An Inverse Dynamics Approach to Control Lyapunov Functions

Jenna Reher, Claudia Kann, and Aaron D. Ames[1]

Abstract— With the goal of moving towards implementation of increasingly dynamic behaviors on underactuated systems, this paper presents an optimization-based approach for solving full-body dynamics based controllers on underactuated bipedal robots. The primary focus of this paper is on the development of an alternative approach to the implementation of controllers utilizing control Lyapunov function based quadratic programs. This approach utilizes many of the desirable aspects from successful inverse dynamics based controllers in the literature, while also incorporating a variant of control Lyapunov functions that renders better convergence in the context of tracking outputs. The principal benefits of this formulation include a greater ability to add costs which regulate the resulting behavior of the robot, in addition, the model error-prone inertia matrix is used only once, in a non-inverted form. The result is a successful demonstration of the controller for walking in simulation, and applied on hardware in real-time for crouching.

I. INTRODUCTION

Model based control methods can help enable dynamic and compliant motion of robots while achieving remarkable control accuracy. However, implementing such techniques on floating base robots is non-trivial due to model inaccuracy, underactuation, dynamically changing contact constraints, and possibly conflicting objectives for the robot [2], [6]. Unlike their classical counterparts, optimization based approaches to handling these control problems allow for the inclusion of physical constraints that the system is subjected to [8], [25]. Because of this feature, quadratic programming (QP) based controllers have been increasingly used to stabilize real-world systems on complex robotic platforms without the need to algebraically produce a control law or enforce convergence guarantees [10], [14], [18].

These examples, however, typically do not consider periodic notions of stability for highly underactuated systems; these systems often require additional convergence guarantees in order to realize stability. It was shown in [5] that through the use of a *rapidly exponentially stable control Lyapunov function* (RES-CLF), coupled with *hybrid zero dynamics* (HZD) [12], [31], a wide class of controllers can be designed to create rapidly exponentially convergent hybrid periodic orbits. It was also shown that this could be posed as a QP, in which the convergence is enforced via an inequality constraint; forming a control Lyapunov function based quadratic program (CLF-QP) [5] [6]. Often, the desires of a control engineer are in conflict with the capabilities of the robot, as the system cannot produce

Fig. 1. The Cassie biped, built by Agility Robotics, and used as an experimental platform to demonstrate the controllers presented in this work.

sufficient convergence without violating physical constraints. One approach to address this is to relax convergence guarantees, which allows (local) drift in the control objectives to accommodate feasibility. These formulations have since been used to achieve dynamic locomotion on robotic systems both in simulation [13], [17], [24], [32] and on hardware [11].

While high level task controllers based on inverse dynamics approaches pose similar problems as CLF-QPs, they have traditionally not been formulated in the same way. In implementations of CLF-QPs the vector fields associated with robotic systems are typically utilized, which involves costly computations; while in task based controllers, the dynamics are an equality constraint, and objectives are driven towards their targets through PD controllers in the cost [10]. There have been several connections shown in related research [19], [20], where control Lyapunov functions were included in an inverse dynamics controller via an LQR in the cost. In this work we aim to repurpose several of the more mature concepts from inverse dynamics based approaches and demonstrate a more efficient CLF inspired formulation.

The main result of this paper is an optimization-based control framework that couples convergence constraints from CLF-QPs with concepts from inverse dynamics based controllers. We begin in Section II by the CLF framework which yields rapid convergence for dynamic motions on underactuated systems. This is followed by Section III which explores existing optimization based techniques for control. Section IV details a new class of optimization based controllers based on the CLF construction. Section V discusses how to apply these methods practically. In Section VI the model for the bipedal robot Cassie is shown and the new controller is demonstrated in both simulation for walking, and in real-time on hardware for standing and crouching.

II. Preliminaries on Control Lyapunov Functions

In classical nonlinear control design, analysis is typically performed on a dynamical system of the form:

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

where $x \in X \subseteq \mathbb{R}^n$ is the set of controllable states and $u \in U \subseteq \mathbb{R}^m$ is the control input. The mappings $f : \mathbb{R}^n \to \mathbb{R}^n$, $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are assumed to be locally Lipschitz continuous functions of x. Let us consider a feedback control system which tracks a set of desired trajectories of the form:

$$y(x) = y^a(x) - y^d(\tau(x)), \tag{2}$$

where $y^a : X \to \mathbb{R}^m$ 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. We assume that $y(x)$ has (vector) relative degree r [27]. It is often the case in robotic systems that $r = 1$ if $y(x)$ depends on position and velocity and $r = 2$ if it only depends on position, i.e., configuration variables. Taking the derivatives of the outputs along $f(x)$ and $g(x)$ we obtain,

$$y^{(r)}(x) = L_f^{(r)} y(x) + \underbrace{L_g L_f^{(r-1)} y(x)}_{A} u, \tag{3}$$

where A is called the *decoupling matrix* which is invertible in the case of a (vector) relative degree. This implies that the system (1) is feedback linearizable, and we can then prescribe the following control law:

$$u(x) = A^{-1}\left(- L_f^{(r)} y(x) + v\right) \Rightarrow y^{(r)} = v, \tag{4}$$

where v is an auxiliary feedback control value.

To motivate later constructions, we consider a mechanical system with configuration space $\mathcal{Q}$, (local) coordinates $q \in \mathcal{Q}$, and states $x = (q^T, \dot{q}^T)^T \in T\mathcal{Q} = X$. Suppose that for (1) there is a set of outputs $y(x) = (y_1(q, \dot{q})^T, y_2(q)^T)^T$ of vector relative degree 1 and 2, respectively, on a region of interest; that is for $y_1(q, \dot{q}) \in \mathbb{R}^{m_1}$ and $y_2(q) \in \mathbb{R}^{m_2}$ with $m = m_1 + m_2$ we assume the vector relative degree is 1 for y_1 and the 2 for y_2, i.e., $(1, \ldots, 1, 2, \ldots, 2)$ with 1's appearing m_1 times and 2's appearing m_2 times. We can then write an output tracking problem:

$$y_1(q, \dot{q}, t) = y_1^a(q, \dot{q}) - y_1^d(\tau(t, q)) \tag{5}$$
$$y_2(q, t) = y_2^a(q) - y_2^d(\tau(t, q)), \tag{6}$$

where y^a and y^d are the actual and desired outputs, and $\tau(t, q)$ is some parameterization of time for the desired outputs to evolve on. Assuming that the preliminary feedback (4) has been applied to (1), we will render a linear system for the output dynamics with coordinates $\eta := (y_1^T, y_2^T, \dot{y}_2^T)^T$,

$$\dot{\eta} = \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \ddot{y}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & \mathbf{I} \\ 0 & 0 \end{bmatrix}}_{F} \eta + \underbrace{\begin{bmatrix} \mathbf{I} & 0 \\ 0 & 0 \\ 0 & \mathbf{I} \end{bmatrix}}_{G} v. \tag{7}$$

A valid choice of v which stabilizes this system is:

$$v = \begin{bmatrix} \dot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{\epsilon} K_{\bar{v}} y_1 \\ -\frac{1}{\epsilon^2} K_P y_2 - \frac{1}{\epsilon} K_D \dot{y}_2 \end{bmatrix}, \tag{8}$$

where $0 < \epsilon \leq 1$ is a tunable parameter, and $K_{\bar{v}}$, K_P, K_D are control gains for the relative degree 1 and relative degree 2 output errors, respectively. While this controller yields convergence to the target outputs, it does not leverage the natural dynamics of the system, and disregards torque and feasibility constraints by which the system must abide. Thus, for practical systems, additional considerations for selecting our control input are often required.

The *exponentially stabilizing control Lyapunov function* [4] (ES-CLF) and *rapidly exponentially stabilizing control Lyapunov function* (RES-CLF) frameworks [5] were introduced as methods for achieving stability in the output dynamics. In the context of the control system (7), we consider the continuous time algebraic Riccati equations (CARE):

$$F^T P + PF - PGG^T P + Q = 0, \tag{9}$$

for $Q = Q^T > 0$ and with solution $P = P^T > 0$. The method presented in [5] can then be employed to construct a (R)ES-CLF,

$$V(\eta) = \eta^T \underbrace{\mathbf{I}_\epsilon P \mathbf{I}_\epsilon}_{P_\epsilon} \eta, \quad \text{with } \mathbf{I}_\epsilon := \operatorname{diag}\left(\mathbf{I}, \frac{1}{\epsilon}\mathbf{I}, \mathbf{I}\right), \tag{10}$$

where the selection of $0 < \epsilon < 1$ creates a RES-CLF, and $\epsilon = 1$ instead renders an ES-CLF. We can find the derivative of (10) to be:

$$\dot{V}(\eta) = L_F V(\eta) + L_G V(\eta)v, \tag{11}$$

where the Lie derivatives of V_ϵ along the linear output system's dynamics (7) are

$$L_F V(\eta) = \eta^T (F^T P_\epsilon + P_\epsilon F)\eta, \tag{12}$$
$$L_G V(\eta) = 2\eta^T P_\epsilon G. \tag{13}$$

An exponential convergence constraint can then be prescribed as,

$$L_F V(\eta) + L_G V(\eta)v \leq -\underbrace{\frac{\lambda_{\min}(Q)}{\epsilon \lambda_{\max}(P_\epsilon)}}_{\gamma} V(x), \tag{14}$$

where γ is related to the convergence rate. This constraint is in terms of our auxiliary control input v and not the actual feedback control u. In order to convert back into a form which can be represented in terms of the control input, we can use the previous relationship between u and v

$$A(x)u + L_f^{(r)} y(x) = v, \tag{15}$$

to obtain the CLF constraint stated in terms of x since η depends on x (via y_1, y_2 and $\dot{y}_2$):

$$\underbrace{L_F V(x) + L_G V(x)L_f^{(r)} y(x)}_{L_f V(x)} + \underbrace{L_G V(x)A(x)}_{L_g V(x)} u \leq -\gamma V(x). \tag{16}$$

In the context of (R)ES-CLF, we can then define the set

$$K(x) = \{u \in U : L_f V(x) + L_g V(x)u + \gamma V(x) \leq 0\}, \tag{17}$$

consisting of the control values which result in (rapidly) exponential convergence, wherein $\dot{V}(\eta(x)) \leq -\gamma V(\eta(x))$.

III. Optimization Based Controllers

The dynamics of robotic systems can be formulated using the method of Lagrange, with positional constraints on the system incorporated via D'Alembert's principle [22],

$$D(q)\ddot{q} + H(q, \dot{q}) = Bu + J^T(q)\lambda \qquad (18)$$

$$J(q)\ddot{q} + \dot{J}(q, \dot{q})\dot{q} = 0, \qquad (19)$$

where $D(q)$ is the inertia matrix, $H(q, \dot{q}) = C(q, \dot{q})\dot{q} + G(q) + F$ is the vector sum for the Coriolis, centripital, gravitational forces, and additional non-conservative generalized forces, B is the actuation matrix, and the Jacobian of the holonomic constraints is $J(q) = \partial h / \partial q$ with its corresponding constraint wrenches $\lambda \in \mathbb{R}^{m_h}$. This can be converted to an ODE in the form of (1) as:

$$f(x) = \begin{bmatrix} \dot{q} \\ -D^{-1}(q)\left(J^T(q)\lambda - H(q, \dot{q})\right) \end{bmatrix},$$

$$g(x) = \begin{bmatrix} 0 \\ D(q)^{-1}B \end{bmatrix}. \qquad (20)$$

We begin the derivation of our controller for this system by considering again the outputs (5) and (6), and taking the necessary derivatives of the outputs:

$$\begin{bmatrix} \dot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\partial y_1}{\partial q} \\ \frac{\partial}{\partial q}\left(\frac{\partial y_2}{\partial q}\dot{q}\right) \end{bmatrix}}_{\dot{J}_y} \dot{q} + \underbrace{\begin{bmatrix} \frac{\partial y_1}{\partial \dot{q}} \\ \frac{\partial y_2}{\partial q} \end{bmatrix}}_{J_y} \ddot{q}. \qquad (21)$$

Or alternatively, by taking the derivatives along the vector fields (20), where the dependencies have been dropped for the sake of clarity, we can write:

$$\begin{bmatrix} \dot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\partial y_1}{\partial q} & \frac{\partial y_1}{\partial \dot{q}} \\ \frac{\partial}{\partial q}\left(\frac{\partial y_2}{\partial q}\dot{q}\right) & \frac{\partial y_2}{\partial q} \end{bmatrix}}_{L_f y(x)} f(x) + \underbrace{\begin{bmatrix} \frac{\partial y_1}{\partial \dot{q}} \\ \frac{\partial y_2}{\partial q} \end{bmatrix}}_{A} g(x)\, u, \qquad (22)$$

now in the form of (4). This can then be combined with the convergence constraint given for a CLF (16) and posed as an optimization problem to find a satisfactory input u.

Control Lypaunov Function Quadratic Programs: In its traditional implementation [5], the inequality constraint (16) can be posed in a QP optimization based controller to find a torque in the set (17), where $\|v\|^2$ is minimized, as:

CLF-QP:

$$u^*(x) = \underset{u \in U \subset \mathbb{R}^m}{\operatorname{argmin}} \quad \|A(x)u + L_f^{(r)}y(x)\|^2$$

$$\text{s.t.} \quad L_f V(x) + L_g V(x)u \leq \gamma V(x)$$

In the case of a RES-CLF, γ depends on ϵ. For the holonomic constraints to be satisfied in the dynamics (20), and thus in the QP constraint (16), we must either augment u with λ as an additional decision variable [6], [13], or solve for the generalized force assuming satisfying the holonomic constraint:

$$\lambda = (J_c D^{-1} J_c^T)^{-1}\left(J_c D^{-1}(H - Bu) - \dot{J}_c \dot{q}\right), \qquad (23)$$

and substitute back into the expression (20). However, even if λ is included as an additional optimization variable, (23) must be evaluated in order to apply feasibility constraints such as the friction cone to the problem. Additionally, Featherstone showed in previous work that the condition number of the joint space inertia matrix increases quartically with the length of a kinematic chain [9]. This points to an obvious source of numerical stiffness, and can lead to controller degradation on hardware [23]. For complex multi-link robots, such as bipedal robots, these condition numbers are often exceptionally large (for full humanoids sometimes on the order of 10^8). In addition, performing the required inversions for evaluating the vector fields (20) are very computationally expensive, and can often violate strict timing requirements when implementing these controllers on hardware.

Inverse Dynamics Approaches to Locomotion: *Inverse dynamics* is a widely used method to approaching controller design for achieving a variety of motions and force interactions, typically in the form of task-space objectives. Given a target behavior, the dynamics of the robotic system are inverted to obtain the desired torques. In most formulations, the system dynamics are mapped onto a support-consistent manifold using methods such as the dynamically consistent support null-space [28], linear projection [3], and orthogonal projection [21]. When prescribing behaviors in terms of purely task space objectives, this is commonly referred to as task- or operational-space control (OSC) [16]. In many recent works, variations of these approaches have been shown to allow for high-level tasks to be encoded with intuitive constraints and costs in optimization based controllers, some examples being [7], [10], [14], [15], [18]–[20].

Here we present a minimal implementation of an inverse dynamics controller. First, let us consider a set of variables $\mathcal{X} = [\ddot{q}^T, u^T, \lambda^T]^T \in \mathbb{X}_{\text{ext}} := \mathbb{R}^n \times U \times \mathbb{R}^{m_h}$, which are linear with respect to (18)

$$\begin{bmatrix} D(q) & -B & -J^T(q) \end{bmatrix}\mathcal{X} + H(q, \dot{q}) = 0. \qquad (24)$$

We can pose the holonomic constraints (19) as:

$$\begin{bmatrix} J(q) & 0 & 0 \end{bmatrix}\mathcal{X} + \dot{J}(q)\dot{q} = 0. \qquad (25)$$

Also consider a positional objective in the task space of the robot, which can be characterized using (21) as

$$J_y(q, \dot{q})\ddot{q} + \dot{J}_y(q, \dot{q})\dot{q} - \ddot{y}_2^* = 0, \qquad (26)$$

where $\ddot{y}_2^* = K_P y_2 + K_D \dot{y}_2$ is a PD control law which can be tuned to achieve convergence. In it's most basic case, not considering physical limitations on torque and frictional contact, we can pose this tracking problem as:

ID-QP:

$$\mathcal{X}^*(x) = \underset{\mathcal{X} \in \mathbb{X}_{\text{ext}}}{\operatorname{argmin}} \quad \|\dot{J}_y(q, \dot{q})\ddot{q} + J_y(q, \dot{q})\dot{q} - \ddot{y}_2^*\|^2 + \sigma W(\mathcal{X})$$

$$\text{s.t.} \quad D(q)\ddot{q} + H(q, \dot{q}) = Bu + J^T(q)\lambda$$

$$J(q)\ddot{q} + \dot{J}(q)\dot{q} = 0$$

---

where $W(\mathcal{X})$ is included as a regularization term with a small weight σ such that the problem is well posed.

IV. CONTROLLER FORMULATION

In this section, a new controller is presented that combines aspects of ID and CLF based control. The resulting optimization only requires a single use of the mass matrix, in its uninverted form, and incentivizes fast convergence rates.

A Combined Approach. Taking inspiration from inverse dynamics approaches, we return to (15) where the auxiliary control input v is set to equal the second time derivative of the output. Rather than directly choosing an input u, a $\ddot{q}$ is solved for that generates an equivalent response in the outputs. Using (21), $\ddot{q}$ can be chosen to satisfy

$$\begin{bmatrix} \dot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = \dot{J}_y \dot{q} + J_y \ddot{q} = v. \tag{27}$$

By constraining

$$\ddot{q} = J_y^\dagger(-\dot{J}_y \dot{q} + v) \tag{28}$$

where $J_y^\dagger$ is a right pseudo inverse of the full rank matrix J_y, with $J_y J_y^\dagger = I$, and the outputs evolve as:

$$\begin{bmatrix} \dot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = \dot{J}_y \dot{q} + J_y J_y^\dagger(-\dot{J}_y \dot{q} + v) = v. \tag{29}$$

More formally, we have shown the following result:

Theorem 1. *For a robotic system with dynamics* (18) *and outputs of the form* (5) *and* (6)*, any controller in the set:*

$$K(q, \dot{q}) = \{u \in U : \ddot{q} = J_y^\dagger(-\dot{J}_y \dot{q} + v)\} \tag{30}$$

elicits the same response in the output dynamics as the IO feedback linearizing controller

$$u = A^{-1}(-L_f y(x) + v). \tag{31}$$

As discussed in Section II, feedback linearizing controllers fail to take advantage of the natural dynamics of a system. Therefore, we introduce the *Inverse Dynamics Control Lyapunov Function Quadratic Program* (**ID-CLF-QP**) a parallel to (**CLF-QP**), which similarly enables the system to evolve to the prescribed dynamics in a more natural way, while still enforcing convergence guarentees.

ID-CLF-QP

$$\mathcal{X}^* = \underset{\mathcal{X} \in \mathbb{X}_{\text{ext}}}{\arg\min} \quad \left\| \frac{\partial \dot{y}}{\partial q} \dot{q} + \frac{\partial y(q)}{\partial q} \ddot{q} \right\|^2 + \sigma W(\mathcal{X})$$

$$\text{s.t.} \quad L_F V(x) + L_G V(x)\left(\frac{\partial \dot{y}}{\partial q} \dot{q} + \frac{\partial y(q)}{\partial q} \ddot{q} \right) \leq -\gamma V(x)$$

$$D(q)\ddot{q} + H(q, \dot{q}) = Bu + J^T(q)\lambda$$

$$J(q)\ddot{q} + \dot{J}(q)\dot{q} = 0$$

This formulation imposes an equivalent convergence condition as (**CLF-QP**). However, using $\ddot{q}$ as an optimization variable leads to a formulation that is less numerically stiff and less sensitive to estimation errors in the mass matrix.

Incentivized Convergence. A secondary weakness of the standard (**CLF-QP**) is that it does not incentivize faster convergence rates than the chosen γ. It can be seen that

when the outputs are written as in (21), the derivative of the Lyapunov function is only in terms of the decision variable $\ddot{q}$ and scalar functions of the states. Therefore, we add the $\ddot{q}$-dependent portion to the cost.

ID-CLF-QP$^+$

$$\mathcal{X}^* = \underset{\mathcal{X} \in \mathbb{X}_{\text{ext}}}{\arg\min} \quad \left\| \frac{\partial \dot{y}}{\partial q} \dot{q} + \frac{\partial y(q)}{\partial q} \ddot{q} \right\|^2 + \sigma W(\mathcal{X}) + \dot{V}(x, \mathcal{X})$$

$$\text{s.t.} \quad L_F V(x) + L_G V(x)\left(\frac{\partial \dot{y}}{\partial q} \dot{q} + \frac{\partial y(q)}{\partial q} \ddot{q} \right) \leq -\gamma V(x)$$

$$D(q)\ddot{q} + H(q, \dot{q}) = Bu + J^T(q)\lambda$$

$$J(q)\ddot{q} + \dot{J}(q)\dot{q} = 0$$

Theorem 2. *Through the addition of a Lyapunov term in the cost,* (**ID-CLF-QP$^+$**) *will induce an equal or faster convergence rate than* (**ID-CLF-QP**)*. Concretely, given solutions to these optimization problems, denoted by $\mathcal{X}^+$ and $\tilde{\mathcal{X}}$, respectively, for:*

$$\begin{aligned} \dot{V}(x, \mathcal{X}^+) &\leq -\gamma^+ V(x) \\ \dot{V}(x, \tilde{\mathcal{X}}) &\leq -\tilde{\gamma} V(x) \end{aligned} \qquad \Rightarrow \qquad \tilde{\gamma} \leq \gamma^+.$$

Proof: We begin by noting that (**ID-CLF-QP**) and (**ID-CLF-QP$^+$**) have the same solution space. The cost of (**ID-CLF-QP**) as will be denoted:

$$C(\mathcal{X}) = \left\| \frac{\partial \dot{y}}{\partial q} \dot{q} + \frac{\partial y(q)}{\partial q} \ddot{q} \right\|^2 + \sigma W(\mathcal{X}) \tag{32}$$

For a given feasible point $\mathcal{X}^*$ of (**ID-CLF-QP$^+$**), the unique instantaneous convergence rate γ^* is defined as solving:

$$L_F V(x) + L_G V(x)\left(\frac{\partial \dot{y}}{\partial q} \dot{q} + \frac{\partial y(q)}{\partial q} \ddot{q}^* \right) = -\gamma^* V(x), \tag{33}$$

and the cost is given by:

$$\mathcal{J}(\mathcal{X}^*) = C(\mathcal{X}^*) + L_G V(x) \frac{\partial y(q)}{\partial q} \ddot{q}. \tag{34}$$

A larger value for γ means that the solution has a faster convergence rate. Let us call the solution to (**ID-CLF-QP**) $\tilde{\mathcal{X}}$. By definition:

$$C(\tilde{\mathcal{X}}) \leq C(\mathcal{X}^*), \quad \forall \mathcal{X}^* \in \mathcal{X} \tag{35}$$

and additionally, for the solution to (**ID-CLF-QP$^+$**), $\mathcal{X}^+$:

$$\mathcal{J}(\mathcal{X}^+) \leq \mathcal{J}(\tilde{\mathcal{X}}) \tag{36}$$

as the two problems have the same feasible space. This can be expanded to

$$\underbrace{L_G V(x) \frac{\partial y(q)}{\partial q} \ddot{q}^+ + C(\mathcal{X}^+)}_{\mathcal{J}(\mathcal{X}^+)} \leq \underbrace{L_G V(x) \frac{\partial y(q)}{\partial q} \ddot{\tilde{q}} + C(\tilde{\mathcal{X}})}_{\mathcal{J}(\tilde{\mathcal{X}})}$$

If we solve for the Lyapunov portion of each side using (33), this reduces to

$$-\gamma^+ V(x) + C(\mathcal{X}^+) \leq -\tilde{\gamma} V(x) + C(\tilde{\mathcal{X}})$$

This can be rearranged, and (35) can be leveraged to find

$$\tilde{\gamma}V(x) - \gamma^+ V(x) \leq C(\tilde{\mathcal{X}}) - C(\mathcal{X}^+) \leq 0$$
$$(\tilde{\gamma} - \gamma^+)V(x) \leq 0$$
$$\tilde{\gamma} \leq \gamma^+$$

Which proves that the solution to (**ID-CLF-QP$^+$**) will have an equal or faster convergence rate as (**ID-CLF-QP**). ∎

V. IMPLEMENTABLE METHODS

In order to implement these methods on robotic systems, there must be a discussion on practicality and how to better encode and satisfy the physical limitations of the system at hand. In this section, some of the barriers to implementation are presented as well as methods for mitigating them.

Holonomic Constraints. For robotic systems, two types of holonomic constraints are commonly considered, external contact constraints depending on the current configuration of the robot and it's interactions with the world, and internal kinematic constraints resulting from the robot geometry.

1) Contact Constraints: When the robot is in contact with the world, its motion can be restricted. This results in force terms in the equations of motion ((18) and (19)). These contacts are often required to follow friction models. Ideally, a classical *Amontons-Coulomb model* of (dry) friction is used to avoid slippage and is represented as a friction cone constraint. For a friction coefficient μ and a surface normal, the space of valid reaction forces is,

$$\mathcal{C} = \left\{ (\lambda_x, \lambda_y, \lambda_z) \in \mathbb{R}^3 \,\middle|\, \lambda_z \geq 0; \sqrt{\lambda_x^2 + \lambda_y^2} \leq \mu \lambda_z \right\}. \quad (37)$$

However, this constraint is nonlinear, and cannot be implemented as a linear constraint. An alternative solution is to use a *pyramidal friction cone* approximation [12],

$$\mathcal{P} = \left\{ (\lambda_x, \lambda_y, \lambda_z) \in \mathbb{R}^3 \,\middle|\, \lambda_z \geq 0; |\lambda_x|, |\lambda_y| \leq \frac{\mu}{\sqrt{2}} \lambda_z \right\}. \quad (38)$$

This is a more conservative model than the friction cone, but is advantageous in that it is a linear inequality constraint. When a surface is in contact with the outside world, additional constraints are introduced to prevent it from rolling over the contact edge in the form:

$$-\frac{l}{2}\lambda_z < \lambda_{mx} < \frac{l}{2}\lambda_z \quad (39)$$
$$-\frac{w}{2}\lambda_z < \lambda_{my} < \frac{w}{2}\lambda_z \quad (40)$$

where l and w are the lengths and widths of the surface [29].

2) Internal Constraints: It is common practice to model robotic manipulators in tree structures. When the mechanism has parallel manipulators, this is managed by cutting the loop and enforcing a holonomic constraint dictating the physical reality [26]. These contraints add further degrees of complexity to the optimization problem.

Relaxed CLF-QP. Due to these constraints, as well as limits on feasible torques, it is not always possible for the system to converge according to the bound [11]. The accepted way of dealing with this is to add a relaxation term δ to the

convergence constraint. In our formulation, this transforms the problem to:

ID-CLF-QP$^+$ δ

$$\mathcal{X}^* = \underset{\mathcal{X} \in \mathbb{X}_{\text{ext}}, \delta \in \mathbb{R}}{\operatorname{argmin}} \left\| \frac{\partial \dot{y}}{\partial q}\dot{q} + \frac{\partial y(q)}{\partial q}\ddot{q} \right\|^2 + W(\mathcal{X}) + \dot{V}(x, \mathcal{X}) + \rho\delta^2$$

$$\text{s.t.} \quad L_F V(x) + L_G V(x)\left(\frac{\partial \dot{y}}{\partial q}\dot{q} + \frac{\partial y(q)}{\partial q}\ddot{q} \right) \leq -\gamma V(x) + \delta$$

$$D(q)\ddot{q} + H(q, \dot{q}) = Bu + J^T(q)\lambda$$

$$J(q)\ddot{q} + \dot{J}(q)\dot{q} = 0$$

In practice it can be seen that if we take away the hard constraint in (**ID-CLF-QP$^+$**), we are left with a relaxation

---
ID-CLF-QP$^+$ relaxed

$$\mathcal{X}^* = \underset{\mathcal{X} \in \mathbb{X}_{\text{ext}}}{\operatorname{argmin}} \left\| \frac{\partial \dot{y}}{\partial q}\dot{q} + \frac{\partial y(q)}{\partial q}\ddot{q} \right\|^2 + W(\mathcal{X}) + \dot{V}(x, \mathcal{X})$$

$$\text{s.t.} \quad D(q)\ddot{q} + H(q, \dot{q}) = Bu + J^T(q)\lambda$$

$$J(q)\ddot{q} + \dot{J}(q)\dot{q} = 0$$

---

that still incentivizes fast convergence and penalizes slow convergence. Further, whenever it is feasible to do so, this problem will render $\dot{V}$ as negative as possible. In the simulation results we show how these methods compare and we implement the final approach on hardware.

Hard and Soft Constraints. When implementing on hardware, often holonomic constraints are not satisfied precisely. The analytical solutions presented thus far make the problem more prone to infeasibility. To solve this, we once again look to the inverse dynamics community where it has become practice to differentiate between *hard* and *soft* constraints. Hard constraints are formulated as traditionally seen in Section IV, they cannot be violated. Soft constraints, however, refer to an addition to the cost function which penalizes violation of a preferred relationship. These are frequently added as the norm of a least squares problem:

$$w\|\mathcal{A}\mathcal{X} - b\|^2 \quad (41)$$

where $\mathcal{X}$ is as in (24) and w is a weight. On hardware, holonomic constraints for footholds are the perfect candidate to be implemented as soft constraints

$$J(q)\ddot{q} + \dot{J}\dot{q} = 0 \Rightarrow \underbrace{\begin{bmatrix} J(q) & 0 & 0 \end{bmatrix}}_{\mathcal{A}} \mathcal{X} = \underbrace{-\dot{J}\dot{q}}_{b} \quad (42)$$

The formulation of holonomic constraints in this way voids the need to explicitly compute reaction forces (as in (23)) and allows for small violations, which is necessary in practice. Additional soft constraints that are beneficial for robotic walking include specifying force distributions (weight per foot or in different places on the foot). In this case the $\mathcal{A}$ matrix represents the fractional representation and the b matrix is all zeros. Finally, direct tracking of decision variables is possible by making $\mathcal{A}$ the identity and b the desired values. In each of these cases, a desirable cost is the exact satisfaction of the output dynamics. The benefit of
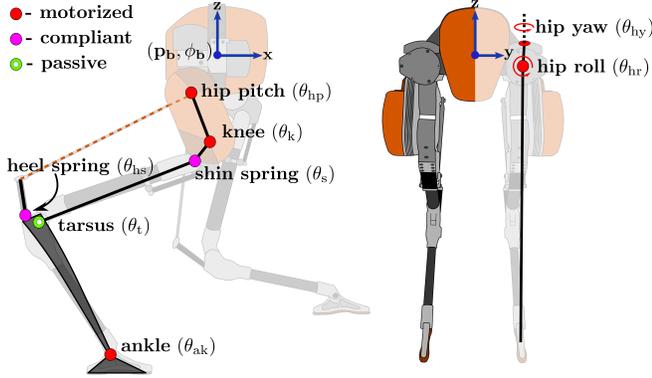
Fig. 2. The configuration coordinates of the Cassie robot, on the left is a side view of the robot, which highlights the compliant mechanism, and on the right is the front view of the robot model.

using soft constraints in this way is two-fold; as previously mentioned it allows for small violations of constraints and, in addition, it speeds up computation time as the problem becomes better posed.

VI. APPLICATION TO THE CASSIE BIPED

In this section the robot model will be introduced, followed by a presentation of simulation results for walking with each of the controllers mentioned, as well as real-time hardware results on crouching and standing behaviors.

Robot Model. The bipedal robot Cassie was designed and manufactured by Agility Robotics[1]. The design of the robot encompasses the physical attributes of the spring loaded inverted pendulum (SLIP) model dynamics. The primary characteristic being a pair of light-weight legs with a heavy torso so that the system is approximated by a point-mass with virtual springy legs. On Cassie, a *compliant multi-link mechanism* is used to transfer power from higher to lower limbs without allocating the actuators' weight onto the lower limbs, and effectively acts as a pair of springy legs. Contacts with the ground are assumed to be rigid and only occurring at specified points on the feet of the robot. This allows for the equations of motion for the robot to be described as (18).

A. Walking in Simulation

The simulation presents a side-by-side comparison of the traditional (**CLF-QP**) with the new controllers proposed. Two walking gaits are generated using the *partial hybrid zero dynamic* framework as presented in [30]. Both gaits use a single continuous domain, and progress is dictated by $\tau(t, q)$, a parameterization of time either by the gait duration (time-based outputs) or by the relative degree 1 output as

$$\tau(t) := \frac{t - t^1}{t^2 - t^1} \quad \text{or} \quad \tau(q) := \frac{\delta y_1(q) - \delta y_1(q^+)}{\bar{v}} \quad (43)$$

where t_0 and t_f are the start and end times of the current domain, respectively, $\delta y_1(q^+)$ is the initial value of the velocity modulating output and $\bar{v}$ is a parameter for the desired velocity of the output.

Planar Walking Simulation. The first gait we consider is state-based and is designed on a planar, rigid model of Cassie. One relative degree 1 input, the linearized forward hip velocity, is used and five relative degree 2 outputs:

$$y_2^a(q) := \begin{bmatrix} \|\psi^{sw}\|_{\ell_2} \\ \|\psi^{st}\|_{\ell_2} \\ \text{atan2}(\psi^{sw}(1), \psi^{sw}(3)) \\ \phi_y \\ \phi^y(q) \end{bmatrix} \begin{pmatrix} \text{swing leg length} \\ \text{stance leg length} \\ \text{swing leg pitch} \\ \text{pelvis pitch} \\ \text{swing foot pitch} \end{pmatrix} \quad (44)$$

where $\phi^y(\theta_{tp})$ is the ankle Cartesian pitch,

$$\bar{p}_{\text{com}}(q) = p_b + R(\phi_b) \begin{bmatrix} 0 & 0 & -0.125 \end{bmatrix}^T \quad (45)$$

is the "average" center of mass position of the robot, $R(\phi_b)$ is the rotation matrix associated with the floating base, and

$$\psi^{st/sw} = \bar{p}_{\text{com}} - p_{tp}^{st/sw} \quad (46)$$

represent the length from the center of mass to each foot.

Four controllers are then applied to the model; (**CLF-QP-δ**), (**ID-CLF-QP-δ**), (**ID-CLF-QP+-δ**), and (**ID-CLF-QP+-relaxed**). In Fig. 3 the convergence of the Lyapunov function can be seen for a system that begins at rest (not started on its periodic orbid) and must converge onto the periodic gait. It can be seen that the more traditionally formulated controllers do not converge quickly enough in the first step, causing an amplification of error in the second, while the two cases with the Lyapunov term in the cost do. It is also interesting to note that when the Lyapunov term is in the cost, the existence of the hard convergence constraint does not significantly affect the response. While the performance differs between the four controllers, the torque applied from each is similar in magnitude and form, as can be seen in Fig. 3. The inverse dynamics torques are overall smoother, and the controllers with the Lyapunov term in the cost have the smoothest torque profiles and best convergence performance.

3D Compliant Walking Simulation. The second simulation case studied a time-based walking gait on the 3D compliant model of the robot [26]. For this formulation the relative degree 1 output is disregarded and four new relative degree 2 outputs - both hip yaws (θ_{hy}), the swing hip roll (θ_{hr}) and the floating base roll (ϕ_r) - are added. The gait generated is
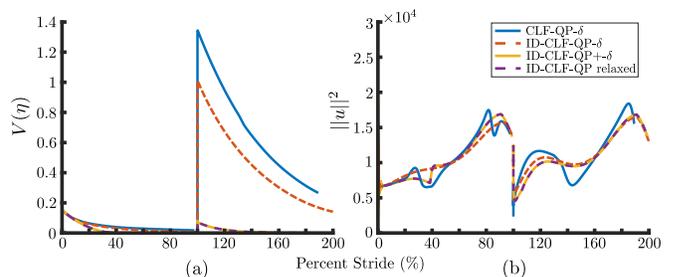

Fig. 3. Performance of the planar simulated walking gait over two steps, started from rest. Improvement is seen when the Lyapunov term is added to the cost (a) Lyapunov function (b) $\|u\|^2$
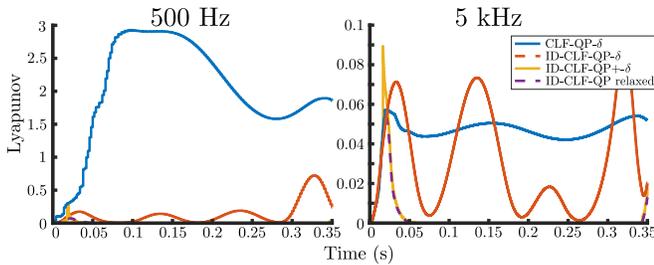
Fig. 4. Lyapunov function convergence on the 3D compliant robot for a time based step at 500 Hz and 5 kHz control frequencies.
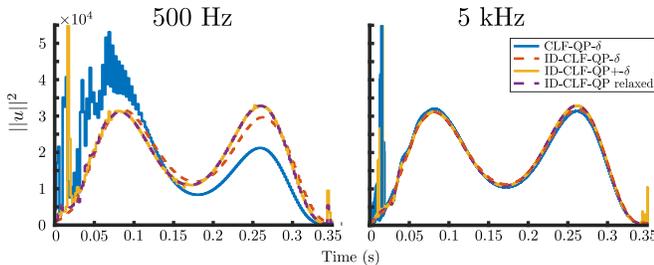

Fig. 5. Torque of the 3D compliant robot for time based step at 500 Hz and 5 kHz control frequencies.

tracking with the controllers directly on the nominal walking gait motions from an offline trajectory optimization.

The Lyapunov function convergence and motor torques for each of the controllers can be seen in Fig. 4 and 5, respectively. The theory referenced in this work assumes purely continuous control, however, in reality torques are applied at a discrete intervals. We thus included the simulation results when the controllers are applied at 500 Hz and 5 kHz. While the traditional (**CLF-QP-**δ) and (**ID-CLF-QP-**δ) controllers see a marked degradation as loop rates decreases, the controllers which have Lyapunov terms in the cost, (**ID-CLF-QP+-**δ), and (**ID-CLF-QP+-relaxed**), seem minimally affected. Because this controller is run on the compliant model, the ODE is much more numerically stiff than in the rigid planar case. As such, we see that (**CLF-QP-**δ), which uses the inverted form of the mass inertia matrix, is much more sensitive when applied at coarse frequencies.

Crouching in Real-Time on Hardware. Finally, the inverse dynamics motivated control Lyapunov based controller, (**ID-CLF-QP+-relaxed**), was implemented on hardware, as Cassie went through a dynamic crouching and standing motion (Fig. 6). In addition, a video of the experiment, along with animations of the previous simulations are provided in [1]. Six relative degree two outputs for standing were prescribed, the base positions and rotations, $y(q) = [p_b, \; \phi_b]^T$. They were then specified as high level targets on hardware. Because we are using task-space approach, it is not necessary to encode these objectives as combinations of the actuated joint angles, and no joint level stabilization (i.e. individual joint tracking or control) was used. The controller was run on the real-time PC aboard Cassie, in a multithreaded real-time Simulink model, and with a control rate of 250 Hz. There were only three sets of hard constraints; the dynamics

as in (18), torque bounds for each joint, and the friction constraints as in (38) and (39). The cost function included soft constraints for the remaining holonomic constraints as well as a cost on torque smoothness $(u_k - u_{k-1})$, in addition to the costs explicitly prescribed in (**ID-CLF-QP+-relaxed**). As can be seen in Fig. 8 the height was smoothly tracked to within several centimeters for the entirety of the motion. The norm of torque applied to all motors for one step can be seen in Fig. 9, which are mostly smooth with the exception of some oscillation during the downward portion of the motion. This was observed to be a byproduct of the passive compliance from the springs being unloaded as the robot swept downwards. In addition, in Fig. 7 the contact forces are shown to adhere to the friction cone (37) constraint.

VII. Conclusions

This paper presented an optimization based controller which leverages the desireable convergence results provided by control Lyapunov functions combined with implementation concepts from inverse dynamics based controllers. The approach was shown to be successful both in simulation and on hardware in real time. Further, the inclusion of a Lyapunov term in the cost helped incentive the system to converge more rapidly (as was proved in Theorem 2) and improved performance with respect to discretization and model innaccuracy/stiffness. This was demonstrated in simulation with walking, and experimentally with crouching.

Future work will explore improving the efficiency of the control method, with the intention of demonstrating walking in real-time on hardware. Additionally, it opens to door to providing more implementable methods for safety critical systems through control barrier functions. If this paper is accepted, the authors intend to release open-source versions of the code for both the simulation and real-time hardware.

References

[1] Video of simulation and experimental results on Cassie. https://vimeo.com/362721158.

[2] Yeuhi Abe, Marco Da Silva, and Jovan Popović. Multiobjective control with frictional contacts. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 249–258. Eurographics Association, 2007.

[3] Farhad Aghili. A unified approach for inverse and direct dynamics of constrained multibody systems based on linear projection operator: applications to control and simulation. *IEEE Transactions on Robotics*, 21(5):834–849, 2005.

[4] Aaron D Ames, Kevin Galloway, and Jessy W Grizzle. Control lyapunov functions and hybrid zero dynamics. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 6837–6842. IEEE, 2012.

[5] Aaron D Ames, Kevin Galloway, Koushil Sreenath, and Jessy W Grizzle. Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics. *IEEE Transactions on Automatic Control*, 59(4):876–891, 2014.

[6] Aaron D Ames and Matthew Powell. Towards the unification of locomotion and manipulation through control lyapunov functions and quadratic programs. In *Control of Cyber-Physical Systems*, pages 219–240. Springer, 2013.

[7] Taylor Apgar, Patrick Clary, Kevin Green, Alan Fern, and Jonathan W Hurst. Fast online trajectory optimization for the bipedal robot cassie. In *Robotics: Science and Systems*, 2018.

[8] John Betts and Ilya Kolmanovsky. Practical methods for optimal control using nonlinear programming, 2002.
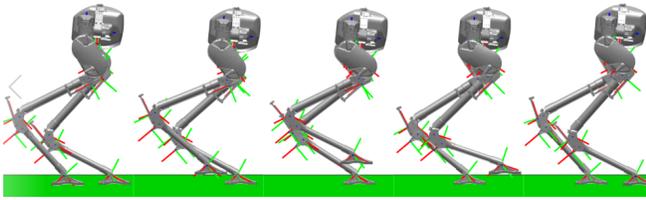
Fig. 6. Time series motion tiles for simulated walking in 3D (left) and on hardware for the crouching experiment (right).
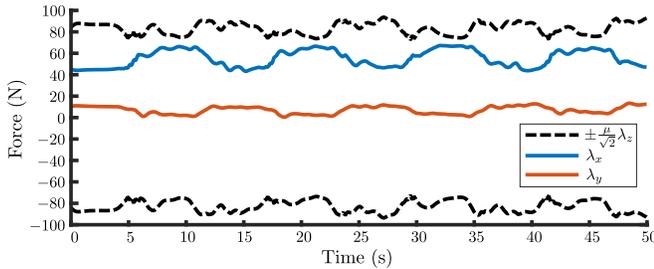

Fig. 7. Contact forces adhering to the friction cone over four crouches on hardware.
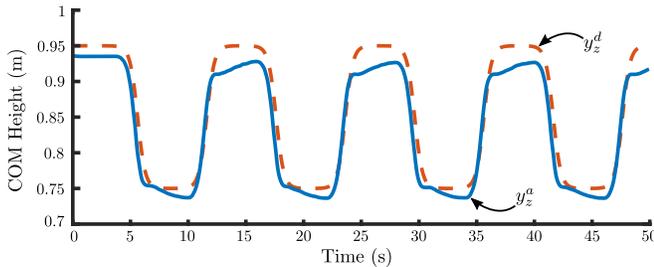

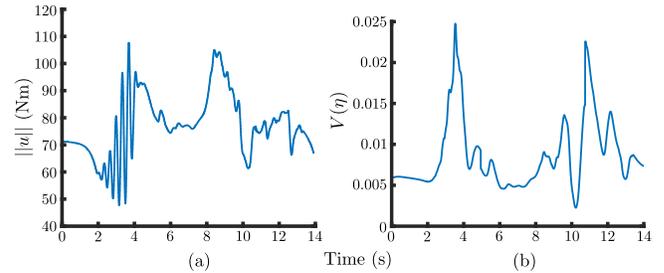Fig. 8. COM height tracking over four crouches on hardware.

Fig. 9. Torque and Lyapunov function over one crouch on hardware.

Design of a momentum-based control framework and application to the humanoid robot atlas. *International Journal of Humanoid Robotics*, 13(01):1650007, 2016.

[19] Scott Kuindersma, Robin Deits, Maurice Fallon, Andrés Valenzuela, Hongkai Dai, Frank Permenter, Twan Koolen, Pat Marion, and Russ Tedrake. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous Robots*, 40(3):429–455, 2016.

[20] Scott Kuindersma, Frank Permenter, and Russ Tedrake. An efficiently solvable quadratic program for stabilizing dynamic locomotion. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2589–2594. IEEE, 2014.

[21] Michael Mistry, Jonas Buchli, and Stefan Schaal. Inverse dynamics control of floating base systems using orthogonal decomposition. In *2010 IEEE international conference on robotics and automation*, pages 3406–3412. IEEE, 2010.

[22] Richard M. Murray, Zexiang Li, Sastry, and S. Shankar Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[23] Jun Nakanishi, Rick Cory, Michael Mistry, Jan Peters, and Stefan Schaal. Operational space control: A theoretical and empirical comparison. *The International Journal of Robotics Research*, 27(6):737–757, 2008.

[24] Quan Nguyen and Koushil Sreenath. Optimal robust control for bipedal robots through control lyapunov function based quadratic programs. In *Robotics: Science and Systems*. Rome, Italy, 2015.

[25] Michael Posa, Cecilia Cantu, and Russ Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *The International Journal of Robotics Research*, 33(1):69–81, 2014.

[26] Jenna Reher, Wen-Loong Ma, and Aaron D Ames. Dynamic walking with compliance on a cassie bipedal robot. *arXiv preprint arXiv:1904.11104*, 2019.

[27] Shankar Sastry. *Nonlinear systems: analysis, stability, and control*, volume 10. Springer Science & Business Media, 2013.

[28] Luis Sentis. *Synthesis and control of whole-body behaviors in humanoid systems*. Stanford university USA, 2007.

[29] Miomir Vucobratovic, Brainislav Borovac, Dusan Surla, and Dragan Stokic. Biped locomotion: dynamics, stability, control and applications, 1990.

[30] Eric R Westervelt, Jessy W Grizzle, Christine Chevallereau, Jun Ho Choi, and Benjamin Morris. *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2018.

[31] Eric R Westervelt, Jessy W Grizzle, and Daniel E Koditschek. Hybrid zero dynamics of planar biped walkers. *IEEE transactions on automatic control*, 48(1):42–56, 2003.

[32] Xiaobin Xiong and Aaron D. Ames. Coupling reduced order models via feedback control for 3D underactuated bipedal robotic walking. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 1–9. IEEE, 2018.

[9] Roy Featherstone. An empirical study of the joint space inertia matrix. *The International Journal of Robotics Research*, 23(9):859–871, 2004.

[10] Siyuan Feng, Eric Whitman, X Xinjilefu, and Christopher G Atkeson. Optimization-based full body control for the DARPA robotics challenge. *Journal of Field Robotics*, 32(2):293–312, 2015.

[11] Kevin Galloway, Koushil Sreenath, Aaron D Ames, and Jessy W Grizzle. Torque saturation in bipedal robotic walking through control lyapunov function-based quadratic programs. *IEEE Access*, 3:323–332, 2015.

[12] Jessy Grizzle, Christine Chevallereau, Ryan Sinnet, and Aaron Ames. Models, feedback control, and open problems of 3D bipedal robotic walking. *Automatica*, 50(8):1955 – 1988, 2014.

[13] Ayonga Hereid, Matthew J Powell, and Aaron D. Ames. Embedding of SLIP dynamics on underactuated bipedal robots through multiobjective quadratic program based control. In *53rd IEEE Conference on Decision and Control*, pages 2950–2957. IEEE, 2014.

[14] Alexander Herzog, Nicholas Rotella, Sean Mason, Felix Grimminger, Stefan Schaal, and Ludovic Righetti. Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid. *Autonomous Robots*, 40(3):473–491, 2016.

[15] Marco Hutter, Hannes Sommer, Christian Gehring, Mark Hoepflinger, Michael Bloesch, and Roland Siegwart. Quadrupedal locomotion using hierarchical operational space control. *The International Journal of Robotics Research*, 33(8):1047–1062, 2014.

[16] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

[17] Shishir Kolathaya, Ayonga Hereid, and Aaron D Ames. Time dependent control lyapunov functions and hybrid zero dynamics for stable robotic locomotion. In *2016 American Control Conference (ACC)*, pages 3916–3921. IEEE, 2016.

[18] Twan Koolen, Sylvain Bertrand, Gray Thomas, Tomas De Boer, Tingfan Wu, Jesper Smith, Johannes Englsberger, and Jerry Pratt.