

DSCC2016-9910

OPTIMAL ROBUST TIME-VARYING SAFETY-CRITICAL CONTROL WITH APPLICATION TO DYNAMIC WALKING ON MOVING STEPPING STONES

Quan Nguyen

Department of Mechanical Engineering,
Carnegie Mellon University,
Pittsburgh, Pennsylvania 15213.
Email: quannguyen@cmu.edu.

Koushil Sreenath

Depts. of Mechanical Engineering, Robotics
Institute, and Electrical & Computer Engineering,
Carnegie Mellon University,
Pittsburgh, Pennsylvania 15213.
Email: koushils@cmu.edu.

ABSTRACT

This paper presents a novel methodology to handle time-varying safety-critical constraints under high level of model uncertainty with application to dynamic bipedal walking with strict foot-placement constraints. This paper builds off recent work on optimal robust control through quadratic programs that can handle stability, input / state dependent constraints, as well as safety-critical constraints, in the presence of high level of model uncertainty. Under the assumption of bounded uncertainty, the proposed controller strictly guarantees time-varying constraints without violating them. We evaluate our proposed control design for achieving dynamic walking of an underactuated bipedal robot subject to (a) torque saturation constraints (input constraints), (b) contact force constraints (state constraints), and (c) precise time-varying footstep placements (time-varying and safety-critical constraints). We present numerical results on RABBIT, a five-link planar bipedal robot, subject to a large unknown load on its torso. Our proposed controller is able to demonstrate walking while strictly enforcing the above constraints with an unknown load of up to 15 Kg (47% of the robot mass.)

1 Introduction

Control design for nonlinear systems to achieve stability while simultaneously enforcing input, state, and safety constraints is challenging. Doing this under high levels of model

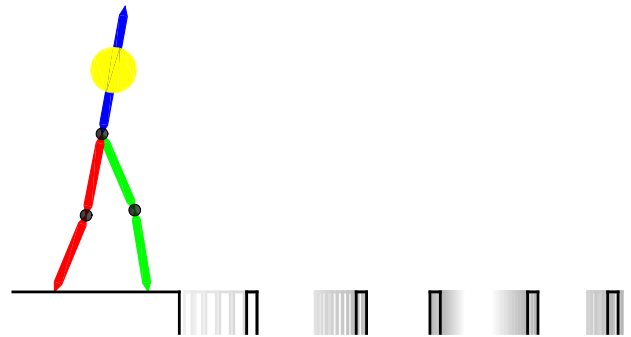


FIGURE 1: Dynamic bipedal walking while carrying unknown load, subject to torque saturation constraints (input constraints), contact force constraints (state constraints), and moving stepping stones (time-varying safety-critical constraints). Dynamical walking over 10 steps is carried out, subject to model uncertainty of carrying an unknown load of 15 Kg (47 %). Simulation video: <https://youtu.be/nBShDTn7II1>.

uncertainty is even harder. The proposed controller offers a way to address stability while strictly enforcing input, state, and *time-varying* safety constraints without violations even under the presence of *model uncertainty*. Although several approaches exist separately for constrained control (MPC, reference governors, etc.) and robust control (H_∞ , LQG, sliding mode, etc.), addressing these constraints simultaneously for nonlinear systems in real-time is challenging.

Recently, a novel method of Control Barrier Function incor-

This work is supported in part by NSF grants IIS-1526515, IIS-1464337, CMMI-1538869, and in part by the Google Faculty Research Award.

porated with Control Lyapunov Function based Quadratic Programs (CBF-CLF-QPs) was introduced in [1], that can handle state-dependent constraints effectively in real-time. Preliminary experimental validations were carried out on the problem of Adaptive Cruise Control in [13]. The methodology has also been extended to safety constraints on Riemannian manifolds [23] and dynamic walking of bipedal robots [17], [9]. Time-varying safety-critical constraints are addressed in [24]. Although these controllers provide impressive results in terms of stability and constraints, they are highly dependent on the knowledge of an accurate system model. A robust version of the CBF-CLF-QP controller was presented in [19] which handles model uncertainty, but does not handle time-varying safety-critical constraints.

Here, we develop a controller that robustly enforces time-varying safety-critical constraints, in addition to input and state constraints. We provide a validation through dynamic bipedal walking with precise footstep placements on moving stepping stones. Designing real-time feedback controllers for achieving dynamic walking with precise footstep placement is challenging. Trying to do this for time-varying (moving) stepping stones when subject to high levels of model uncertainty is even harder. We develop an optimal robust controller using control Lyapunov functions for stability and control Barrier functions for safety and strictly enforce footstep placement and friction constraints. Furthermore, we consider large model uncertainty and still guarantee no constraint violation.

Foot placements for fully actuated legged robots essentially rely on quasi-static walking using the ZMP criterion that requires slow walking speed and small steps [11, 12, 4]. Impressive results in footstep planning and placements in obstacle filled environments with vision-based sensing was carried out in [14, 3]. However, these methods are not applicable for dynamic walking with faster walking gaits. The DARPA Robotics Challenge has inspired several new methods, some based on mixed-integer quadratic programs [6]. However, as mentioned in [7, Chap. 4], the proposed method of mixed-integer based footstep planning does not offer dynamic feasibility even on a simplified model.

Prior work in [19] presented an optimal robust controller to enforce input, state, and safety constraints under model uncertainty. However robust time-varying safety constraints were not handled. Prior work in [17] presented dynamic bipedal walking over stepping stones, but did not handle time-varying (moving) stepping stones or model uncertainty. Our preliminary results in [21] addresses time-varying stepping stones, but not model uncertainty. With respect to prior work, the primary contributions of this paper are:

- Introduction of a new technique of optimal robust control via quadratic programs for strict enforcement of input constraints, state constraints, and time-varying safety constraints without any violations, all under the presence of model uncertainty.

- Novel footstep placement approach based on real-time feedback controller that can tackle time-varying (moving) stepping stones by explicitly taking into account their motion.
- The application of the optimal robust control via quadratic programs to achieve dynamic bipedal walking over time-varying (moving) stepping stones while carrying a large unknown load.

The rest of the paper is organized as follows. Sections 2 and 3 revisit Control Lyapunov Functions and Control Barrier Functions. Section 4 presents the optimal robust control using a CBF-CLF-QP. Section 5 describes how the proposed controller is applied for the problem of robust dynamic walking with time-varying stepping stones. Section 6 presents numerical validation on RABBIT, a 5-link bipedal walker. Finally, Section 7 provides concluding remarks.

2 Control Lyapunov Function based Quadratic Programs Revisited

In this section we start by introducing a hybrid dynamical model that captures the dynamics of a bipedal robot. We then review recent innovations on control Lyapunov functions for hybrid systems and control Lyapunov function based quadratic programs, introduced in [2] and [8] respectively.

2.1 Model

This paper will focus on the specific problem of walking of bipedal robots such as RABBIT (described in [5]), which are characterized by single-support continuous-time dynamics, when one foot is assumed to be in contact with the ground, and double-support discrete-time impact dynamics, when the swing foot undergoes an instantaneous impact with the ground. Such a hybrid model is written as,

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

where $q \in \mathcal{Q}$ is the robot's configuration variables, $u \in \mathbb{R}^m$ is the control inputs, representing the motor torques, (q^-, \dot{q}^-) represents the state before impact and (q^+, \dot{q}^+) represents the state after impact, S represents the switching surface when the swing leg contacts the ground, and Δ represents the discrete-time impact map.

We also define output functions $y \in \mathbb{R}^m$ of the form

$$y(q) := H_0 q - y_d(\theta(q)), \quad (2)$$

where $\theta(q)$ is a strictly monotonic function of the configuration variable q , H_0 is an appropriately-sized matrix prescribing linear combinations of state variables to be controlled, and $y_d(\cdot)$ prescribes the desired evolution of these quantities (see [22] for details.) The method of Hybrid Zero Dynamics (HZD) aims to drive these output functions (and their first derivatives) to zero, thereby imposing “virtual constraints” such that the system evolves on the lower-dimensional zero dynamics manifold, given by

$$Z = \{(q, \dot{q}) \in T\mathcal{Q} \mid y(q) = 0, L_f y(q, \dot{q}) = 0\}, \quad (3)$$

where L_f denotes the Lie derivative, [10]. In particular, the dynamics of the system \mathcal{H} in (1) restricted to Z , given by $\mathcal{H}|_Z$, is the underactuated dynamics of the system and is forward-invariant. Periodic motion such as walking is then a hybrid periodic orbit \mathcal{O} in the statespace with \mathcal{O}_Z being its restriction to Z .

2.2 Input-output linearization

If $y(q)$ has vector relative degree 2, then its second derivative takes the form

$$\ddot{y} = L_f^2 y(q, \dot{q}) + L_g L_f y(q, \dot{q}) u. \quad (4)$$

We can then apply the following pre-control law

$$u(q, \dot{q}) = u^*(q, \dot{q}) + (L_g L_f y(q, \dot{q}))^{-1} \mu, \quad (5)$$

where

$$u^*(q, \dot{q}) := -(L_g L_f y(q, \dot{q}))^{-1} L_f^2 y(q, \dot{q}), \quad (6)$$

and μ is a stabilizing control to be chosen. Defining transverse variables $\eta = [y, \dot{y}]^T$, and using the IO linearization controller above with the pre-control law (5), we have,

$$\ddot{y} = \mu, \quad (7)$$

$$\implies \begin{bmatrix} \dot{y} \\ y \end{bmatrix} = \dot{\eta} = F\eta + G\mu, \quad \text{where } F = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}, G = \begin{bmatrix} 0 \\ I \end{bmatrix}. \quad (8)$$

The closed-loop dynamics can be expressed in terms of the transverse variables η and the states $z \in Z$ from (3) (instead of in terms

of the state (q, \dot{q})), to take the form,

$$\begin{aligned} \dot{\eta} &= \bar{f}(\eta, z) + \bar{g}(\eta, z)\mu, \\ \dot{z} &= p(\eta, z), \end{aligned} \quad (9)$$

where $\bar{f}(\eta, z) = F\eta$ and $\bar{g}(\eta, z) = G$, with

$$F = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ I \end{bmatrix}, \quad (10)$$

and $p(\eta, z)$ arises from the mapping between (q, \dot{q}) and (η, z) .

We then can design a linear controller $\mu = K\eta$ so that the closed loop system

$$\dot{\eta} = (F - GK)\eta =: A\eta \quad (11)$$

is exponentially stable, i.e., A is Hurwitz.

2.3 CLF-based Quadratic Programs

A control approach based on control Lyapunov functions, introduced in [2], provides guarantees of exponential stability for the transverse variables η . In particular, a function $V_e(\eta)$ is an *exponentially stabilizing control Lyapunov function (ES-CLF)* for the system (9) if there exist positive constants $c_1, c_2, \lambda > 0$ such that

$$c_1 \|\eta\|^2 \leq V(\eta) \leq c_2 \|\eta\|^2, \quad (12)$$

$$\dot{V}(\eta, \mu) + \lambda V(\eta) \leq 0. \quad (13)$$

In our problem, we chose a quadratic CLF candidate as follows

$$V(\eta) = \eta^T P \eta, \quad (14)$$

where P is the solution of the Lyapunov equation $A^T P + P A = -Q$ (where A is given by (11) and Q is any symmetric positive-definite matrix). The time derivative of the CLF (14) is computed as

$$\dot{V}(\eta, \mu) = L_{\bar{f}} V(\eta) + L_{\bar{g}} V(\eta) \mu, \quad (15)$$

where

$$\begin{aligned} L_{\bar{f}} V(\eta) &= \eta^T (F^T P + P F) \eta, \\ L_{\bar{g}} V(\eta) &= 2\eta^T P G. \end{aligned} \quad (16)$$

The CLF condition takes the form

$$L_{\bar{f}}V(\eta) + L_{\bar{g}}V(\eta)\mu + \lambda V(\eta) \leq 0. \quad (17)$$

If this inequality holds, then it implies that the output η will be exponentially driven to zero by the controller. The CLF-QP based controller presented in [8] takes the form:

$$\begin{aligned} \mu^* = \operatorname{argmin}_{\mu, d_1} \quad & \mu^T \mu + p \delta^2 \\ \text{s.t.} \quad & \dot{V}(\eta, \mu) + \lambda V(\eta) \leq \delta \quad \text{(CLF)} \\ & A_{AC}(q, \dot{q}) \mu \leq b_{AC}(q, \dot{q}) \quad \text{(Constraints)} \end{aligned} \quad (18)$$

where p is a large positive number that represents the penalty of relaxing the CLF condition (13) and A_{AC} , b_{AC} represent additional constraints such as torque constraints, contact force constraints, friction constraints and joint limit constraints.

This formulation opened a novel method to guarantee stability of the nonlinear systems with respect to additional constraints, such as torque saturation in [8] and L_1 adaptive control in [15].

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

3 Control Barrier Function

3.1 Time-Varying Control Barrier Function

Here, we present the definition of time-varying CBFs introduced in [24].

Consider an affine control system:

$$\dot{x} = f(x) + g(x)u \quad (19)$$

with the goal to design a controller to keep the state x in the set

$$\mathcal{C}_t = \{x \in \mathbb{R}^n : h(t, x) \geq 0\} \quad (20)$$

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

$$\frac{1}{\alpha_1(h(t, x))} \leq B(t, x) \leq \frac{1}{\alpha_2(h(t, x))}, \quad (21)$$

$$\dot{B}(t, x, u) = \frac{\partial B}{\partial t} + L_f B(x) + L_g B(x) u \leq \frac{\gamma}{B(t, x)}. \quad (22)$$

The important properties of the CBF condition in (22) is that if there exists a Control Barrier Function, $B : \mathcal{C}_t \rightarrow \mathbb{R}$, then \mathcal{C}_t is forward invariant, or in other words, if $x(t_0) = x_0 \in \mathcal{C}_t$, i.e., $h(t_0, x_0) \geq 0$, then $x = x(t) \in \mathcal{C}_t, \forall t \geq t_0$, i.e., $h(t, x(t)) \geq 0, \forall t \geq t_0$.

3.2 Time-Varying Exponential Control Barrier Function

The above formulation of control Barrier functions required relative-degree one $h(t, x)$. In order to systematically design safety-critical controllers for higher order relative degree constraints, we will use ‘‘Exponential Control Barrier Functions’’ (ECBFs) [18].

With application to precise footstep placement, our constraints will be position based, $h(t, q) \geq 0$, which has relative degree 2. For this problem, we can design an Exponential CBF as follows:

$$B(t, q, \dot{q}) = \dot{h}(t, q, \dot{q}) + \gamma_1 h(t, q). \quad (23)$$

The Exponential CBF condition is then defined as:

$$\dot{B}(t, q, \dot{q}, u) + \gamma B(t, q, \dot{q}) \geq 0, \quad (24)$$

where $\gamma_1 > 0, \gamma > 0$. Enforcing (24) will then enforce $B(t, q, \dot{q}) \geq 0$, which then ensures forward-invariance of \mathcal{C}_t in (20) (see [23] for details.) Moreover, we also note that by plugging the ECBF (23) into the condition (24), we have,

$$\left(\frac{d}{dt} + \gamma_1\right) \circ \left(\frac{d}{dt} + \gamma\right) \circ h(t, q) \geq 0. \quad (25)$$

Thus, γ_1, γ play the role of pole locations for the constraint dynamics $\dot{h}(t, q, \dot{q}, u)$.

3.3 Combination of CBF and CLF-QP

Consider the Exponential Control Barrier Candidate Function (23), then we can incorporate the condition (24) into the Quadratic Program as follows,

$$\begin{aligned} \mu^* = \operatorname{argmin}_{\mu, d_1} \quad & \mu^T \mu + p \delta^2 \\ \text{s.t.} \quad & \dot{V}(\eta, \mu) + \lambda V(\eta) \leq \delta \quad \text{(CLF)} \\ & \dot{B}(t, x, \mu) + \gamma B(t, x) \geq 0 \quad \text{(CBF)} \\ & u_{\min} \leq u \leq u_{\max} \quad \text{(TS)} \end{aligned} \quad (26)$$

Having revisited control Barrier function based quadratic programs, we will now present the robust CBF-CLF-QP controller to enforce constraints under the presence of model uncertainty.

4 Robust Optimal Control based Quadratic Programs

The optimization-based control approaches presented in Section 3 have interesting properties. However, a primary disadvantage of these controllers is that they require an accurate dynamical model of the system. Uncertainty in the model can cause poor quality of control leading to tracking errors, and could potentially lead to instability [16]. Moreover, uncertainty in the model also makes enforcing input and state constraints on the true system hard. Furthermore, uncertainty could potentially lead to violation of the safety-critical constraints [25]. In this section, we will present a systematic methodology to overcome this challenge for both CLF (stability), CBF (safety) and other constraints.

4.1 Effect of uncertainty on CLF-QP

We begin by considering uncertainty in the dynamics and assume that the vector fields, $f(x), g(x)$ of the true dynamics (1), 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 (5) get's reformulated as

$$u(x) = u^*(x) + (L_{\tilde{g}}L_{\tilde{f}}y(x))^{-1}\mu, \quad (27)$$

with

$$u^*(x) := -(L_{\tilde{g}}L_{\tilde{f}}y(x))^{-1}L_{\tilde{f}}^2y(x), \quad (28)$$

where we have used the known nominal model rather than the unknown true dynamics. Substituting $u(x)$ from (27) into (4), the input-output linearized system then becomes

$$\dot{y} = \mu + \Delta_1 + \Delta_2\mu, \quad (29)$$

where

$$\begin{aligned} \Delta_1 &:= L_{\tilde{f}}^2y(x) - L_gL_fy(x)(L_{\tilde{g}}L_{\tilde{f}}y(x))^{-1}L_{\tilde{f}}^2y(x), \\ \Delta_2 &:= L_gL_fy(x)(L_{\tilde{g}}L_{\tilde{f}}y(x))^{-1} - I. \end{aligned} \quad (30)$$

Remark 1. In the definitions of Δ_1, Δ_2 , note that when there is no model uncertainty, i.e., when $\tilde{f} = f, \tilde{g} = g$, then $\Delta_1 = \Delta_2 = 0$.

4.2 Robust CLF-QP

Under the assumption of bounded model uncertainty, we can express the CLF condition in term of a min-max optimization problem as follows:

$$\begin{aligned} \underset{\mu, \delta}{\operatorname{argmin}} \quad & \mu^T \mu + p\delta^2 \\ \text{s.t.} \quad & \max_{\substack{\|\Delta_1\| \leq \Delta_1^{\max} \\ \|\Delta_2\| \leq \Delta_2^{\max}}} \dot{V}(x, \Delta_1, \Delta_2, \mu) + \lambda V(x) \leq \delta \\ & u_{\min} \leq u \leq u_{\max} \end{aligned} \quad (31)$$

Remark 2. Note that the above controller minimizes the control effort while enforcing the CLF condition on \dot{V} under all possible bounded disturbance. Since \dot{V} is affine in the uncertainty Δ_1, Δ_2 and the control input μ , the above optimization problem is still a quadratic program, enabling solving it in real-time speeds.

4.3 Virtual Input-Output Linearization

In order to extend the same methodology of robust CLF-QP into robust constraints, we use Virtual Input-Output Linearization [19]. We will explain this method for CBF constraints (a similar formulation holds for input and state constraints). We begin with the CBF $B(t, x)$ and define a virtual control input μ_b such that (22), with u substituted from (5), can be written as,

$$\dot{B}(t, x, \mu) = A_b(x)\mu + b_b(t, x) =: \mu_b(t, x, \mu), \quad (32)$$

where

$$\begin{aligned} A_b(x) &:= L_gB(x) L_gL_f^{-1}y(x), \\ b_b(t, x) &:= \frac{\partial B}{\partial t} + L_fB(x) + L_gB(x) u^*(x), \end{aligned} \quad (33)$$

with $u^*(x)$ as defined in (6).

The CBF condition (22) then becomes,

$$\mu_b(t, x, u) + \gamma B(t, x) \geq 0. \quad (34)$$

By using VIOL, the time-derivative of the CBF, (32), now takes the similar form of a linear system, $\dot{B}(t, x, \mu_b) = \mu_b$, and therefore the effect of uncertainty can be easily extended by using the same approach as with the robust CLF-QP to obtain,

$$\dot{B}(t, x, \Delta_1^b, \Delta_2^b, \mu_b) = \mu_b + \Delta_1^b + \Delta_2^b\mu_b, \quad (35)$$

where Δ_1^b, Δ_2^b are functions of both the true and nominal system models.

Remark 3. We should note from the equation (33) that the time-varying term $\frac{\partial B}{\partial t}$ is independent of the system dynamics that is represented by Lie derivatives on the functions f and g . Therefore, the model uncertainty will not affect $\frac{\partial B}{\partial t}$, and thus the robustness analysis still holds with time-varying constraints.

4.4 Robust Time-varying CBF-CLF-QP

The same procedure that was used to “robustify” stability and safety constraints can also be applied for robust state constraints. We then have the following unified robust optimal control based quadratic program:

$$\begin{aligned}
& \underset{\mu, \mu_b, \mu_c, \delta}{\operatorname{argmin}} && \mu^T \mu + p \delta^2 \\
& \text{s.t.} && \text{(Robust CLF)(Stability)} \\
& && \max_{\substack{\|\Delta_1\| \leq \Delta_1^{\max} \\ \|\Delta_2\| \leq \Delta_2^{\max}}} \dot{V}(x, \Delta_1, \Delta_2, \mu) + \lambda V(x) \leq \delta \\
& && \text{(Robust CBF)(Footstep placement)} \\
& && \max_{\substack{\|\Delta_1^b\| \leq \Delta_1^{b,\max} \\ \|\Delta_2^b\| \leq \Delta_2^{b,\max}}} \dot{B}(t, x, \Delta_1^b, \Delta_2^b, \mu_b) + \gamma B(x) \geq 0 \\
& && A_b(x) \mu + b_b(t, x) = \mu_b \\
& && \text{(Robust Constraints)(Friction)} \\
& && \max_{\substack{\|\Delta_1^c\| \leq \Delta_1^{c,\max} \\ \|\Delta_2^c\| \leq \Delta_2^{c,\max}}} \bar{\mu}_c(x, \Delta_1^c, \Delta_2^c, \mu_c) \leq 0 \\
& && A_c(x) \mu + b_c(x) = \mu_c \\
& && \text{(Input Saturation)} \\
& && u_{\min} \leq u \leq u_{\max}
\end{aligned} \tag{36}$$

Having presented a control framework to handle robust time-varying constraints under model uncertainty, in the next Section, we will see how to employ this CBF-CLF-QP based controller for the problem of dynamic walking with precise time-varying foot placement.

5 Dynamical Bipedal Walking over Time-Varying Stepping Stones

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 (see Fig. 1). We define the step length at impact as,

$$l_s := l_f(q) \Big|_{h_f(q)=0, \dot{h}_f(q, \dot{q}) < 0}. \tag{37}$$

If we want to force the robot to step onto a specific position, we need to guarantee that the step length when the robot swing foot

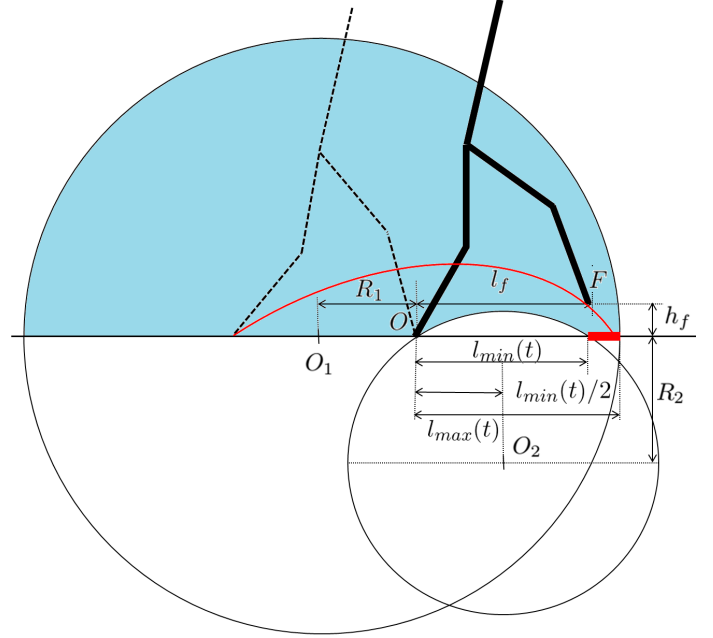


FIGURE 2: Geometric explanation of CBF constraints for the problem of bipedal walking over discrete footholds. If we can guarantee the trajectory of the swing foot F (the red line) to be limited in the blue domain, we will force our robot to step onto a discrete foothold position (thick red range on the ground). This approach therefore also provides a safety guarantee against foot scuffing or swing foot being always above the ground prior to contact.

hits the ground is bounded within a given range:

$$l_{\min}(T_I) \leq l_s \leq l_{\max}(T_I). \tag{38}$$

where T_I is the time at impact.

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 time-varying stepping stone constraint (38). The solution for this problem can be explained geometrically in Fig. 2. 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 $[l_{\min}(t), l_{\max}(t)]$. These constraints can be represented mathematically as the following:

$$\begin{aligned}
O_1 F &= \sqrt{(R_1 + l_f)^2 + h_f^2} \leq R_1 + l_{\max}(t), \\
O_2 F &= \sqrt{(R_2 + h_f)^2 + (l_f - \frac{l_{\min}(t)}{2})^2} \geq \sqrt{R_2^2 + (\frac{l_{\min}(t)}{2})^2}.
\end{aligned}$$

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,

$$\begin{aligned} \sqrt{(R_1 + l_s)^2} &\leq R_1 + l_{max}(t), \\ \sqrt{R_2^2 + (l_s - \frac{l_{min}(t)}{2})^2} &\geq \sqrt{R_2^2 + (\frac{l_{min}(t)}{2})^2}. \end{aligned} \quad (39)$$

Enforcing the above constraints guarantees the constraint (38), no matter if the desired stone location is constant or varies over time.

We now define the two position constraints based on this approach:

$$\begin{aligned} h_1(t, q) &= R_1 + l_{max}(t) - \sqrt{(R_1 + l_f(q))^2 + h_f(q)^2} \geq 0, \\ h_2(t, q) &= \sqrt{(R_2 + h_f)^2 + (l_f - \frac{l_{min}(t)}{2})^2} \\ &\quad - \sqrt{R_2^2 + (\frac{l_{min}(t)}{2})^2} \geq 0. \end{aligned} \quad (40)$$

Exponential Control Barrier Functions can then be formulated for the above position constraints as,

$$\begin{aligned} B_1(t, q, \dot{q}) &= \gamma_1 h_1(t, q) + \dot{h}_1(t, q, \dot{q}) \geq 0, \\ B_2(t, q, \dot{q}) &= \gamma_1 h_2(t, q) + \dot{h}_2(t, q, \dot{q}) \geq 0. \end{aligned} \quad (41)$$

We now can apply the robust CBF-CLF-QP based controller (36) to enforce these constraints under model uncertainty.

6 Application: Robust Dynamic Bipedal Walking over Time-Varying Stepping Stones

To demonstrate the effectiveness of the proposed controller, we will conduct numerical simulations on the model of RABBIT, [5], 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,

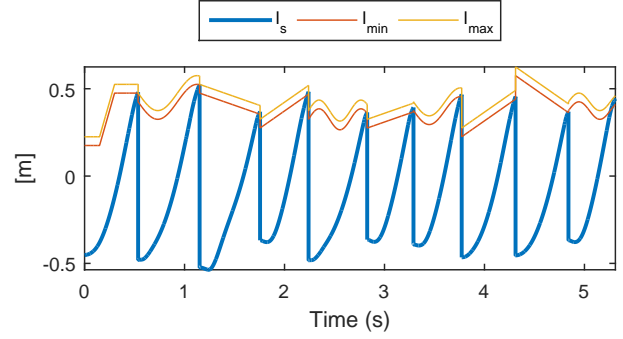


FIGURE 3: Dynamic walking of bipedal robot while carrying unknown load of 15 Kg (47 %). Ten steps of the horizontal position of the swing leg (thick blue) with respect to the stance leg, and desired stone locations varying either linearly or sinusoidally over time (thin orange / red) are shown. The proposed controller guarantees strict enforcement of the foot placement constraints on the moving stepping stones, even with large model uncertainty.

and input saturation. Model uncertainty appears in the form of an unknown heavy load added to the torso of RABBIT.

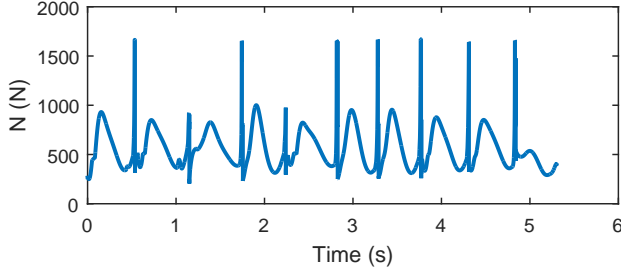
In bipedal robotic walking, contact force constraints are very important. Any violation of these constraints will result in the leg slipping and the robot potentially falling. Although a nominal walking gait is usually designed to respect these constraints, however, we cannot guarantee these constraints under transients or under model uncertainty. Our proposed controller addresses this through the robust constraints formulation in (36). In particular, we consider, $F(x, u)$ and $N(x, u)$, the friction force and vertical contact force between the stance foot and the ground. Then, the constraints to avoid slipping during walking are,

$$N(x, u) \geq \delta_N > 0, \quad (42)$$

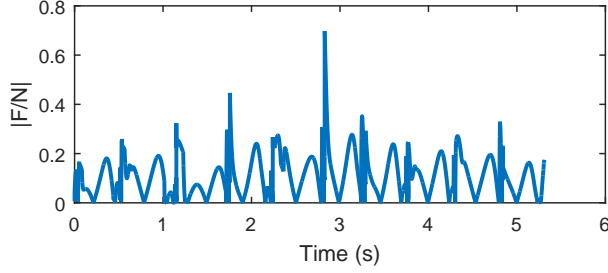
$$\left| \frac{F(x, u)}{N(x, u)} \right| \leq k_f, \quad (43)$$

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 5. 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.



(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 4: 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.

Our proposed robust controller in (36) strictly enforces the state constraints corresponding to the ground contact and friction constraints, as seen in Figure 3, and the time-varying safety-critical constraints corresponding to stepping onto the moving stepping stones, as seen in Figure 4. These figures illustrate only 10 steps of walking since the time-varying profile of only ten stepping stones were selected as below,

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

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

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

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

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

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

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

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

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

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

where τ is the time reinitialized 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 ± 2.5 cm about the desired footstep location l_i^d , i.e.,

$$l_{min}^i(\tau) = l_i^d(\tau) - 0.025, \quad l_{max}^i(\tau) = l_i^d(\tau) + 0.025, \quad (44)$$

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

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 [20], 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.

7 Conclusion

We have presented a novel method of optimal robust control through quadratic programs to handle time-varying safety-critical constraints under high level of model uncertainty with application to dynamic bipedal walking with strict foot-placement constraints. The proposed controller was evaluated for achieving dynamic walking of RABBIT, an underactuated bipedal robot subject to (a) torque saturation constraints (input constraints), (b) contact force constraints (state constraints), and (c) precise time-varying footstep placements (time-varying and safety-critical constraints) with desired stone locations varying either linearly or sinusoidally over time. All the above constraints were strictly enforced while in the presence of a large model uncertainty, in the form of an unknown load of 15 Kg (47% of the robot mass) on the torso of the robot.

REFERENCES

- [1] 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.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. R. Westervelt, C. Canudas-de-Wit, and J. W. Grizzle, "RABBIT: A testbed for advanced control theory," vol. 23, no. 5, pp. 57–79, October 2003.
- [6] 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.
- [7] R. L. H. Deits, "Convex segmentation and mixed-integer footstep planning for a walking robot," Master's thesis, Massachusetts Institute of Technology, 2014.
- [8] 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.
- [9] 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.
- [10] A. Isidori, *Nonlinear Control Systems: An Introduction*, 2nd ed. Berlin, Germany: Springer-Verlag, 1989.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] P. Michel, J. Chestnutt, J. Kuffner, and T. Kanade, "Vision-guided humanoid footstep planning for dynamic environments," in *Humanoids*, 2005, pp. 13–18.
- [15] 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.
- [16] —, "Optimal robust control for bipedal robots through control lyapunov function based quadratic programs," in *Robotics: Science and Systems*, 2015.
- [17] —, "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.
- [18] —, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *American Control Conference*, 2016.
- [19] —, "Optimal robust control for constrained nonlinear hybrid systems with application to bipedal locomotion," in *American Control Conference*, 2016.
- [20] —, "3d dynamic walking on stepping stones with control barrier functions," in *2016 IEEE 55th Conference on Decision and Control*, Submitted, 2016.
- [21] —, "Dynamic bipedal walking over time-varying stepping stones," in *Dynamic Walking Conference*, To appear, 2016.
- [22] 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.
- [23] 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.
- [24] —, "Safety-critical control of a planar quadrotor," in *American Control Conference*, 2016.
- [25] 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.