Towards Real-Time Parameter Optimization for Feasible Nonlinear Control with Applications to Robot Locomotion

Matthew J. Powell and Aaron D. Ames

***Abstract*— This paper considers the application of classical control methods, designed for unconstrained nonlinear systems, to systems with nontrivial input constraints. As shown throughout the literature, unconstrained classical methods can be used to stabilize constrained systems, however, (without modification) these unconstrained methods are not guaranteed to work for a general control problem. In this paper, we propose conditions for which classical unconstrained methods can be guaranteed to exponentially stabilize constrained systems – which we term "feasibility" conditions – and we provide examples of how to construct explicitly feasible controllers. The control design methods leverage control Lyapunov functions (CLF) describing the "desired behavior" of the system; and we claim that in the event that a system's input constraints prevent the production of an exponentially stabilizing input for a particular CLF, a new, locally feasible CLF must be produced. To this end, we propose a novel hybrid feasibility controller consisting of a continuous-time controller which implements a CLF and a discrete parameter update law which finds feasible controller parameters as needed. Simulation results suggest that the proposed method can be used to overcome certain catastrophic infeasibility events encountered in robot locomotion.**

I. Introduction

A constant challenge in the construction of nonlinear control systems for physical applications is the task of producing control laws which are *feasible*: laws in which stabilizing control actions are within a system's input limits. In recent years, quadratic programming has become an increasingly prevalent tool in the design of feasible control systems for robot locomotion [1], [3], [11]. In affine control systems, such as those used to model walking robots, stabilizing constraints can be expressed as linear inequality constraints on the control input and included as constraints in a quadratic program (QP)-based controller. Unlike classical, monolithic control laws, controllers implemented via quadratic programs can also include constraints on the input that reflect physical limitations of the system being controlled. Thus QP-based controllers can be used to stabilize real-world systems without having to algebraically produce a control law which is both stabilizing and which satisfies input bounds (often a difficult task). However, input and stability constraints are not always consistent. Intuitively, if the control objective is aggressive and the admissible control set is conservative, it is possible that the corresponding set of inequalities does not have a solution; i.e. the physical system cannot produce a stabilizing input; such conditions are termed "infeasible points". When a quadratic program encounters an infeasible point, it will generally exit without a solution; consequently, controllers based on quadratic programs fail when they encounter infeasible points. One approach to address infeasible points is to relax stability constraints, such as in [1]; this will allow (local) drift in control objectives to accommodate input constraints. However, the control engineer does not any influence as to how the control objectives drift.

The feasible control problem can sometimes be addressed offline, i.e. before run-time of the controller. Nonlinear optimization methods, such as [18], [19] and [21], can produce parameterized controller outputs which satisfy both convergence and physical constraints over an operating range of consideration. These methods work extremely well – when the physical system is within the operating range. However, it is often the case that operating conditions will diverge from the planned conditions. In such scenarios, there is often not enough time to recalculate a new, feasible set of controller parameters using these methods (as these methods are generally not real-time), and as such, they must also incorporate an additional strategy for overcoming real-time controller infeasibilities.

The literature is rich with a variety of approaches of producing feasible controllers. One subset of the literature leverages properties of systems experiencing input saturation to guarantee conditions for stability, see for example [4], [9] [7], or [24]. In Model Predictive Control (MPC), the control is obtained through solution of a nonlinear program for which the objective function is made to be a CLF, see for example [15], [12], or [2]. Additionally, constrained control Lyapunov Functions have been considered, see for example [14] or [3]. Other approaches to relax Lyapunov constraints would be to use dynamic CLFs as in [20] or flexible CLFs as in [13].

In this paper, we address feasibility directly. Specifically, we consider the definition of the width of the feasible set [16] for a system of linear inequalities comprised of control Lyapunov function convergence and physical constraints. We show how to leverage the quadratic structure of control Lyaupnov function constraints to construct a Quadratically Constrained Quadratic Program which can be used to modify an infeasible CLF to make it feasible. Next, we exploit linearity in controller parameters to express the CLF modification scheme in terms of modifications to control parameters. Finally, we propose a hybrid control scheme which uses the parameter modification scheme to produce feasible controller parameters in the event that the CLF using the "old parameters" becomes infeasible.

Matthew J. Powell is with the the School of Mechanical Engineering, and Aaron D. Ames is with the School of Mechanical Engineering and the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 {*mpowell35@gatech.edu,ames@gatech.edu*}
The research of Matthew J. Powell and Aaron D. Ames is supported by NSF CPS award 1239055.

II. Locally Feasible Control

In classical nonlinear control design, see for example [8], [10], or [22], analysis is performed on dynamical systems of the following form

$$\dot{x} = f(x) + g(x)u, \tag{1}$$

where $x \in X \subseteq \mathbb{R}^n$ and $u \in U \subseteq \mathbb{R}^m$. The state-space, X, and the input-space, U, are typically chosen to be n and m dimensional Euclidean spaces, respectively. Thus, classical nonlinear control design and analysis is performed on *unconstrained* nonlinear control systems. We posit the reason for analyzing unconstrained systems (rather than constrained ones) is likely one or both of the following reasons: unconstrained systems are *general* and *provable*. It is difficult enough to design a controller to provably stabilize an unconstrained dynamical system; constraints add complexity and remove generality.

In real-world applications, however, nonlinear control systems are often *constrained*. That is, the set of admissible states, X, and the set of admissible inputs, U, are defined implicitly by *constraint functions*

$$X = \{x \in \mathbb{R}^n : C_x(x) \leq 0\}, \tag{2}$$
$$U(x) = \{u \in \mathbb{R}^m : C_u(x,u) \leq 0\}, \tag{3}$$

where $C_x(x)$ and $C_u(x,u)$ are functions describing the constraints on the nonlinear control system. These constraints are *properties* of real systems, and thus, *cannot be violated*.

Interestingly, classical nonlinear control techniques likewise rely on inequality constraints. For example, when the control design specification is to exponentially stabilize (in the sense of Lyapunov) the origin of (1), one (either directly or indirectly) constructs a control Lyapunov function (CLF) $V(x) > 0$ with the property that $\dot{V}(x,u) \leq -\frac{c_3}{\varepsilon}V(x)$. This constraint implicitly defines a set, $K(x)$, of control values:

$$K(x) := \{u \in \mathbb{R}^m : L_f V(x) + L_g V(x)u + \frac{c_3}{\varepsilon}V(x) \leq 0\},$$

for all $x \in X$. The constraint describing this set is a *stability constraint*; it *can be violated* in a physical system, but if it is violated, *the system will no longer meet the control design specifications*, i.e. exponential stability at a rate $0 < \varepsilon < 1$.

Challenges arise when there is disagreement between stability constraints and input constraints, i.e. when the physical system cannot produce a stabilizing input:

$$U(x) \cap K(x) = \{\emptyset\}. \tag{4}$$

In this situation, the input constraint will always dominate; the system will continue to evolve according to the dynamics (1), but the behavior will not be stable in the originally intended sense. The control engineer is presented with a challenge: design (directly or indirectly) a new control Lyapunov function that satisfies both

$$C_u(x,u) \leq 0 \tag{5}$$
$$L_f V(x) + L_g V(x)u + \frac{c_3}{\varepsilon}V(x) \leq 0 \tag{6}$$

over the operating range, or *domain*, of the system.

Intuitively, if the control objective is aggressive and the admissible control set is conservative, the intersection of the two sets U and $K_\varepsilon(x)$ will be empty. In this paper we set out to construct a control method that avoids situations in which the intersection of these two sets is empty, or in other words, a control method that is always *locally feasible*.

Definition 1: For the system (1)*, a RES-CLF, $V_\varepsilon(x)$, designed with $U = \mathbb{R}^m$, is said to be* **locally feasible** *for the same system with $U(x) = \{u \in \mathbb{R}^m : C_u(x,u) \leq 0\}$, at a point $x_0 \in X$, if there exists a $u \in \mathbb{R}^m$ such that*

$$C_u(x_0,u) \leq 0, \tag{7}$$
$$L_f V_\varepsilon(x_0) + L_g V_\varepsilon(x_0)u + \frac{c_3}{\varepsilon}V(x_0) \leq 0, \tag{8}$$

and locally infeasible otherwise.

In this paper, we will consider constrained control systems with input constraints of the (affine) form

$$C_u(x,u) = A_u(x)u - b_u(x) \tag{9}$$

where $A_u(x)$ and $b_u(x)$ are mathematical representations of the system's input limits. Constraints of this form are used to model several constraints in robot locomotion, e.g. actuator power limitations and zero-moment point constraints.

The local feasibility (or infeasibility) of a CLF at a point $x_0 \in X$, in the sense of Definition 1 with input constraints of the form (9), can be established by determining whether a solution u exists to the following set of inequalities

$$\underbrace{\begin{bmatrix} L_g V_\varepsilon(x_0) \\ A_u(x_0) \end{bmatrix}}_{A(x_0)} u - \underbrace{\begin{bmatrix} -L_f V_\varepsilon(x_0) - \frac{c_3}{\varepsilon}V(x_0) \\ b_u(x_0) \end{bmatrix}}_{b(x_0)} \leq 0. \tag{10}$$

A concrete metric for establishing the feasibility of a set of linear inequalities comes from the following metric, the width of the feasible set, as defined by [16].

Definition 2: For the system of inequalities (10)*, the* **width of the feasible set** *is the unique solution to the following Linear Program (LP):*

$$\begin{aligned} \omega(x_0) = &\max_{(u,w)\in\mathbb{R}^{m+1}} w \\ &\text{s.t.} \begin{bmatrix} A(x_0) & \mathbf{1} \\ \mathbf{0} & -1 \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix} \leq \begin{bmatrix} b(x_0) \\ 0 \end{bmatrix} \end{aligned} \tag{11}$$

where A and b are defined in (10)*.*

Thus, for controllers, utilizing an unconstrained CLF applied to a control system under affine constraints of the form (9), such as the following Quadratic Program:

$$\begin{aligned} u^* = &\operatorname*{argmin}_{\mu\in\mathbb{R}^m} u^T u \\ &\text{s.t.} \quad A(x_0)u - b(x_0) < 0, \end{aligned} \tag{12}$$

if we can show that the width of the corresponding feasible set is greater than zero, we can *guarantee* that the unconstrained CLF can be used to achieve local exponential stability in the constrained system. In the following section, we give an example of how to design a parameterized CLF which explicitly ensures that the width of the feasible set is always greater than zero and thus (12) is always solvable.

III. Feasible Control Parameter Modification

In this section, we provide an example of one way of using Definition 2 to design feasible control systems. The method will entail modification of parameters α of desired reference trajectories in input-output control systems of the form

$$\dot{x} = f(x) + g(x)u, \tag{13}$$

$$y = y^a(x) - y^d(t, \alpha), \tag{14}$$

where $x \in X \subseteq \mathbb{R}^n$ and $u \in U \subseteq \mathbb{R}^m$, and $y^a : X \to \mathbb{R}^m$ and $y^d : \mathbb{R} \times \mathbb{R}^a \to \mathbb{R}^m$ are smooth functions encoding the desired behavior to be realized via control. For the following analysis, assume that the system (13) is feedback linearizable, with corresponding control law

$$u = A_{io}^{-1}(-b_{io} + (\mu + \ddot{y}^d)). \tag{15}$$

where $A_{io} = L_g L_f y$ and $b_{io} = L_f^2$ are Lie derivatives of the (relative degree two) outputs y along $f(x)$ and $g(x)$. Applying (15) to (13) and defining coordinates $\eta = (y^T, \dot{y}^T)^T$ results in a local coordinate transformation $\Phi : x \mapsto \eta$, with corresponding linear dynamics $\dot{\eta} = F\eta + G\mu$.

A. Local CLF-Based Control

As the control objective is to drive $y^a(x) \to y^d(t, \alpha)$ and equivalently $\eta \to 0$, the method of [1] is used to construct a rapidly exponentially stabilizing control Lyapunov function (RES-CLF) denoted $V(\eta) = \eta^T P_\varepsilon \eta$, where P_ε is obtained from (47) in [1] (a modified continuous-time algebraic Riccati equation). The time derivative of V is

$$\dot{V}(\eta, \mu) = \eta^T (F^T P_\varepsilon + P_\varepsilon F)\eta + 2\eta^T P_\varepsilon G\mu. \tag{16}$$

The stabilizing constraint (8) can thus be expressed as:

$$\underbrace{2\eta^T P_\varepsilon G}_{A_{CLF}(\eta)}\mu \leq \underbrace{-\eta^T M \eta}_{b_{CLF}(\eta)}, \tag{17}$$

where $M := F^T P_\varepsilon + P_\varepsilon F + \frac{c_3}{\varepsilon} P_\varepsilon$ is used to simplify the expression. As (17) is affine in μ, it can be incorporated as a constraint in a μ-based QP of the form (12)

B. Constraints

In input-output systems of the form (13), constraints are placed on u, e.g. in the control of robot locomotion, the constraints (9) come in the form of torque bounds and ZMP constraints. These bounds can be mapped (locally) into μ bounds using the feedback linearization control law (15)

$$\underbrace{A_u A_{io}^{-1}}_{A_\mu}(\mu + \ddot{y}^d) - \underbrace{(A_u A_{io}^{-1} b_{io} + b_u)}_{b_\mu} \leq 0, \tag{18}$$

and included in a QP-based controller of the form (12)

$$\mu^* = \underset{\mu \in \mathbb{R}^m}{\operatorname{argmin}} \ \mu^T \mu \tag{19}$$

$$\begin{bmatrix} A_{CLF} \\ A_\mu \end{bmatrix} \mu - \begin{bmatrix} b_{CLF} \\ b_\mu - A_\mu \ddot{y}^d \end{bmatrix} < 0.$$

As the input constraints are properties of the system, we will have to modify the control objectives to resolve feasibility of (19). In the following section, we propose a method of modifying the parameters α of the desired outputs corresponding to a minimal (local) change in the RES-CLF.

C. A Modified Control Lyapunov Function

To ensure that the QP-based controller (19) is feasible, we propose the following modification to the RES-CLF:

$$V^*(\eta) = (\eta - \eta^*)^T P_\varepsilon (\eta - \eta^*), \tag{20}$$

in which the zero level set of the Lyapunov function has been shifted from the origin to $\eta^* \in \mathbb{R}^n$. The corresponding CLF constraint is quadratic (bilinear) in η^* and μ

$$2(\eta - \eta^*)^T P_\varepsilon G\mu + (\eta - \eta^*)^T M(\eta - \eta^*) \leq 0. \tag{21}$$

To remove dependence on the control, we will make an explicit choice of the input

$$\mu = -K(\eta - \eta^*), \tag{22}$$

where K is obtained by solving the LQR problem for the system (F, G) with cost function $J(\eta, \eta^*, \mu) = (\eta - \eta^*)^T P_\varepsilon (\eta - \eta^*) + \mu^T R \mu$, with $R > 0$. This results in a quadratic constraint on η^*.

When the parameters α appear linearly in y^d and $\dot{y}^d$, the shift in outputs $\eta^* = (y^*, \dot{y}^*)$ can be written

$$\eta^* = \tau(t)(\alpha^* - \alpha). \tag{23}$$

where $\tau(t) = \frac{\partial \eta}{\partial \alpha}$. Substituting (23) and (22) into (21) results in a quadratic constraint on the updated parameters α^*. This constraint can be combined with the input constraints (18) and the definition of the width of the feasible set (11) to construct a quadratically constrained quadratic program with decision variables α^* as follows

---

Feasible Parameter Optimization

$$\mathcal{P}(t, x, \alpha, \omega^*) = \underset{(\alpha^*, w)}{\operatorname{argmin}} \ \alpha^{*T} Q \alpha^* + f^T \alpha^* + c \tag{24}$$

$$\text{s.t.}$$

$$\alpha^{T*} H \alpha^* + k^T \alpha^* + d + w \leq 0,$$

$$A\alpha^* + b + w \leq 0,$$

$$-w + \omega^* \leq 0.$$

The cost function, $\eta^{*T}\eta^* = (\alpha^* - \alpha)^T \tau(t)^T \tau(t)(\alpha^* - \alpha)$, is represented using the following quantities:

$$Q = \tau^T \tau, \quad f^T = -2\alpha^T \tau^T \tau, \quad c = \alpha^T \tau^T \tau, \alpha$$

and the constraints are represented by

$$\begin{aligned} H &= \tau^T (M - 2P_\varepsilon GK)\tau, \\ k^T &= -2(\eta + \tau\alpha)^T (M - 2P_\varepsilon GK)\tau, \\ d &= (\eta + \tau\alpha)^T (M - 2P_\varepsilon GK)(\eta + \tau\alpha), \\ A &= A_\mu (K\tau + \ddot{\kappa}), \\ b &= -A_\mu K(\eta + \tau\alpha) - b_\mu. \end{aligned}$$

These quantities are computed with $\eta = \eta(t, x, \alpha)$.

This program solves for new parameters α^* corresponding to minimal changes in the desired trajectories y^d subject to the corresponding CLF being feasible. In the following section we show how to use this QCQP to in a hybrid system which updates parameters as needed, i.e. when a CLF computed with unmodified parameters becomes infeasible.

IV. Hybrid Feasibility Controller

In this section we present a novel control scheme that updates its parameters in the event of a significant drop in feasibility. Specifically, we propose a hybrid control system consisting of: (1) a continuous-time controller that implements a control Lyapunov function subject to input constraints and (2) a discrete-time component which updates desired trajectories in the event that the CLF computed with "old" parameters becomes infeasible. Here a hybrid control system, $\mathscr{HC}$, is defined as a tuple

$$\mathscr{HC} = (\mathcal{D}, S, \Delta, f, g, U), \tag{25}$$

where

- $\mathcal{D}$ is the *domain* with $\mathcal{D} \subseteq \mathcal{X}$ a smooth submanifold of the state space $\mathcal{X} \subseteq \mathbb{R}^{n+n_\alpha}$,
- $S \subset \mathcal{D}$ is a proper subset of $\mathcal{D}$ called the *guard* or *switching surface*,
- $\Delta : S \to \mathcal{D}$ is a smooth map called the *reset map*,
- (f, g) is a *control system* on $\mathcal{D}$,
- $U \subseteq \mathbb{R}^m$ is the set of admissible control.

The admissible control takes the form (9), stated again for reference:

$$U(x) = \{u \in \mathbb{R}^m : A_u(x)u - b_u(x) \leq 0\}. \tag{26}$$

The coordinates for this system is a vector of state variables together with a vector of controller coefficients $q = (x, \alpha)$; thus, the control system is

$$\dot{x} = f(x) + g(x)u \tag{27}$$

$$\dot{\alpha} = 0 \tag{28}$$

The controller u is input-output linearization (15) with $\mu = \mu^*$ obtained through the solution of (19). The following hybrid elements of the controller ensure that the inequality constraints in (19) are always feasible through use of the proposed parameter update law.

Here, the domain is characterized by the width of the feasible set (11) for the constraints in (19) being above a threshold ω^* and the guard is hit when the width of the feasible set reaches the threshold, i.e.

$$\mathcal{D}(t) = \{x \in \mathbb{R}^n, \alpha \in \mathbb{R}^{n_\alpha} : \omega(t, x, \alpha) - \omega^* \geq 0\}, \tag{29}$$

$$S(t) = \{x \in \mathbb{R}^n, \alpha \in \mathbb{R}^{n_\alpha} : \omega(t, x, \alpha) - \omega^* = 0\}. \tag{30}$$

The restriction of the reset map to the state variables is simply the identity function, i.e. $x^+ = \Delta_x(x^-) = x^-$. However, the restriction of the reset map to the controller parameters is where feasibility is resolved. Here, we solve for a new set of parameters $\alpha = \Delta_\alpha(\eta, t, \alpha, \omega^*)$, using the parameter modification QCQP (24)

$$\Delta_\alpha(t, x, \alpha, \omega^*) = \mathcal{P}(t, x, \alpha, \omega^*). \tag{31}$$

The idea is to empower the robot control system to handle infeasibilities "on-the-fly", rather than having to synthesize a library of feasible parameters for every conceivable evolution of the system. In the following section, the proposed hybrid feasibility controller is applied to the control of constrained bipedal robotic locomotion.

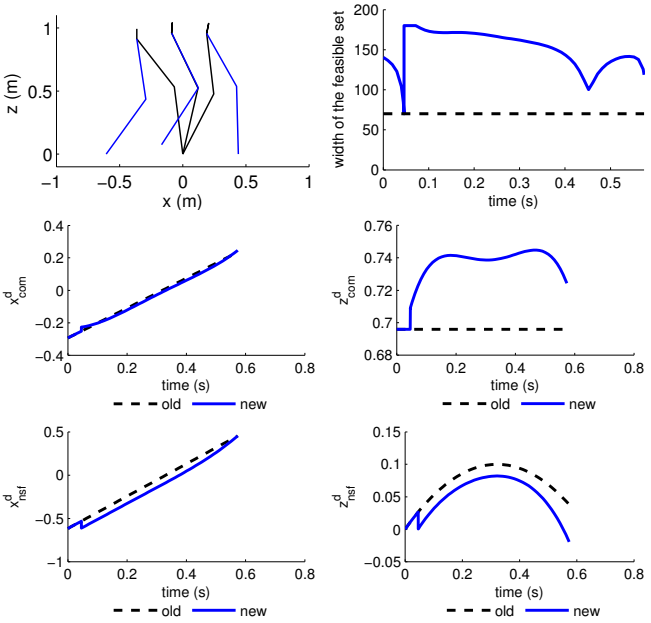

Fig. 1. Results from simulation of the proposed method applied to a planar, 5-DOF walking robot. Snapshots of the robot from one continuous-time step of walking are shown in the top left, and the corresponding width of the feasible set is shown in the top right. The original desired trajectories (dashed, black) are shown together with modified trajectories (solid, blue) in the middle and bottom rows.

V. Application: Robot Walking

This section presents the results from simulation of the proposed hybrid feasibility controller applied to two bipedal robotic walking platforms. The first system is a fully-actuated, planar biped with point feet and five independently actuated joints. The second system considered is a ten DOF, 3D biped walking with both torque and ZMP constraints. Simulation results for the 3D biped show how catastrophic events (events in which the ZMP constraints vanish) can be avoided via the proposed hybrid feasibility controller.

A. Walking Dynamics and Control System

The standard robot dynamics (see [17], [23]) are

$$D(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = u, \tag{32}$$

with joint angles and velocities, $(\theta, \dot{\theta})$, inertia matrix $D(\theta)$, Coriolis matrix $C(\theta, \dot{\theta})$, gravity vector $G(\theta)$, and actuator torque vector $u \in \mathbb{R}^m$. These dynamics can be used to express a general nonlinear control system for the robot:

$$\dot{x} = f(x) + g(x)u, \tag{33}$$

in which $x = (\theta, \dot{\theta})^T \in \mathbb{R}^n$. In light of the current paper, we consider *constrained control systems* with constraints of the form (9), stated again for reference:

$$U(x) = \{u : A_u(x)u - b_u(x) \leq 0\}.$$

We first consider the case when the input (torque) constraints $A_u(x)$ and $b_u(x)$ are simply constant torque bounds, and then we consider both torque bounds and zero moment point (ZMP) constraints.

B. Walking Control Objectives

For the sagittal outputs in both the 2D and the 3D robots, we've chosen to control quantities which are common in the control of robot locomotion: the x-z coordinates of the center of mass (2 outputs), the x-z coordinates of the swing foot (2 outputs), and the torso angle with respect to the vertical axis. The actual quantities are

$$y^a(\theta) = \begin{bmatrix} x_{com}(\theta) \\ z_{com}(\theta) \\ x_{nsf}(\theta) \\ z_{nsf}(\theta) \\ \theta_{tor}(\theta) \end{bmatrix}. \tag{34}$$

To show the generality of the proposed method, and for quick controller development, we've chosen 4th-order polynomials to describe the *desired walking trajectories*, $y^d(t, \alpha)$

$$y_i^d(t, \alpha) := \sum_{j=0}^{4} \alpha_{i,j} t^j, \quad i \in 1, 2, \ldots, 5. \tag{35}$$

For a nominal α, these reference trajectories encode the following goals: move the horizontal components of the center of mass and the nonstance foot forward at constant velocity, regulate the vertical center of mass and the torso angles to constant values and follow a parabolic path for the vertical position of the nonstance foot.

C. Robot Locomotion Case 1: a 2D Robot Walking with Torque Bounds.

In the first case study, we consider a planar, fully actuated point-foot robot walking with torque bounds. The control objectives described in Section V-B are implemented via a control Lyapunov function of the form (20). The physical constraint set of interest for this example is

$$A_u = \begin{bmatrix} I \\ -I \end{bmatrix}, \quad b_u = \begin{bmatrix} u^{\max} \\ u^{\max} \end{bmatrix}.$$

These physical constraints are mapped into μ space using (18) and included with the CLF constraints to form the total constraint set:

$$\begin{bmatrix} A_{CLF} \\ A_\mu \end{bmatrix} \mu - \begin{bmatrix} b_{CLF} \\ b_\mu - A_\mu \ddot{y}^d(t, \alpha) \end{bmatrix} < 0.$$

In the simulation, one continuous-time phase of walking (impact-to-impact) is considered. The proposed hybrid feasibility controller (25) is employed to ensure that the robot completes an entire step despite drops in feasibility. The feasible set threshold is chosen (heuristically) to be $\omega^* = 70$.

Fig. 1 shows results from simulation of this case study. In the top left subplot, a depiction of the robot is shown at the beginning, middle and end of the continuous-time step. A plot of the width of the feasible set versus time shows that the guard, $\omega = \omega^*$, is hit very near 0.055 seconds. This prompts the hybrid feasibility controller to produce new, feasible parameters via the reset map (31) which uses the proposed feasible parameter optimization (24). The resulting modified trajectories are shown against the originals in Fig. 1. Note that in this case, the optimization produces horizontal center of mass and swing foot trajectories which are similar in shape to the originals, but it allows the vertical center of mass trajectory to drift 4 cm from its original desired value.

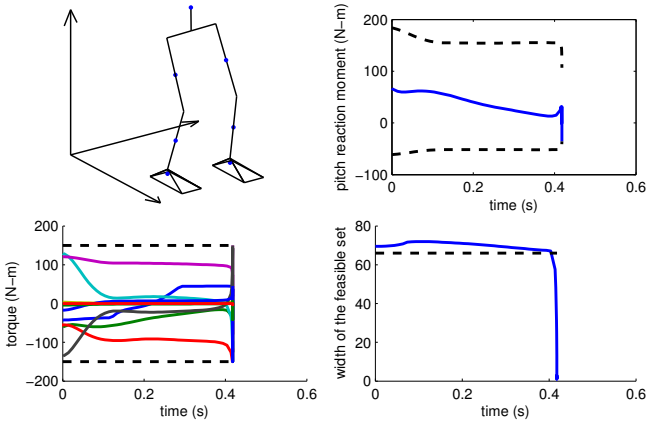

Fig. 2. Simulation results of the 3D robot walking with ZMP and torque bounds. In this simulation no modification is made to the desired trajectories. Note that very near 0.42 seconds, the width of the feasible set plummets, the torques chatter, the ZMP constraints vanish and the simulation stops. We would expect the robot to fall (a catastrophic event).

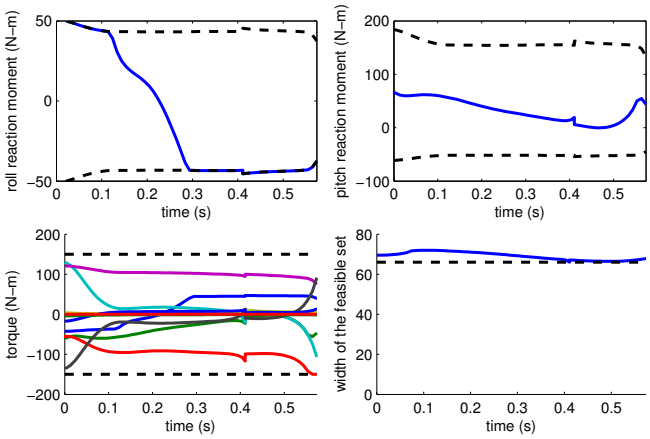

Fig. 3. Simulation results of the 3D robot walking with ZMP and torque bounds. In this simulation, the parameters α for the desired trajectories over $I_T = [0, 0.42]$ seconds are the same as those used in 3D simulation corresponding to Fig. 2. Note, in the current simulation, that very near 0.42s the guard of the hybrid feasibility controller is met $\omega = \omega^*$, and a new set of feasible parameters α are resolved via the proposed feasible control design method. The robot continues to take a step, (it does not fall), and the catastrophic event is successfully avoided.

D. Robot Locomotion Case 2: a 3D Robot Walking with Torque and ZMP Bounds.

In the second case study, we consider a more complex example: a 10-DOF robot walking in 3D under torque and ZMP constraints. As in the planar case, the control objectives described in Section V-B – together with objectives which minimize lateral movement and keep the swing foot parallel to the ground – are implemented via a control Lyapunov function of the form (20). The ZMP constraints, as in [5], are linear inequalities in u, thus, we can write them as $A_{ZMP}(x)u - b_{ZMP}(x) \leq 0$, and include them with torque

bounds, yielding the constraint set:

$$A_u = \begin{bmatrix} I \\ -I \\ A_{ZMP}(x) \end{bmatrix}, \quad b_u = \begin{bmatrix} u^{\max} \\ u^{\max} \\ b_{ZMP}(x) \end{bmatrix}.$$

In this simulation study, as in the first, one continuous-time phase of walking (impact-to-impact) is considered. Maintaining feasibility is a greater challenge in this case study due to the inclusion of ZMP constraints; without explicit update of the parameters, and without some prior optimization of the parameters, ZMP is unlikely to be satisfied. Fig. 2 shows the results of simulation of a local QP controller without the proposed feasible parameter update, wherein the width of the feasible set drops to zero very near 0.42 seconds, the controller torques chatter, the ZMP constraints vanish and the numeric integration solver stops. As ZMP constraints determine whether the robot is balanced, we would expect the robot to fall in this case.

Fig. 3 show results from simulation of the robot under the same conditions as the simulation shown in Fig. 2, except now the proposed hybrid feasibility controller is employed. The feasible set threshold is chosen (heuristically) to be $\omega^* = 66$. Note that in this case the guard, $\omega = \omega^*$, is met milliseconds before the time the catastrophic event occurs, the proposed feasible parameter optimization (24) is used to solve for new (feasible) parameters which are then used in the local CLF controller and the robot is able to successfully complete a continuous-time step.

VI. Conclusion and Future Work

In this paper, we presented a novel method for updating control parameters to overcome infeasibilities that arise due to the conflict between aggressive control and physical input limitations. The method is a hybrid control system which continually evaluates the local feasibility of the current controller and updates parameters as needed. Parameter modifications are performed through a novel quadratically constrained quadratic program. Initial simulation results of the proposed method show promising performance and motivate further development of the method.

The parameter optimization is a quadratically constrained quadratic program; however, actual implementation of the method uses MATLAB's `fmincon` function for solving general nonlinear optimization problems. Although the gradient and Hessian of the problem can be supplied to `fmincon` to increase convergence speed, future work will be to further take advantage of the structure of the QCQP and use more efficient methods to bring it to real-time computation.

It is important to note that the proposed method addresses feasibility of *continuous-time* control of (fully actuated) robot locomotion. However, robot locomotion is *hybrid* by nature: collisions between the robot and the ground destabilize walking if not accounted for. Furthermore, locomotion often involves phases of under-actuation wherein the evolution of the system includes uncontrollable dynamics. To fully ensure feasibility of underactuated robotic walking, one must address hybrid system constraints, as in [6], in addition to continuous-time control and physical constraints; existing methods to solving this class of problems use (offline) nonlinear optimization. The goal of future work is to develop a method of including hybrid system constraints in the proposed continuous-time feasibility QCQP of this paper to achieve feasible, underactuated robot walking in real-time.

References

[1] A. D. Ames, K. Galloway, J. W. Grizzle, and K. Sreenath. Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics. *Automatic Control, IEEE Transactions on*, 2014.

[2] S. de Oliveira Kothare and M. Morari. Contractive model predictive control for constrained nonlinear systems. *Automatic Control, IEEE Transactions on*, 2000.

[3] K. S. Galloway, K. Sreenath, A. D. Ames, and J. W. Grizzle. Torque saturation in bipedal robotic walking through control lyapunov function based quadratic programs. *CoRR*, abs/1302.7314, 2013.

[4] E. Gilbert, I. Kolmanovsky, and K. T. Tan. Nonlinear control of discrete-time linear systems with state and control constraints: a reference governor with global convergence properties. In *IEEE Conference on Decision and Control*, 1994.

[5] J. W. Grizzle, C. Chevallereau, A. D. Ames, and R. W. Sinnet. 3D bipedal robotic walking: models, feedback control, and open problems. In *Proceedings of the 8th International Federation of Automatic Control Symposium on Nonlinear Control Systems*, pages 505–532, Bologna, Italy, Sept. 2010.

[6] J. W. Grizzle and E. R. Westervelt. Hybrid zero dynamics of planar bipedal walking. In *Analysis and Design of Nonlinear Control Systems*. 2008.

[7] T. Hu and Z. Lin. Composite quadratic lyapunov functions for constrained control systems. *Automatic Control, IEEE Transactions on*, 48(3):440–450, Mar 2003.

[8] A. Isidori. *Nonlinear Control Systems*. Springer, London, 1995.

[9] P. Kapasouris, M. Athans, and G. Stein. Design of feedback control systems for stable plants with saturating actuators. In *Decision and Control, 1988., Proceedings of the 27th IEEE Conference on*, pages 469–479 vol.1, Dec 1988.

[10] H. Khalil. *Nonlinear Systems*. Prentice Hall PTR, 2002.

[11] S. Kuindersma, F. Permenter, and R. Tedrake. An efficiently solvable quadratic program for stabilizing dynamic locomotion. *CoRR*, 2013.

[12] M. J. Kurtz and M. A. Henson. Input-output linearizing control of constrained nonlinear processes. *Journal of Process Control*, 1997.

[13] M. Lazar. Flexible control lyapunov functions. In *American Control Conference, 2009. ACC '09.*, pages 102–107, June 2009.

[14] M. Mahmood and P. Mhaskar. On constructing constrained control lyapunov functions for linear systems. In *American Control Conference (ACC), 2010*, pages 5191–5196, June 2010.

[15] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Survey constrained model predictive control: Stability and optimality. *Automatica*, 2000.

[16] B. Morris, M. Powell, and A. Ames. Sufficient conditions for the lipschitz continuity of qp-based multi-objective control of humanoid robots. In *IEEE Conference on Decision and Control*, 2013.

[17] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, 1994.

[18] M. Posa, C. Cantu, and R. Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *Int. J. Rob. Res.*, 33(1):69–81, Jan. 2014.

[19] M. Powell, A. Hereid, and A. Ames. Speed regulation in 3d robotic walking through motion transitions between human-inspired partial hybrid zero dynamics. In *IEEE ICRA*, 2013.

[20] L. Praly, D. Carnevale, and A. Astolfi. Dynamic versus static weighting of lyapunov functions. *IEEE Transactions on Automatic Control*, 58(6):1557–1561, June 2013.

[21] P. Reist and R. Tedrake. Simulation-based lqr-trees with input and state constraints. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 5504–5510, May 2010.

[22] S. S. Sastry. *Nonlinear Systems: Analysis, Stability and Control*. Springer, 1999.

[23] M. W. Spong and M. Vidyasagar. *Robotic Dynamics and Control*. John Wiley and Sons, New York, NY, 1989.

[24] A. R. Teel. Global stabilization and restricted tracking for multiple integrators with bounded controls. *Systems and Control Letters*, 1992.