepsilon + \Delta_1^v + (1 + \Delta_2^v)\mu_v + \frac{c_3}{\varepsilon}V_\varepsilon \leq 0, \end{aligned} \quad (57)$$

and the above robust CLF condition holds if

$$\begin{cases} \Psi_0^{max} + \Psi_1^p \mu_v \leq 0 \\ \Psi_0^{max} + \Psi_1^n \mu_v \leq 0 \end{cases}, \quad (58)$$

where

$$\begin{aligned}\Psi_0^{max} &= L_{\tilde{f}}V_\varepsilon + \frac{c_3}{\varepsilon}V_\varepsilon + \Delta_{1,max}^v \\ \Psi_1^p &= 1 + \Delta_{2,max}^v, \\ \Psi_1^n &= 1 - \Delta_{2,max}^v.\end{aligned}\quad (59)$$

Thus the robust optimal CLF in (56) can be satisfied using the following quadratic program based controller:

---

**Robust CLF-QP:**

$$\begin{aligned}u^* &= \underset{u, \mu, \mu_v}{\operatorname{argmin}} \quad \mu^T \mu & (60) \\ \text{s.t.} \quad & \Psi_0^{max}(\eta, \Delta_{1,max}^v) + \Psi_1^p(\Delta_{2,max}^v) \mu_v \leq 0, \\ & \Psi_0^{max}(\eta, \Delta_{1,max}^v) + \Psi_1^n(\Delta_{2,max}^v) \mu_v \leq 0, \\ & \mu_v = L_{\tilde{g}}V_\varepsilon \mu, & \text{(Robust CLF)} \\ & u = \tilde{u}_{ff}(x) + (L_{\tilde{g}}L_{\tilde{f}}y(x))^{-1} \mu. & \text{(IO)}\end{aligned}$$


---

**Remark 6:** Note that since  $V(\eta)$  is a local CLF for the nonlinear system (1) and the fact that  $\dot{V}$  depends continuously on  $\Delta_1^v, \Delta_2^v$ , there always exists  $\mu$  satisfying (55) for a local region around  $\eta = 0, \Delta_1^v = 0, \Delta_2^v = 0$ . Moreover, if  $\Delta_{1,max}^v, \Delta_{2,max}^v$  are chosen to be within this region, then the above QP is guaranteed to be feasible. For QPs with additional constraints we assume point-wise feasibility.

**Remark 7:** Note that  $\Delta_{1,max}^v, \Delta_{2,max}^v$  are design parameters for control design, and can be easily changed online. While we do not present a principled way of choosing these parameters, it must be noted that  $\Delta_{1,max}^v, \Delta_{2,max}^v$  can be chosen based on knowledge of the expected uncertainty the controller is to be designed to handle.

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 CLF condition (see [51]). Similarly, here we relax the robust CLF condition to obtain,

---

**Robust CLF-QP with Constraints:**

$$\begin{aligned}u^*(x) &= \underset{u, \mu, \mu_v, \delta_1, \delta_2}{\operatorname{argmin}} \quad \mu^T \mu + p_1 \delta_1^2 + p_2 \delta_2^2 & (61) \\ \text{s.t.} \quad & \Psi_0^{max}(\eta, \Delta_{1,max}^v) + \Psi_1^p(\Delta_{2,max}^v) \mu_v \leq \delta_1, \\ & \Psi_0^{max}(\eta, \Delta_{1,max}^v) + \Psi_1^n(\Delta_{2,max}^v) \mu_v \leq \delta_2, \\ & \mu_v = L_{\tilde{g}}V_\varepsilon \mu, & \text{(Robust CLF)} \\ & A_c(x)u \leq b_c(x), & \text{(Constraints)} \\ & u = u_{ff}(x) + (L_g L_f y(x))^{-1} \mu. & \text{(IO)}\end{aligned}$$


---

**Remark 8:** It's critical to note that the additional constraints in (61) 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 (26), 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 that are invariant to the model uncertainty are not common, and we will have to explicitly address and capture the effect of uncertainty on constraints. We will address this in Section IV-D.

**C. Robust CBF**

Following the same approach we use for the robust CLF, we now analyze the effect of model uncertainty on the CBF condition and then present our robust CBF controller.

Similar to what we have seen about the effect of uncertainty on CLFs, we will now see the effect of uncertainty on CBFs. We note that the time-derivative of the Barrier function in (32) depends on the real model. Therefore we need to enforce the following constraint given by (29):

$$\dot{B}(x, f, g, u) = L_f B(x) + L_g B(x)u \leq \frac{\gamma}{B(x)} \quad (62)$$

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,

$$\dot{B}(x, \tilde{f}, \tilde{g}, u) = L_{\tilde{f}} B(x) + L_{\tilde{g}} B(x)u \leq \frac{\gamma}{B(x)} \quad (63)$$

where  $\dot{x} = \tilde{f}(x) + \tilde{g}(x)u$  is the nominal system dynamics known by the controller. Clearly due to model uncertainty, or the difference between  $(f(x), g(x))$  and  $(\tilde{f}(x), \tilde{g}(x))$ , the constraint in (63) is different from the one in (62). In fact, as analyzed in [9], this results in violation of the safety-critical constraint established by the Barrier function.

We then define a virtual control input  $\mu_b \in \mathbb{R}$  such that the time derivative of the CBF is

$$\dot{B}(x, \tilde{f}, \tilde{g}, u) = L_{\tilde{f}} B(x) + L_{\tilde{g}} B(x)u =: \mu_b. \quad (64)$$

We call this virtual input-output linearization (VIOL). The CBF now takes the form of a linear system,  $\dot{B} = \mu_b$ , and therefore the effect of uncertainty can be easily addressed by using the same approach as with the robust CLF-QP. In particular, similar to what we have done with CLF in Section IV-A, the effect of model uncertainty on CBF transforms (62) into the following form:

$$\begin{aligned}\dot{B}(x, \Delta_1^b, \Delta_2^b, \mu_b) &= L_f B(x) + L_g B(x)u \\ &= \mu_b + \Delta_1^b + \Delta_2^b \mu_b,\end{aligned}\quad (65)$$

with  $\Delta_1^b, \Delta_2^b \in \mathbb{R}$  and  $\mu_b$  is defined based on the nominal model  $\tilde{f}, \tilde{g}$ :

$$\mu_b = L_{\tilde{f}} B(x) + L_{\tilde{g}} B(x)u. \quad (66)$$

The CBF condition (62) under model uncertainty becomes

$$\dot{B}(x, \Delta_1^b, \Delta_2^b, \mu_b) - \frac{\gamma}{B(x)} \leq 0. \quad (67)$$



Because we developed the CLF and CBF having the similar form of I-O 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.,

$$|\Delta_1^b| \leq \Delta_{1,max}^b, \quad |\Delta_2^b| \leq \Delta_{2,max}^b. \quad (68)$$

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

$$\max_{\substack{|\Delta_1^b| \leq \Delta_{1,max}^b \\ |\Delta_2^b| \leq \Delta_{2,max}^b}} \dot{B}(x, \Delta_1^b, \Delta_2^b, \mu_b) - \frac{\gamma}{B(x)} \leq 0. \quad (69)$$

Note that choosing  $\mu_b$  that satisfies (69) implies  $x \in \mathcal{C}$  or  $B(x) \geq 0$  for every  $\Delta_1^b, \Delta_2^b$  satisfying  $|\Delta_1^b| \leq \Delta_{1,max}^b, |\Delta_2^b| \leq \Delta_{2,max}^b$ . Here,  $\mathcal{C}$  is as defined in (27) and  $B(x)$  is as defined in (32). Note that the robust CBF condition in (69) is affine in  $\mu_b$  and can be expressed as

$$\max_{\substack{|\Delta_1^b| \leq \Delta_{1,max}^b \\ |\Delta_2^b| \leq \Delta_{2,max}^b}} \Psi_0^b + \Psi_1^b \mu_b \leq 0 \quad (70)$$

where “ $b$ ” refers to the CBF constraint, and

$$\begin{aligned} \Psi_0^b(x, \Delta_1^b) &:= \Delta_1^b - \frac{\gamma}{B(x)}, \\ \Psi_1^b(x, \Delta_2^b) &:= 1 + \Delta_2^b, \end{aligned} \quad (71)$$

where the above arises due to the time-derivative of the CBF from (65). Since  $\Psi_0^b, \Psi_1^b$  are affine with respect to  $\Delta_1^b, \Delta_2^b$ , and with the assumptions on the bounds on the uncertainty in (68), the robust CBF condition (70) will hold if the following two inequalities hold

$$\begin{aligned} \Psi_{0,max}^b(x) + \Psi_{1,p}^b(x) \mu_b &\leq 0, \\ \Psi_{0,max}^b(x) + \Psi_{1,n}^b(x) \mu_b &\leq 0. \end{aligned} \quad (72)$$

where

$$\Psi_{0,max}^b := \Delta_{1,max}^b - \frac{\gamma}{B(x)}, \quad (73)$$

$$\Psi_{1,p}^b := 1 + \Delta_{2,max}^b, \quad (74)$$

$$\Psi_{1,n}^b := 1 - \Delta_{2,max}^b, \quad (75)$$

We then can incorporate the above robust CBF conditions into the robust CLF-QP (61) resulting in a quadratic program.

**Remark 9:** While we use reciprocal control barrier function (32) in this paper, our proposed approach on robust CBF does not depend on the choice of barrier function. To be more specific, the virtual control input  $\mu_b$  (64) and the effect of model uncertainty on CBF (65) are defined for a general nonlinear function  $B(x)$ .

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

#### D. Robust Constraints

As developed in Section II, the input and state constraints can be expressed as (24). 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,

$$A_c(x, f, g)u \leq b_c(x, f, g). \quad (76)$$

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

$$A_c(x, \tilde{f}, \tilde{g})u \leq b_c(x, \tilde{f}, \tilde{g}). \quad (77)$$

Clearly, the constraint in (77) is different from the desired constraint we want to enforce on the real model (76). 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 10:** As noted earlier, 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 (76) results in  $A_c = I, b_c = u_{max}$ , which clearly is not dependent on the model.

To robustify the “constraints”, we can use a similar method as we did for the control barrier functions. We start by reformulating the constraints (76) by using Virtual Input-Output Linearization (VIOL) for to obtain,

$$A_{c,i}(x, f, g)u - b_{c,i}(x, f, g) =: \mu_{c,i}, \quad (78)$$

and then enforce

$$\mu_{c,i} \leq 0, \quad (79)$$

where the index  $i$  indicates the  $i^{th}$  constraints in (76).

With model uncertainty, we now have,

$$\begin{aligned} A_{c,i}(x, f, g)u - b_{c,i}(x, f, g) &= \mu_{c,i} + \Delta_1^{c,i} + \Delta_2^{c,i} \mu_{c,i} \\ &=: \bar{\mu}_{c,i}(\mu_{c,i}, \Delta_1^{c,i}, \Delta_2^{c,i}), \end{aligned} \quad (80)$$

with  $\Delta_1^{c,i}, \Delta_2^{c,i} \in \mathbb{R}$  and the virtual input  $\mu_{c,i}$  is now defined based on the nominal model  $\tilde{f}, \tilde{g}$ :

$$\mu_{c,i} = A_{c,i}(x, \tilde{f}, \tilde{g})u - b_{c,i}(x, \tilde{f}, \tilde{g}). \quad (81)$$

The robust constraint now becomes

$$\bar{\mu}_{c,i}(\mu_{c,i}, \Delta_1^{c,i}, \Delta_2^{c,i}) \leq 0. \quad (82)$$

Once again, we make the assumption on bounded uncertainty,

$$|\Delta_1^{c,i}| \leq \Delta_{1,max}^{c,i}, \quad |\Delta_2^{c,i}| \leq \Delta_{2,max}^{c,i}, \quad (83)$$

such that the robust constraint condition becomes,

$$\max_{\substack{|\Delta_1^{c,i}| \leq \Delta_{1,max}^{c,i} \\ |\Delta_2^{c,i}| \leq \Delta_{2,max}^{c,i}}} \bar{\mu}_{c,i}(\mu_{c,i}, \Delta_1^{c,i}, \Delta_2^{c,i}) \leq 0. \quad (84)$$

Similar to robust CLF and robust CBF, the above robust constraints condition is affine in  $\mu_{c,i}$  and can be expressed as

$$\max_{\substack{|\Delta_1^{c,i}| \leq \Delta_{1,max}^{c,i} \\ |\Delta_2^{c,i}| \leq \Delta_{2,max}^{c,i}}} \Psi_0^{c,i} + \Psi_1^{c,i} \mu_{c,i} \leq 0 \quad (85)$$

where “ $(c, i)$ ” refers to the  $i^{\text{th}}$  constraint, and

$$\begin{aligned}\Psi_0^{c,i}(x, \Delta_1^{c,i}) &:= \Delta_1^{c,i}, \\ \Psi_1^{c,i}(x, \Delta_2^{c,i}) &:= 1 + \Delta_2^{c,i},\end{aligned}\quad (86)$$

where the above arises due to the form of the constraints in (80). The same procedure used for the robust CBF in the previous section can be applied for the robust constraints as well to show how the max inequality gets converted to a set of linear inequalities that are enforced by a quadratic program.

### E. Robust CBF-CLF-QP with Robust Constraints

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

---

#### Robust CBF-CLF-QP with Robust Constraints:

$$u^*(x) = \quad (87)$$

$$\underset{u, \mu, \mu_v, \mu_b, \mu_c, \delta}{\operatorname{argmin}} \quad \mu^T \mu + p\delta^2$$

$$\text{s.t.} \quad \max_{\substack{|\Delta_1^v| \leq \Delta_{1,max}^v \\ |\Delta_2^v| \leq \Delta_{2,max}^v}} \dot{V}_\varepsilon(\eta, \Delta_1^v, \Delta_2^v, \mu_v) + \frac{c_3}{\varepsilon} V_\varepsilon \leq \delta,$$

$$L_{\bar{g}} V_\varepsilon \mu = \mu_v, \quad \text{(Robust CLF)}$$

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

$$L_{\bar{f}} B(x) + L_{\bar{g}} B(x) u = \mu_b, \quad \text{(Robust CBF)}$$

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

$$A_c(x) \mu - b_c(x) = \mu_c, \quad \text{(Robust Constraints)}$$

$$u = \tilde{u}_{ff}(x) + (L_{\bar{g}} L_{\bar{f}} y(x))^{-1} \mu. \quad \text{(IO)}$$


---

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 bipedal robot and experiments on a spring cart.

## V. SIMULATION AND EXPERIMENTAL VALIDATION

### A. 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, a planar five-link bipedal robot. Further description of RABBIT and the associated mathematical model can be found in [52, 53].

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.

In the following simulations, we run an offline optimization process to generate a walking gait for the nominal system.

This results in a set of outputs (virtual constraints) that need to be regulated to zero by the controller. Although the offline optimization is on the nominal system, as we will see, the robust controller is able to guarantee enforcement of the virtual constraints on the true model while subject to input constraints, contact force constraints, safety constraints, and model uncertainty.

1) *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 in initial configuration of the robot, resulting in an initial CLF  $V_0 = \eta_0^T P \eta_0 = 5.8$  (see Fig.(1)). We will compare three different controllers,

$$\begin{cases} \text{I: CLF-QP} \\ \text{II: CLF-QP with Constraints (Contact Force Constraints)} \\ \text{III: Robust CLF-QP with Robust Constraints} \end{cases} \quad (88)$$

We also enforce input saturation constraints for all three controller in (88). 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:

$$\begin{cases} \text{Case 1: } m_{load} = 0 \text{ [kg]} \\ \text{Case 2: } m_{load} = 5 \text{ [kg]} \text{ (16\%)} \\ \text{Case 3: } m_{load} = 15 \text{ [kg]} \text{ (47\%)} \end{cases} \quad (89)$$

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:

$$F_{st}^v(x) \geq \delta_N > 0 \quad (90)$$

$$\left| \frac{F_{st}^h(x)}{F_{st}^v(x)} \right| \leq k_f \quad (91)$$

where  $\delta_N$  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.1, 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

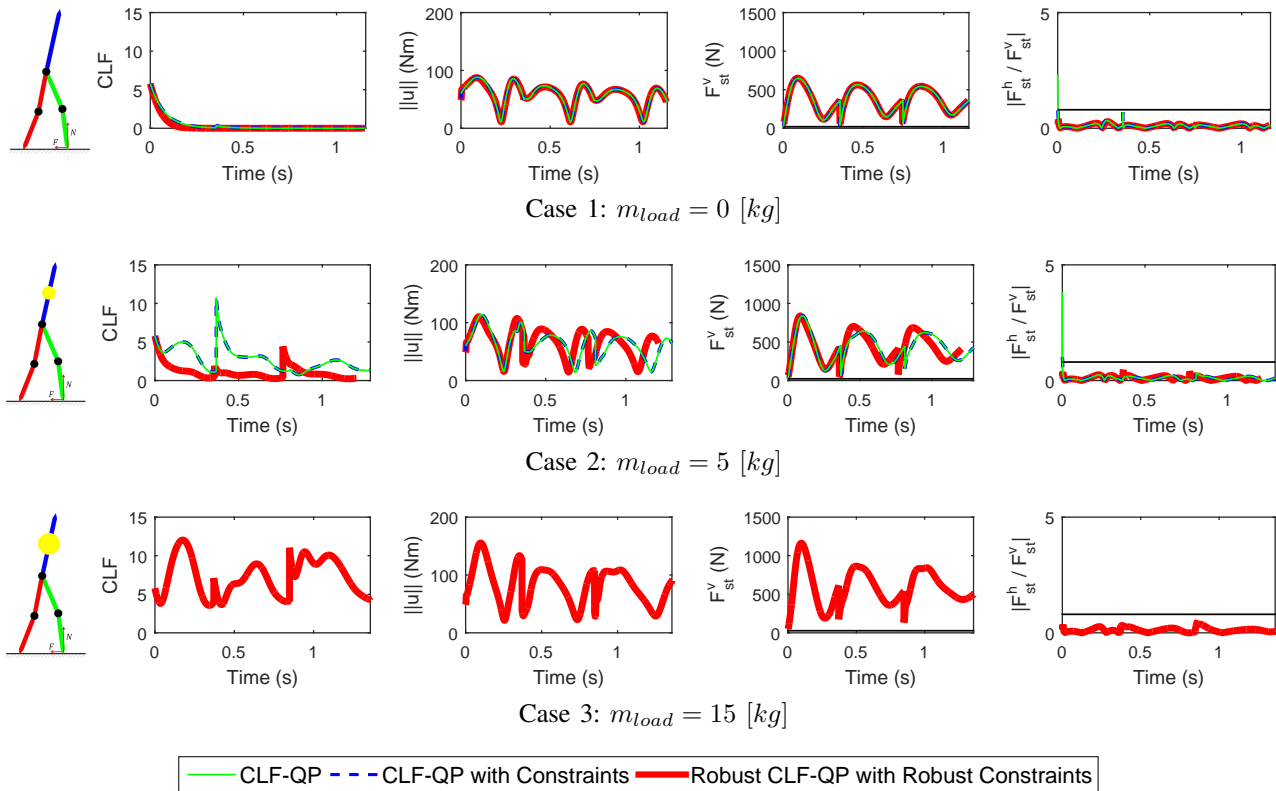


Fig. 1: Three steps of walking of a bipedal robot while carrying an unknown load with contact force constraints. The figure illustrates the tracking performance directly through the CLF (column 1), norm of the control inputs (column 2), non-negative vertical ground reaction force constraint (column 3) and friction cone constraints of being less than 0.8 (column 4). Even in the nominal case of NO uncertainty, the CLF-QP controller fails (the robot slips) due to violation of the contact force constraints (as seen in the rightmost figure of Case 1). The CLF-QP with Constraints (Contact Force Constraints) works well with perfect model but fails with only 5 kg of load (as seen in the violation of the friction constraints in the rightmost figure of Case 2). 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.

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 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.

2) *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 [7]. 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:

- I : CBF-CLF-QP (Foot-Step Placement)
  - II: CBF-CLF-QP with Constraints (Friction Constraints)
  - III: Robust CBF-CLF-QP
  - IV: Robust CBF-CLF-QP with Robust Constraints
- (92)

As you can see from Fig.2, the performance of our proposed robust CBF-CLF-QP dominates that of the CBF-CLF-QP (97% of success in comparison 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.

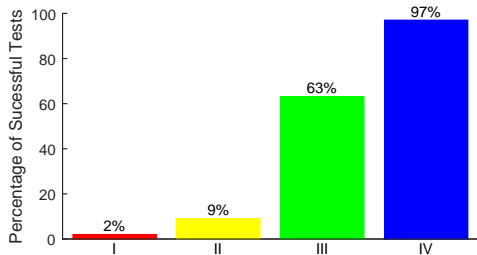


Fig. 2: Dynamic Walking of Bipedal Robot while Carrying Unknown Load, subject to Contact Force Constraints and Foot-Step Location Constraints. 100 random simulations were tested. For each simulation, the unknown load was choose randomly between 5-15 kg, the desired footstep locations for 10 steps were 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 (92).

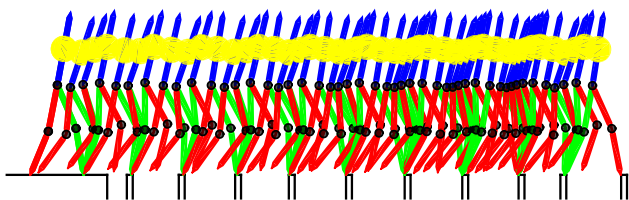


Fig. 3: Dynamic bipdal walking while carrying unknown load, subject to torque saturation constraints (input constraints), contact force constraints (state constraints), and foot-step location constraints (safety constraints). Simulation of the Robust CBF-CLF-QP with Robust Constraints controller for walking over 10 discrete foot holds is shown, subject to model uncertainty of 15 Kg (47 %). A video of the simulation is available at <http://youtu.be/tT0xE1XlyDI>.

Figures 3, 4, 5, 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. Simulations for larger number of steps were also successful as well, but are not presented here due to space constraints.

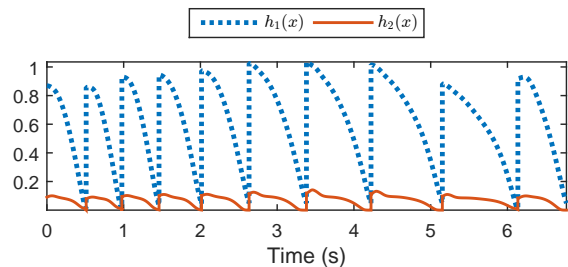
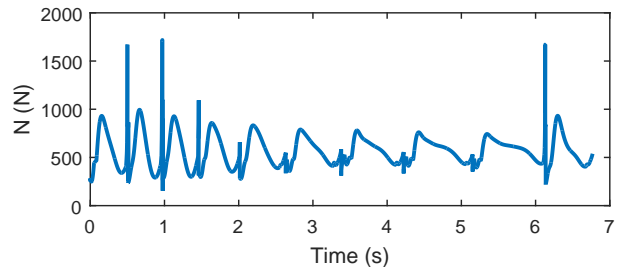
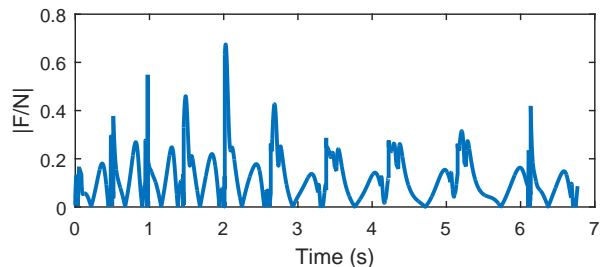


Fig. 4: Dynamic walking of bipedal robot while carrying unknown load of 15 Kg (47 %). The CBF constraints,  $h_1(x) \geq 0$  and  $h_2(x) \geq 0$  defined in [7], guarantee precise foot-step locations. The figure depicts data for 10 steps of walking. As can be clearly seen, the constraints are strictly enforced despite the large model uncertainty.

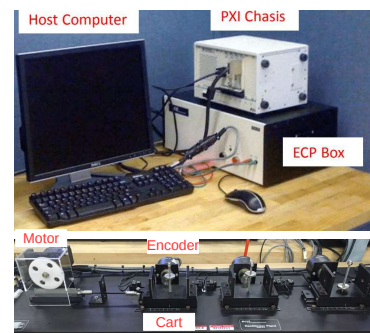


(a) Vertical Contact Force:  $N(x) > \delta_N$ , ( $\delta_N = 0.1mg$ ).

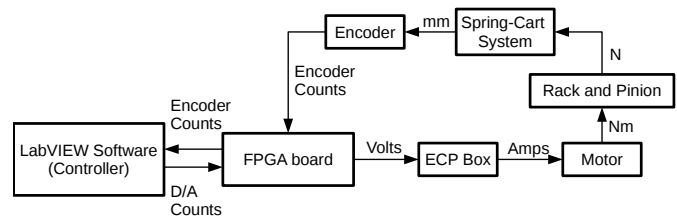


(b) Friction Constraint:  $|F/N| \leq k_f$ , ( $k_f = 0.8$ )

Fig. 5: 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.



(a) Real system



(b) System Diagram

Fig. 6: Schematic of experiment setup for the spring-cart system.

## B. Experimental Results on Spring-Cart System

Having presented numerical results of our proposed control method for bipedal robots, we next present experimental validation for the method on a rectilinear spring-cart system, as shown in Fig. 7. It must be noted that the experiments on a rectilinear spring-cart system are connected to the simulations with a bipedal robot. Since our simulations consider nonlinear

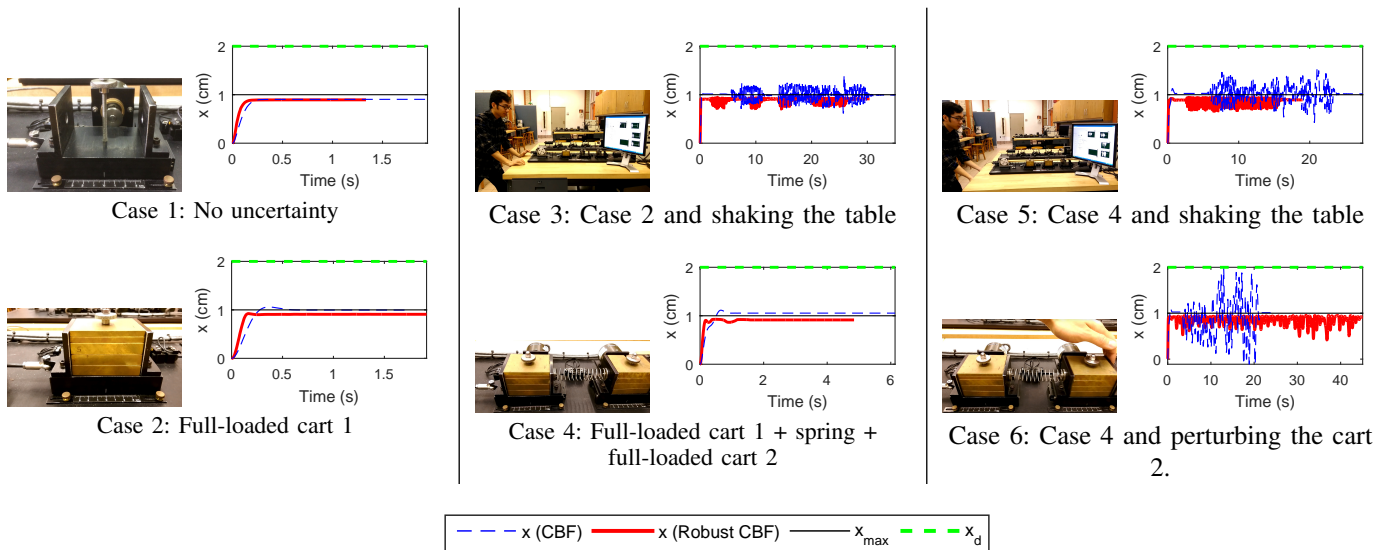


Fig. 7: Experimental results on the spring-mass system. The goal is to drive the cart to the target location of 2 cm, while enforcing the safety constraint that the cart does not cross 1 cm. The controller just uses the nominal model as illustrated in Case 1 for all the 6 different cases. Model uncertainty 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 unknown 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>.

systems with IO linearization controllers and safety-constraints with relative-degree two, our preliminary experiments are thus with linear systems with relative-degree two safety-constraints. Furthermore, with this experimental result, we will perturb the system with different types of model uncertainties (see Fig. 7). It therefore can represent model uncertainty in the IO linearized system discussed in Section IV. Future work will consider experiments with bipedal systems.

For this experiment, our control problem is to track desired set-point ( $x \rightarrow x_d = 2$  [cm]) by using a CLF, and guarantee state-dependent constraint ( $x \leq x_{max} = 1$  [cm]) by using a CBF, where  $x$  is the position of Cart 1.

The experimental setup of the rectilinear spring-cart system (ECP Model 210) is described in Fig. 6. Our controller is implemented using LabVIEW, wherein we call a custom-generated C++ code that implements a fast QP solver. This QP solver code is autogenerated using CVXGEN [46]. The controller runs at 100 Hz on a LabVIEW PXI DAQ and outputs the control input to the ECP system. In particular, the control input is sent to a FPGA board in the PXI DAQ, which then generates and outputs an analog driving voltage (through a 16-bit Digital-to-Analog Converter) to the current amplifier in the ECP system. This amplifier generates the required current to drive the motor which in turn produces a torque. For rectilinear systems this torque is converted to a linear force through a rack and pinion mechanism. The motion of the moving cart is measured by an encoder and this information in encoder counts is acquired by the FPGA board in the PXI DAQ and sent to our controller via the host LabVIEW software.

Fig.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.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.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.

### C. Discussion

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 6, 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.

Therefore, in the future, a more formal design of the bounded uncertainty assumption should be explored.

## VI. CONCLUSION

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 dynamically walking of bipedal robots 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. Future directions involve experimental validations on bipedal robots and other dynamic robotic systems.

### APPENDIX A PROOF OF THEOREM 2

In this Appendix, we will present a detailed proof of Theorem 2 about the stability of CLF based controller with relaxed RES-CLF condition for hybrid systems.

A large part of this proof directly follows from results and proofs in [45], 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  $u_\varepsilon$  of the relaxed CLF-QP controller (25). From [45, (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  $(\eta, z) \in B_\delta(0, 0) \cap S$ ,

$$T^* - \Delta T \leq T_I^\varepsilon(\eta, z) \leq T^* + \Delta T, \quad (93)$$

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 [45], 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 1:** *Let  $\mathcal{O}_Z$  be an exponentially stable periodic orbit of the hybrid zero dynamics  $\mathcal{H}|_Z$  (43) transverse to  $S \cap Z$  and the continuous dynamics of  $\mathcal{H}$  (8) controlled by a CLF-QP with relaxed inequality (25). 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 (25) satisfies  $w_\varepsilon(T_I^\varepsilon(\eta, z)) \leq \bar{w}_\varepsilon$ , then*

$$\|\eta(t)\| \Big|_{t=T_I^\varepsilon(\eta, z)} \leq \sqrt{\frac{c_2}{c_1}} \frac{2e^{-1}}{(T^* - \Delta T)c_3} \|\eta(0)\|. \quad (94)$$

*Proof:* From (41) and because  $w_\varepsilon(T_I^\varepsilon(\eta, z)) \leq \bar{w}_\varepsilon$ , it implies that

$$\begin{aligned} \|\eta(t)\| \Big|_{t=T_I^\varepsilon(\eta, z)} &\leq \sqrt{\frac{c_2}{c_1}} \frac{1}{\varepsilon} e^{-\frac{c_3}{2\varepsilon} T_I^\varepsilon(\eta, z) + \frac{1}{2} w_\varepsilon(T_I^\varepsilon(\eta, z))} \|\eta(0)\| \\ &\leq \sqrt{\frac{c_2}{c_1}} \frac{1}{\varepsilon} e^{-\frac{c_3}{2\varepsilon} T_I^\varepsilon(\eta, z) + \frac{1}{2} \bar{w}_\varepsilon} \|\eta(0)\| \end{aligned} \quad (95)$$

Then, from (93), we have,

$$\|\eta(t)\| \Big|_{t=T_I^\varepsilon(\eta, z)} \leq \sqrt{\frac{c_2}{c_1}} \frac{1}{\varepsilon} e^{-\frac{c_3}{2\varepsilon} (T^* - \Delta T) + \frac{1}{2} \bar{w}_\varepsilon} \|\eta(0)\|. \quad (96)$$

Furthermore, because  $e^{-\alpha} \leq e^{-1}/\alpha, \forall \alpha \geq 0$ , we have:

$$\frac{1}{\varepsilon} e^{-\frac{c_3}{2\varepsilon} (T^* - \Delta T)} \leq \frac{2e^{-1}}{(T^* - \Delta T)c_3}. \quad (97)$$

Then it implies that there exists a  $\bar{c}_3 \leq c_3$  such that:

$$\frac{1}{\varepsilon} e^{-\frac{c_3}{2\varepsilon} (T^* - \Delta T)} \leq \frac{1}{\varepsilon} e^{-\frac{\bar{c}_3}{2\varepsilon} (T^* - \Delta T)} \leq \frac{2e^{-1}}{(T^* - \Delta T)c_3}. \quad (98)$$

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:

$$\frac{1}{\varepsilon} e^{-\frac{\bar{c}_3^a}{2\varepsilon} (T^* - \Delta T)} = \frac{2e^{-1}}{(T^* - \Delta T)c_3}, \quad (99)$$

then any  $\bar{c}_3 \in [\bar{c}_3^a : c_3]$  will satisfy (98).

We can then define

$$\bar{w}_\varepsilon := \frac{c_3 - \bar{c}_3}{\varepsilon} (T^* - \Delta T) \geq 0, \quad (100)$$

$$\Rightarrow -\frac{c_3}{2\varepsilon} (T^* - \Delta T) + \frac{1}{2} \bar{w}_\varepsilon = -\frac{\bar{c}_3}{2\varepsilon} (T^* - \Delta T). \quad (101)$$

Next, plugging (101) into (96), we have,

$$\|\eta(t)\| \Big|_{t=T_I^\varepsilon(\eta, z)} \leq \sqrt{\frac{c_2}{c_1}} \frac{1}{\varepsilon} e^{-\frac{\bar{c}_3}{2\varepsilon} (T^* - \Delta T)} \|\eta(0)\|, \quad (102)$$

$$V_\varepsilon(\eta(t)) \Big|_{t=T_I^\varepsilon(\eta, z)} \leq e^{-\frac{\bar{c}_3}{\varepsilon} (T^* - \Delta T)} V_\varepsilon(\eta(0)). \quad (103)$$

We now complete the proof of Lemma 1 by substituting (98) into (102) to obtain (94). The solution of  $\bar{w}_\varepsilon$  can be found in (100).  $\blacksquare$

Using Lemma 1, we then can follow the same protocol of the proof in [45, Theorem 2] until equation [45, (64)].

We then define  $\beta_1(\varepsilon) = \frac{c_2}{\varepsilon^2} L_{\Delta X}^2 e^{-\frac{\bar{c}_3}{\varepsilon} (T^* - \Delta T)}$  and  $\bar{\beta}_1(\varepsilon) = \frac{c_2}{\varepsilon^2} L_{\Delta X}^2 e^{-\frac{\bar{c}_3}{\varepsilon} (T^* - \Delta T)}$  where  $L_{\Delta X}$ , defined after [45, (59)], is the Lipschitz constant for  $\Delta_X$ . Because  $\beta_1(0^+) := \lim_{\varepsilon \searrow 0} \beta_1(\varepsilon) = 0$ , then there exists an  $\bar{\varepsilon}$  such that

$$\beta_1(\varepsilon) < c_1 \quad \forall \quad 0 < \varepsilon < \bar{\varepsilon}. \quad (104)$$

and for each  $\varepsilon$ , if we define  $\bar{c}_3^b$  be the solution of

$$\frac{c_2}{\varepsilon^2} L_{\Delta X}^2 e^{-\frac{\bar{c}_3^b}{\varepsilon} (T^* - \Delta T)} = c_1, \quad (105)$$

then for  $\bar{c}_3 \in (\bar{c}_3^b : c_3]$ , we have,

$$\beta_1(\varepsilon) \leq \bar{\beta}_1(\varepsilon) < c_1. \quad (106)$$



However, as presented in proof of Lemma 1,  $\bar{c}_3$  also needs to satisfy (98). Therefore, in order to guarantee the satisfaction of both (98) and (106),  $\bar{c}_3$  needs to be chosen in the following set

$$\bar{c}_3 \in \{[\bar{c}_3^a : c_3] \cap (\bar{c}_3^b : c_3)\}, \quad (107)$$

where  $\bar{c}_3^a$  and  $\bar{c}_3^b$  are defined in (99) and (105) respectively.

The rest of the proof follows from the proof of [45, Theorem 2] using  $\bar{\beta}_1$  instead of  $\beta_1$ . We finish our proof with the value of  $\bar{w}_\varepsilon$  determined via (100), in which the feasible set of  $\bar{c}_3$  is defined in (107).

#### REFERENCES

- [1] R. A. Freeman and P. V. Kokotović, *Robust Nonlinear Control Design*. Birkhäuser, 1996.
- [2] E. D. Sontag, “A ‘universal’ construction of Artstein theorem on nonlinear stabilization,” *Systems & Control Letters*, vol. 13, pp. 117–123, 1989.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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 Conf.*, 2015, pp. 2038–2044.
- [7] 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.
- [8] S.-C. Hsu, X. Xu, and A. D. Ames, “Control barrier function based quadratic programs with application to bipedal robotic walking,” in *American Control Conf.*, 2015, pp. 4542–4548.
- [9] 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.
- [10] M. J. Khojasteh, V. Dhiman, M. Franceschetti, and N. Atanasov, “Probabilistic safety constraints for learned high relative degree system dynamics,” in *Conf. on Learning for Dyn. and Control*, Jun 2020, pp. 781–792.
- [11] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, “End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks,” in *AAAI*, 2019, pp. 3387–3395.
- [12] R. Cheng, M. J. Khojasteh, A. D. Ames, and J. W. Burdick, “Safe multi-agent interaction through robust control barrier functions with learned uncertainties,” in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 777–783.
- [13] A. Taylor, A. Singletary, Y. Yue, and A. Ames, “Learning for safety-critical control with control barrier functions,” in *Conference on Learning for Dynamics and Control*. PMLR, 10–11 Jun 2020, pp. 708–717.
- [14] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson, “Optimization-based full body control for the darpa robotics challenge,” *Journal of Field Robotics*, vol. 32, no. 2, pp. 293–312, 2015.
- [15] 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.
- [16] S. J. Qin and T. A. Badgwell, “A survey of industrial model predictive control technology,” *Control engineering practice*, vol. 11, no. 7, pp. 733–764, 2003.
- [17] Y. Wang and S. Boyd, “Fast model predictive control using online optimization,” *IEEE Transactions on control systems technology*, vol. 18, no. 2, pp. 267–278, 2010.
- [18] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*. John Wiley & Sons, 2012.
- [19] K. Zhou, J. Doyle, and K. Glover, *Robust and Optimal Control*. Prentice Hall PTR, 1996.
- [20] E. D. Sontag, “On the input-to-state stability property,” *European Journal of Control*, vol. 1, pp. 24–36, 1995.
- [21] E. D. Sontag and Y. Wang, “On characterizations of the input-to-state stability property,” *Systems & Control Letters*, vol. 24, pp. 351–359, 1995.
- [22] 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.
- [23] 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.
- [24] J. P. Hespanha, D. Liberzon, and A. R. Teel, “Lyapunov conditions for input-to-state stability of impulsive systems,” *Automatica*, vol. 44, pp. 2735–2744, 2008.
- [25] C. Cai and A. R. Teel, “Characterizations of input-to-state stability for hybrid systems,” *Systems & Control Letters*, vol. 58, pp. 47–53, 2009.
- [26] C. Edwards and S. Spurgeon, *Sliding Mode Control: Theory And Applications*. CRC Press, 1998.
- [27] S. V. Drakunov and V. I. Utkin, “Sliding mode control in dynamic systems,” *International Journal of Control*, vol. 55, pp. 1029–1037, 1992.
- [28] C.-Y. Su and T.-P. Leung, “A sliding mode controller with bound estimation for robot manipulators,” *IEEE Transactions on Robotics and Automation*, vol. 9, no. 2, pp. 208–214, 1993.
- [29] F. Fahimi, “Sliding-mode formation control for underactuated surface vessels,” *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 617–622, 2007.
- [30] M. W. Spong, “On the robust control of robot manipulators,” *IEEE Transactions on Automatic Control*, vol. 37, no. 11, pp. 1782–1786, November 1992.
- [31] F. Lin and R. D. Brandt, “An optimal control approach to robust control of robot manipulators,” *IEEE Trans. on*

- Rob. and Autom.*, vol. 14, no. 1, pp. 69–77, Feb 1998.
- [32] 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.
- [33] M. Kelly and A. Ruina, “Non-linear robust control for inverted-pendulum 2d walking,” *IEEE International Conference on Robotics and Automation*, 2015.
- [34] A. Majumdar and R. Tedrake, “Funnel libraries for real-time robust feedback motion planning,” *Int. J. of Rob. Res.*, 2016.
- [35] S. Kousik, S. Vaskov, M. Johnson-Roberson, and R. Vasudevan, “Safe trajectory synthesis for autonomous driving in unforeseen environments,” in *ASME 2017 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers Digital Collection.
- [36] D. Fridovich-Keil, S. L. Herbert, J. F. Fisac, S. Deglurkar, and C. J. Tomlin, “Planning, fast and slow: A framework for adaptive real-time safe trajectory planning,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 387–394.
- [37] 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.
- [38] —, “Iterative robust stabilization algorithm for periodic orbits of hybrid dynamical systems: Application to bipedal running,” *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 161–168, 2015.
- [39] 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.
- [40] 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.
- [41] M. Benallegue and J.-P. Laumond, “Metastability for high-dimensional walking systems on stochastically rough terrain,” in *Robotics: Science and Systems*, 2013.
- [42] K. Byl and R. Tedrake, “Metastable walking machines,” *The International Journal of Robotics Research*, vol. 28, no. 8, pp. 1040–1064, August 2009.
- [43] Q. Nguyen and K. Sreenath, “Optimal robust control for bipedal robots through control lyapunov function based quadratic programs,” in *Robotics: Science and Systems*, 2015.
- [44] —, “Optimal robust control for constrained nonlinear hybrid systems with application to bipedal locomotion,” in *American Control Conference*, 2016.
- [45] 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.
- [46] 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.
- [47] 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.
- [48] Q. Nguyen and K. Sreenath, “L1 adaptive control for bipedal robots with control lyapunov function based quadratic programs,” in *American Control Conference*, 2015, pp. 862–867.
- [49] H. Khalil, *Nonlinear Systems - 3rd Edition*. Upper Saddle River, NJ: Pearson, 2002.
- [50] J. S. Reed and P. A. Ioannou, “Instability analysis and robust adaptive control of robotic manipulators,” *IEEE transactions on robotics and automation*, vol. 5, no. 3, pp. 381–386, 1989.
- [51] 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.
- [52] 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.
- [53] 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.



control, robust and adaptive control.

**Quan Nguyen** is an Assistant Professor of Aerospace and Mechanical Engineering at the University of Southern California. Prior to joining USC, he was a Postdoctoral Associate in the Biomimetic Robotics Lab at the Massachusetts Institute of Technology (MIT). He received his Ph.D. from Carnegie Mellon University (CMU) in 2017 with the Best Dissertation Award. His research interests span different control and optimization approaches for highly dynamic robotics including nonlinear control, trajectory optimization, real-time optimization-based



**Koushil Sreenath** is an Assistant Professor of Mechanical Engineering at University of California, Berkeley. He received the Ph.D. degree in Electrical Engineering: Systems and the M.S. degree in Applied Mathematics from the University of Michigan at Ann Arbor, MI, in 2011. He was an Assistant Professor at Carnegie Mellon University between 2013–2017. His research interest lies at the intersection of highly dynamic robotics and applied nonlinear control.