Towards the Unification of Locomotion and Manipulation through Control Lyapunov Functions and Quadratic Programs

Aaron D. Ames and Matthew Powell*

Mechanical Engineering
Texas A&M University
College Station TX, 77840
{aames,mjpowell}@tamu.edu

Abstract. This paper presents the first steps toward unifying locomotion controllers and algorithms with whole-body control and manipulation. A theoretical framework for this unification will be given based upon quadratic programs utilizing control Lyapunov functions. In particular, we will first consider output based feedback linearization strategies for locomotion together with whole-body control methods for manipulation. We will show that these two traditionally disjoint methods are equivalent through the correct choice of controller. We will then present a method for unifying these two methodologies through the use of control Lyapunov functions presented in the form of a quadratic program. In addition, it will be shown that these controllers can be combined with force-based control to achieve locomotion and force-based manipulation in a single framework. Finally, simulation results will be presented demonstrating the validity of the proposed framework.

Keywords: Robotics, cyber-physical systems, bipedal locomotion, manipulation, control Lyapunov functions, quadratic programs.

1 Introduction

Robots provide a quintessential example of cyber-physical systems (CPSs); the ability of robots, and especially humanoid robots, to perform complex dynamic tasks requires a complete and unified understanding of dynamics, control, software and hardware, along with their interconnection and integration. In the context of the control of robotic CPSs, a variety of traditionally disparate approaches have been taken ranging from nonlinear control via input/output (IO) feedback linearization through output based control [12,34], force-based control methods [7,17,30,21,31], and whole-body control methods for manipulation [14,15,27], to name a few. Each of these methods, along with the wide variety of other methods taken in the control of robotic systems, have proven success in their domains of consideration. Yet combining these different approaches into a single unified and implementable framework remains an open problem. With

* This research is supported by NASA Grants NNX12AQ68G and NNX11AN06H.

a view toward unification, this paper describes how locomotion controllers, as described through human-inspired control methods which utilize IO feedback linearization, can be unified with whole-body and multi-contact force control into a single coherent framework given in the form of quadratic programs utilizing control Lyapunov functions. These theoretical developments will be supported by simulation results.

We begin by reviewing the basic background necessary to introduce these results. In particular, the robotic models considered are defined, and a review of human-inspired control [1,2,3,4,22] is given. In the context of human-inspired control, we obtain outputs, or virtual constraints, that represent the desired behavior of a bipedal robot. We show how, through input/output (IO) linearization, the dynamics of these outputs can transformed into a linear system. The end result of this process is that the dynamics of the outputs can be chosen to achieve convergence of these outputs to zero at a desired rate. This implies the convergence of the robotic system to the surface defined by the zero level set of these output functions: the *zero dynamics surface*. By ensuring that this surface is invariant through impact, i.e., that we have *hybrid zero dynamics (HZD)* [12,29,34], the end result of these control methods is provably stable locomotion.

The first result of this paper is obtained by considering whole-body control methodologies, and specifically null-space control [14,15,27]. In the context of the outputs associated with locomotion, we demonstrate how Jacobians can be constructed in the case of mixed position and velocity based outputs. In addition, by projecting the dynamics of the system down to the operational space dynamics, we are able to show that the end result is dynamics for the outputs that can be utilized to achieve a linear relationship between the inputs and outputs. In more concrete terms, we establish that IO linearization and null-space control result in equivalent constructions in terms of outputs. Building on this idea, we then consider the case when manipulation tasks have been specified. A procedure for merging the locomotion and manipulation tasks is given in the context of null-space control. This method benefits from explicitly separating the locomotion tasks and the manipulation tasks so that the manipulation tasks do not affect the locomotion tasks. This is an advantage due to the fact that the manipulation tasks can therefore never destabilize the robot. Conversely, it can be a disadvantage since it does not allow for dynamic balancing between the locomotion and manipulation tasks. For example, one may want to slightly relax the tracking of the outputs associated with locomotion—as long as it does not destabilize the system—in order to better achieve a given manipulation task.

The need for a dynamic way to balance multiple tasks related to locomotion, manipulation and force control motivates the introduction of quadratic programs (QPs) that allow for this dynamic balancing coupled with the ability to add constraints on the evolution of the system, e.g., torque bounds. We begin by again considering only the locomotion task, and show how the IO representation achieved through output functions can be used to explicitly construct a control Lyapunov function (CLF) [10,28]. Importantly, this CLF results in an inequality that is linear in torque such that, when it is satisfied, convergence to the hybrid zero dynamics surface is guaranteed [5,6]. This naturally leads to the formulation of a quadratic program (QP) in terms of torque with a constraint given by the CLF associated with locomotion [11]. The strength of this representation is that additional constraints can be added to the controller such as torque bounds, moment bounds, etc., and by solving the QP the controller will naturally find the best balance between these hard physical constraints and the control objective of converging to the HZD surface.

To demonstrate the extensibility of the CLF based QP controller, we then extend the formulation to include manipulation tasks in the form of both position-based tasks and force-based tasks. In these cases, additional CLFs can be added to the QP through the form of additional constraints; these CLFs represent the manipulation objectives in the system. The advantage of this representation is that the locomotion and manipulation tasks can be dynamically balanced through the QP—the QP will naturally find the optimal balance between these control objectives. In addition, we will discuss how null-space control can be expressed in this framework. Finally, we discuss force-based tasks in the presence of multi-contact. Again, we show how these can naturally be expressed in the context of QPs and CLFs.

These formal ideas and results are demonstrated through simulation results. We begin by obtaining a stable walking gait for the lower body of a simple full-body (2D) humanoid robot. As a first step, we demonstrate how this purely lower-body controller can be embedded into the full-body robot through a QP, with the end result again being stable walking. Taking this idea even further, we show how manipulation and force tasks can be accomplished without modifying the original locomotion controller. In particular, we perform the task of holding the hand at a constant height and, more significantly, show how we can hold the hand in contact with a wall with a desired force while simultaneously walking with the locomotion controllers that were designed with no knowledge of the upper body. As a final demonstration, we show the robustness of this method through unknown rough terrain locomotion with the full-body robot.

2 Background

We begin by giving some basic terminology utilized throughout the paper. Specifically, we will introduce the basic equations for a robotic system, and show how these are converted to an affine control system (see [20,34] for additional details).

Let $\mathcal{Q}$ be the configuration space of a robot with n degrees of freedom, i.e., $n = \dim(\mathcal{Q})$, with coordinates $q \in \mathcal{Q}$. For the sake of definiteness, it may be necessary to choose $\mathcal{Q}$ to be a subset of the actual configuration space of the robot so that global coordinates can be defined[1], i.e., such that $\mathcal{Q}$ is embeddable in $\mathbb{R}^n$, or more simply $\mathcal{Q} \subset \mathbb{R}^n$. Consider the equations of motion for a robot given in the general form by the Euler-Lagrange equations:

$$D(q)\ddot{q} + H(q,\dot{q}) = Bu, \tag{1}$$

[1] Note that at various points we will assume that matrix functions have full rank; it may be necessary to carefully choose $\mathcal{Q}$ to satisfy these conditions.

where D is the inertia matrix, H is a vector containing the coriolis and gravity terms, and $B \in \mathbb{R}^{n\times n}$ is the actuation matrix which determines the way in which the torque inputs, $u \in \mathbb{R}^n$, actuate the system. Note that here, for the sake of simplicity, we assume full actuation (and hence a square actuation matrix, B).

In the context of control constructions, it is desirable to convert this system to an ODE of the form:

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

where $x = (q, \dot{q}) \in T\mathcal{Q} \subset \mathbb{R}^{2n}$ and

$$f(q,\dot{q}) = \begin{bmatrix} \dot{q} \\ -D^{-1}(q)H(q,\dot{q}) \end{bmatrix}, \quad g(q,\dot{q}) = \begin{bmatrix} 0 \\ D^{-1}(q)B \end{bmatrix}, \tag{2}$$

where $0 \in \mathbb{R}^{n\times n}$ is a matrix of zeros.

When modeling the bipedal robot, discrete behavior must also be considered in conjunction with these continuous dynamics. In particular, the robotic system exhibits discrete impacts when guards are reached, i.e., when contacts with the world are created or broken [13]. The end result is that the system is a hybrid system:

$$\mathscr{HC} = (\mathcal{D}, \mathcal{U}, \mathcal{S}, \Delta, (f,g)), \tag{3}$$

where $\mathcal{D}$ is the domain of the continuous dynamics, i.e., $\mathcal{D} \subset T\mathcal{Q} \subset \mathbb{R}^{2n}$, $\mathcal{U} \subset \mathbb{R}^n$ is the set of admissible control values, $\mathcal{S}$ is the guard which determines when a discrete change in the dynamics occurs, Δ determines the discrete change in dynamics, and (f, g) is the affine control system dictating the continuous dynamics. In particular, this implies that when $(q^-, \dot{q}^-) \in \mathcal{S}$, there exists an impact of the form: $(q^-, \dot{q}^-) \mapsto (q^+, \dot{q}^+) = \Delta(q^-, \dot{q}^-)$.

3 Human-Inspired Control for Locomotion

In the context of mobility, the control algorithms utilized in this paper build off the framework of human-inspired control. While detailed algorithms can be found in [1,2,3,4,23], we will outline these methods only as they relate to the constructions presented in this paper. We note that human-inspired control builds upon the concept of human-inspired constraints that, when enforced with the proper choice of parameters, provably guarantee robotic walking. These methods are applicable both in the case of under and full actuation, and have been applied to both 2D and 3D robots to achieve walking experimentally [22,35].

Consider a *human output combination*: $Y^H = (\mathcal{Q}, y_1^H, y_2^H)$, consisting of the configuration space of a robot, $\mathcal{Q} \subset \mathbb{R}^n$, a velocity modulating output $y_1^H : \mathcal{Q} \to \mathbb{R}$, position modulating outputs $y_2^H : \mathcal{Q} \to \mathbb{R}^{n_\ell - 1}$ given by $y_2^H(q) = [y_2^H(q)_i]_{i\in O}$ with O an indexing set for y_2^H, and n_ℓ the total number of position and velocity modulation outputs. Human-inspired outputs consist of (vector) relative 1 and 2 degree output functions $y_1 : T\mathcal{Q} \to \mathbb{R}$ and $y_2 : \mathcal{Q} \to \mathbb{R}^{n_\ell - 1}$ of the form:

$$y_1(q,\dot{q}) = \frac{\partial y_1^H(q)}{\partial q}\dot{q} - v, \tag{4}$$

$$y_2(q) = y_2^H(q) - [y_{\mathrm{CWF}}(\tau(q), \alpha_i)]_{i\in O}, \tag{5}$$

where y_1 is the velocity-based output, e.g., based upon the center of mass or forward position of the hip, $v \in \mathbb{R}$ the desired velocity, and y_2 are the position based outputs, e.g., a vector of outputs including things like the angle of the stance knee, etc. In this case, $y_2^H(q)$ is the actual value of these outputs as computed from the robot, and $[y_{\mathrm{CWF}}(\tau(q), \alpha_i)]_{i \in O}$ is the desired value of these outputs as computed from the canonical walking function:

$$y_{CWF}(t, \alpha_i) = e^{-\alpha_{i,4} t}(\alpha_{i,1} \cos(\alpha_{i,2} t) + \alpha_{i,3} \sin(\alpha_{i,2} t)) + \alpha_{i,5},$$

which is simply the time-solution to a linear mass-spring damper system. Thus, we drive the relative degree 2 outputs to the behavior of a compliant system, mirroring methods in compliant-based control. Moreover, τ is a parametrization of time based upon the relative 1 degree output:

$$\tau(q) = \frac{y_1^H(q) - y_1^H(q^+)}{v}, \tag{6}$$

with $y_1^H(q^+)$ the initial value of the velocity modulating output, e.g., at the beginning of a step. It follows from these constructions that the control parameters are $v \in \mathbb{R}$ and $\alpha \in \mathbb{R}^{n_\ell - 1 \times 5}$. To provide a specific example, in the case of a 5-link 2D walking robot, $n_\ell = 5$. Therefore, there are a total of 21 control parameters; while this may appear to be a large number, the parameters are automatically determined by a human-inspired optimization and so no tuning of parameters is needed.

With the objective of driving $y_1 \to 0$ and $y_2 \to 0$, we create a linear input/output (IO) relationship [26]. More formally, the goal is to drive the dynamics of the system to the *zero dynamics surface*:

$$\mathbf{Z}_{v,\alpha} = \{(q, \dot{q}) \in T\mathcal{Q} : y_1(q, \dot{q}) = 0,\ \ y_2(q) = 0,\ \ \dot{y}_2(q, \dot{q}) = 0\},$$

which depends on the parameters v and α. With this goal in mind, differentiating the relative degree 1 output once and differentiating the relative degree 2 output twice yields:

$$\begin{bmatrix} \dot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} L_f y_1(q, \dot{q}) \\ L_f^2 y_2(q, \dot{q}) \end{bmatrix}}_{L_f} + \underbrace{\begin{bmatrix} L_g y_1(q, \dot{q}) \\ L_g L_f y_2(q, \dot{q}) \end{bmatrix}}_{A} u, \tag{7}$$

with L denoting the Lie derivative, and A the decoupling matrix [26]. Note that the dependence of L_f and A on q and $\dot{q}$ has been suppressed for notational simplicity. Traditionally, in nonlinear control, this matrix is assumed to be nonsingular, in which case, we can pick:

$$u = A^{-1}(-L_f + \mu) \tag{8}$$

for some $\mu \in \mathbb{R}^{n_\ell}$ resulting in

$$\begin{bmatrix} \dot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = \mu. \tag{9}$$

Therefore, one obtains a linear relationship between input and output. As a result, one can drive the system to the surface $\mathbf{Z}_{v,\alpha}$ by, for example, choosing

$$\mu = \begin{bmatrix} -\epsilon y_1 \\ -2\epsilon \dot{y}_2 - \epsilon^2 y_2 \end{bmatrix} \tag{10}$$

for $\epsilon > 0$; here, as $\epsilon \to \infty$, the speed of convergence to $\mathbf{Z}_{v,\alpha}$ increases. In particular, it yields the output dynamics:

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \ddot{y}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} -\epsilon & 0 & 0 \\ 0 & 0 & I \\ 0 & -\epsilon^2 I & -\epsilon 2I \end{bmatrix}}_{F_{\mathrm{cl}}} \begin{bmatrix} y_1 \\ y_2 \\ \dot{y}_2 \end{bmatrix}, \tag{11}$$

with F_{cl} chosen so that all of the eigenvalues are located at $-\epsilon$. Note that while this choice of μ guarantees convergence, it does not achieve convergence in an optimal fashion; this motivates later constructions related to CLFs.

It is important to note that, in the context of locomotion, the impacts Δ in the hybrid model (3) that occur at foot strike must be considered when designing the control parameters v and α. In particular, the impact map Δ can throw the system away from the surface $\mathbf{Z}_{v,\alpha}$, resulting in the system being destabilized. With this consideration in mind, since we wish to allow the velocity-based relative degree 1 outputs to "jump" during impact, we consider the partial zero dynamics surface:

$$\mathbf{PZ}_{v,\alpha} = \{(q,\dot{q}) \in T\mathcal{Q} : y_2(q) = 0,\ \dot{y}_2(q,\dot{q}) = 0\}.$$

This surface is termed hybrid invariant, or a hybrid zero dynamics surface, if $\Delta(\mathbf{PZ}_{v,\alpha} \cap \mathcal{S}) \subset \mathbf{PZ}_{v,\alpha}$—that is, if the surface is invariant through impacts in the system. Creating this invariance is the basis for human-inspired optimization:

$$\begin{aligned} (v^*, \alpha^*) = \underset{(v,\alpha)\in\mathbb{R}\times\mathbb{R}^{n_\ell - 1\times 5}}{\operatorname{argmin}} \quad & \mathrm{Cost}(v,\alpha) \\ \text{s.t.} \quad & \Delta(\mathbf{PZ}_{v,\alpha} \cap \mathcal{S}) \subset \mathbf{PZ}_{v,\alpha} \end{aligned} \tag{PHZD}$$

which automatically generates parameters v^*,α^* that ensure invariance of this surface while minimizing a cost, $\mathrm{Cost}(v,\alpha)$, often chosen based upon human data [1]. Detailed methods for constructing this optimization problem are given in [3], but the main result of human-inspired control is that it automatically generates stable periodic walking gaits.

4 Merging Locomotion and Whole-Body Control

In the context of unifying the mobility controllers with other manipulation tasks, it is possible to associate Jacobians and null-space projections [14,15,27] to the locomotion tasks. This allows for the development of a controller that will prioritize locomotion and achieve the requested manipulation tasks with the remaining degrees of freedom of the system. Before developing this idea, it is necessary to note that the constructions based upon IO linearization can be reframed in an equivalent manner in the context of null-space control.

4.1 Equivalence between IO and Null-Space Control for Mobility

Consider the outputs y_1 and y_2 constructed in the previous section (see (4) and (5)). These can be viewed as two separate tasks: one for velocity regulation, and one that drives the remaining degrees of freedom to the system to the partial hybrid zero dynamics surface $\mathbf{PZ}_{v,\alpha}$.

Define the Jacobians associated with velocity and position modulating tasks, $y_1(q,\dot{q})$ and $y_2(q)$, as:

$$J_1(q,\dot{q}) := \frac{\partial y_1(q,\dot{q})}{\partial \dot{q}}, \quad J_2(q) = \frac{\partial y_2(q)}{\partial q}.$$

Differentiating the relative 1 degree task, y_1, once and the relative 2 degree task, y_2, twice yields:

$$\begin{bmatrix} \dot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} J_1(q,\dot{q}) \\ J_2(q) \end{bmatrix}}_{J_\ell} \ddot{q} + \underbrace{\begin{bmatrix} \frac{\partial y_1(q,\dot{q})}{\partial q} \\ \dot{J}_2(q,\dot{q}) \end{bmatrix}}_{\dot{J}_\ell} \dot{q}. \tag{12}$$

For simplicity of notation, we will now suppress the dependence of matrices on q and $\dot{q}$.

Through these constructions, we have a Jacobian $J_\ell \in \mathbb{R}^{n_\ell \times n}$ associated with the locomotion task. This allows for the construction of a pseudo-inverse for the locomotion task given by:

$$\bar{J}_\ell = D^{-1} J_\ell^T [J_\ell D^{-1} J_\ell^T]^{-1}.$$

This is a specific example of a right pseudo-inverse for J_ℓ and therefore satisfies: $J_\ell \bar{J}_\ell = I$. In addition, the null-space projection, N_ℓ, associated with these Jacobians is given by:

$$N_\ell^T = [I - J_\ell^T \bar{J}_\ell^T].$$

The Jacobian used to describe the locomotion task can be used to project down to the locomotion dynamics, even in the case when the full-order robot dynamics are of a much larger dimension, i.e., in the case when locomotion is described only for the lower-body, while the robot consists of both a lower and upper body. In particular, the locomotion dynamics are given by choosing:

$$u = J_\ell^T u_\ell$$

for some $u_\ell \in \mathbb{R}^{n_\ell}$, wherein it follows that, by applying the left pseudo-inverse for J_ℓ^T, given by $\bar{J}_\ell^T$, one obtains:

$$\bar{J}_\ell^T (D(q)\ddot{q} + H(q,\dot{q})) = \bar{J}_\ell^T B J_\ell^T u_\ell,$$

where, as long as the locomotion task is consistent with the actuation in the system, $\bar{J}_\ell^T B J_\ell^T$ is invertible; for example, in the case of full actuation and in the proper coordinates $B = I$ and so $\bar{J}_\ell^T B J_\ell^T = I$. Defining

$$D_\ell = (J_\ell D^{-1} J_\ell^T)^{-1}, \quad H_\ell = \bar{J}_\ell^T H - D_\ell \dot{J}_\ell \dot{q}, \quad B_\ell = \bar{J}_\ell^T B J_\ell^T, \tag{13}$$

yields

$$D_\ell \begin{bmatrix} \dot{y}_1 \\ \ddot{y}_2 \end{bmatrix} + H_\ell = B_\ell u_\ell$$

which yields dynamics for the locomotion outputs. Picking

$$u_\ell = B_\ell^{-1}(D_\ell \mu_\ell + H_\ell) \tag{14}$$

results in

$$\begin{bmatrix} \dot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = \mu_\ell. \tag{15}$$

exactly as in the case of IO linearization, i.e., we arrive at exactly the same form as given in (9). Therefore, one can pick μ_ℓ exactly as in (10) to achieve the same convergence objectives. More formally, we have established the following result:

Theorem 1. *For a robotic system with dynamics expressed as* (1) *and* (2) *with outputs of the form* (4),(5)*, the control laws for IO linearization* (8) *and null-space control* (14)*:*

$$u = A^{-1}(-L_f + \mu)$$
$$u = B_\ell^{-1}(D_\ell \mu + H_\ell)$$

yield equivalent output dynamics of the form:

$$\begin{bmatrix} \dot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = \mu. \tag{16}$$

4.2 Merging Locomotion with Manipulation

The advantage to the null-space formulation, as opposed to the IO approach, is that, although the two are equivalent as established by Theorem 1, in the null-space representation there is a well-defined null-space that can be utilized to ensure that any tasks performed on the upper body do not destabilize locomotion and/or balance. Let a manipulation task, or collection of manipulation tasks, be represented by a set of outputs $y_m(q) \in \mathbb{R}^{n_m}$. Defining the Jacobian for these tasks by J_m we obtain (as in the case of locomotion):

$$\ddot{y}_m = \dot{J}_m \dot{q} + J_m \ddot{q}$$

and the left pseudo-inverse for J_m^T, given by $\bar{J}_m^T$, can be again constructed in this case. This allows us to define the corresponding control law:

$$u = J_\ell^T u_\ell + N_\ell^T J_m^T u_m,$$

which guarantees that manipulation tasks do not interfere with locomotion and balance. Using, $\bar{J}_m^T$, we can again project down to the dynamics for manipulation expressed as:

$$D_m \ddot{y}_m + H_m = B_{m,\ell} u_\ell + B_{m,m} u_m,$$

where D_m and H_m are defined as in (13) with the subscripts changed from ℓ to m and

$$B_{m,m} = \bar{J}_m^T B N_\ell^T J_m^T, \quad B_{m,\ell} = \bar{J}_m^T B J_\ell^T,$$

with the locomotion controllers affecting the manipulation controllers via $B_{m,\ell}$.

Begin by assuming that the manipulation tasks are consistent with the locomotion tasks; formally, this is characterized by $\mathrm{Null}(N_\ell^T J_m^T) = \emptyset$. In this case, $B_{m,m}$ is nonsingular and, as in the case of locomotion, we can again shape the manipulation dynamics to be any desired dynamics. To see this we can pick

$$u_m = B_{m,m}^{-1}(D_m \mu_m + H_m - B_{m,\ell} u_\ell)$$

yielding:

$$\ddot{y}_m = \mu_m.$$

If, for example, the goal was to drive $y_m \to 0$, one need only pick

$$\mu_m = -2\epsilon \dot{y}_m - \epsilon^2 y_m. \tag{17}$$

5 Implementation through Quadratic Programs

A method for achieving convergence for both the locomotion tasks and manipulation tasks was presented in the previous sections. Specifically, μ_ℓ and μ_m were chosen in (10) and (17) such that $y_\ell = (y_1, y_2) \to 0$ and $y_m \to 0$ exponentially. Yet this specific choice is in no way optimal for achieving convergence since it forces the outputs to evolve according to pre-specified dynamics that may not be consistent with the natural dynamics of the system. Therefore, we present a method for achieving convergence through control Lyapunov functions that give the desired convergence without explicitly choosing output dynamics. This not only gives optimal convergence (with respect to controllers of minimum norm), but proves much more robust to disturbances. In addition, this control methodology can be converted to a quadratic program wherein additional physical constraints can be added to the controller construction. Additional details on the mathematics behind these constructions can be found in [5,6,11].

It is important to note that there are numerous QP-based formulations of feedback control laws. A prime example is model predictive control (MPC) [8,9,19,33], but other methods include LQR-trees [32], whole-body control methods that enforce constraints through LQPs and QPs [25,24], and QPs for dynamic balancing [30,31], to name only a few. The fundamental differentiator between existing methods and the proposed method is that control objectives are represented by inequality constraints through CLFs, allowing them to be dynamically balanced with each other and with physical constraints.

5.1 Quadratic Programs for Locomotion

The IO feedback controller results in dynamics of the form given in (16). Therefore, if we define the vector $\eta = (y_1, y_2, \dot{y}_2) \in \mathbb{R}^{2n_\ell - 1}$, (16) can be equivalently written as a linear control system:

$$\dot{\eta} = \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & I \\ 0 & 0 \end{bmatrix}}_{F} \eta + \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & I \end{bmatrix}}_{G} \mu. \tag{18}$$

In the context of this control system, we can consider the continuous time algebraic Riccati equations (CARE):

$$F^T P + PF - PGG^T P + Q = 0 \tag{19}$$

for $Q = Q^T > 0$ with solution $P = P^T > 0$. One can use P to construct a *exponentially stabilizing control Lyapunov function (ES-CLF)* that can be used to stabilize the output dynamics (18) exponentially [6]. It is important to note that if we wish to exponentially stabilize at a rate $\epsilon > 0$, we would instead construct a *rapidly* exponentially stabilizing control Lyapunov function (RES-CLF) as detailed in [6]; this can be easily achieved utilizing P, but we forgo the construction in this paper for simplicity of exposition—all presented results apply equally to ES-CLFs and RES-CLFs.

Defining $V(\eta) = \eta^T P \eta$, it is easy to verify that this is a ES-CLF. In particular, it follows that

$$\dot{V}(\eta) = L_f V(\eta) + L_g V(\eta)\mu$$

with

$$\begin{aligned} L_f V(\eta) &= \eta^T (F^T P + PF)\eta, \\ L_g V(\eta) &= 2\eta^T PG. \end{aligned}$$

The goal is exponentially stabilize η to zero. In other words, we wish to find μ such that:

$$L_f V(\eta) + L_g V(\eta)\mu \leq -\gamma V(\eta)$$

for some $\gamma > 0$. In addition to simply satisfying this inequality, we could search for μ that does this in an optimal fashion:

$$m(\eta) = \operatorname{argmin}\{\|\mu\| : \psi_0(\eta) + \psi_1^T(\eta)\mu \leq 0\} \tag{20}$$

where

$$\begin{aligned} \psi_0(\eta) &= L_f V(\eta) + \gamma V(\eta), \\ \psi_1(\eta) &= L_g V(\eta)^T. \end{aligned} \tag{21}$$

The controller $m(\eta)$ that minimizes the control effort required to achieve exponential convergence is termed the *min-norm* controller [10], and can be stated in close form as:

$$m(\eta) = \begin{cases} -\frac{\psi_0(\eta)\psi_1(\eta)}{\psi_1(\eta)^T \psi_1(\eta)} & \text{if } \psi_0(\eta) > 0 \\ 0 & \text{if } \psi_0(\eta) \leq 0 \end{cases}$$

While the min-norm controller, m, can be computed in closed form, it is important to note that this closed form solution is the solution to the quadratic program (QP):

$$\begin{aligned} m(q,\dot{q}) = \underset{\mu \in \mathbb{R}^{n_\ell}}{\operatorname{argmin}} \quad & \mu^T \mu && (22)\\ \text{s.t.} \quad & \psi_0(q,\dot{q}) + \psi_1^T(q,\dot{q})\mu \leq 0 && \text{(CLF)} \end{aligned}$$

where the the optimization problem is now expressed in terms of $(q,\dot{q})$ since η is a function of $(q,\dot{q})$. The end result of solving this QP is the control law for locomotion:

$$u(q,\dot{q}) = A^{-1}(q,\dot{q})(-L_f(q,\dot{q}) + m(q,\dot{q})).$$

There are numerous advantages to this formulation of the problem, some of which will be developed throughout the rest of this section. Yet the most immediate advantage, as first discovered in [11], is that torque bounds can be directly implemented in this formulation where, as opposed to thresholding, the optimal control value that respects the torque bounds can be found. This is achieved by relaxing the constraints (CLF) and penalizing for this relaxation. In particular we consider the locomotion quadratic program (where we now suppress the dependence of functions on $(q,\dot{q})$) first formulated in [11]:

$$\begin{aligned} \underset{(\delta,\mu) \in \mathbb{R}^{n_\ell+1}}{\operatorname{argmin}} \quad & p\delta^2 + \mu^T\mu && \text{(L-QP)}\\ \text{s.t.} \quad & \psi_0 + \psi_1^T \mu \leq \delta && \text{(CLF)}\\ & A^{-1}(-L_f + \mu) \leq u_{\max} && \text{(Max Torque)}\\ & -A^{-1}(-L_f + \mu) \leq u_{\max} && \text{(Min Torque)} \end{aligned}$$

where $p > 0$ is a large value that penalizes violations of the CLF constraint, and $u_{\max}$ are maximum torque values (in vector form).

5.2 Quadratic Programs for Locomotion and Manipulation

The method for obtaining control laws through CLFs, and specifically QPs, can be easily extended to include mobility tasks, even in the case of manipulation tasks that are not necessarily consistent with the mobility tasks. Before developing this, it is necessary to discuss how mobility tasks are unified with locomotion in the context of IO control.

Given the set of outputs associated with manipulation, y_m, we can differentiate these outputs twice and combine with (7) to obtain:

$$\begin{bmatrix} \dot{y}_1 \\ \ddot{y}_2 \\ \ddot{y}_m \end{bmatrix} = \underbrace{\begin{bmatrix} L_f y_1(q,\dot{q}) \\ L_f^2 y_2(q,\dot{q}) \\ L_f^2 y_m(q,\dot{q}) \end{bmatrix}}_{L_f} + \underbrace{\begin{bmatrix} L_g y_1(q,\dot{q}) \\ L_g L_f y_2(q,\dot{q}) \\ L_g L_f y_m(q,\dot{q}) \end{bmatrix}}_{A} u. \qquad (23)$$

Assuming that A is invertible, we can again utilize the controller $u = A^{-1}(-L_f + \mu)$ to obtain equations of the form:

$$\begin{aligned} \dot{\eta}_\ell &= F_\ell \eta_\ell + G_\ell \mu_\ell, \\ \dot{\eta}_m &= F_m \eta_m + G_m \mu_m, \end{aligned}$$

where $\eta_\ell = (y_1, y_2, \dot{y}_2) \in \mathbb{R}^{2n_\ell - 1}$, $\eta_m = (y_m, \dot{y}_m) \in \mathbb{R}^{2n_m}$, and $\mu = (\mu_\ell, \mu_m) \in \mathbb{R}^{n_\ell + n_m}$. For each of these linear control systems, we can construct control Lyapunov functions $V_\ell(\eta_\ell) = \eta_\ell^T P_\ell \eta_\ell$ and $V_m(\eta_m) = \eta_m^T P_m \eta_m$ by solving the CARE (19) for (F_ℓ, G_ℓ) and (F_m, G_m), respectively. This results in the CLF inequalities:

$$\begin{aligned} \psi_0^\ell + (\psi_1^\ell)^T \mu_\ell \leq 0 \\ \psi_0^m + (\psi_1^m)^T \mu_m \leq 0 \end{aligned}$$

defined as in (20). The top inequality ensures exponential convergence of the locomotion outputs (tasks), and the bottom inequality ensures exponential convergence of the manipulation outputs (tasks).

Since the manipulation tasks may not be consistent with the locomotion tasks, A may not necessarily be invertible. Because it may not be possible to calculate u from μ as in (8), we can convert the QPs (22) and (L-QP) into QPs that are functions of u and do not require A to be inverted. By noting that $Au = -L_f + \mu$ it follows that:

$$\mu^T \mu = u^T A^T A u + 2 L_f^T A u + L_f^T L_f.$$

Therefore, the quadratic program combining locomotion and manipulation is expressed as:

$$\begin{aligned} \underset{(\delta_\ell, \delta_m, u) \in \mathbb{R}^{n+2}}{\operatorname{argmin}} \quad & p_\ell \delta_\ell^2 + p_m \delta_m^2 + u^T A^T A u + 2 L_f^T A u && \text{(L+M-QP)} \\ \text{s.t.} \quad & \psi_0^\ell + (\psi_1^\ell)^T (A^\ell u + L_f) \leq \delta_\ell && \text{(Locomotion CLF)} \\ & \psi_0^m + (\psi_1^m)^T (A^m u + L_f) \leq \delta_m && \text{(Manipulation CLF)} \end{aligned}$$

where $p_\ell, p_m > 0$ are penalties for CLF violations and

$$A^\ell = \begin{bmatrix} L_g y_1(q, \dot{q}) \\ L_g L_f y_2(q, \dot{q}) \end{bmatrix}, \qquad A^m = \begin{bmatrix} L_g L_f y_m(q, \dot{q}) \end{bmatrix}.$$

There are numerous important advantages to this representation of a whole-body controller:

- It does not require A to be invertible. Therefore, conflicting (or inconsistent) locomotion and manipulation tasks can be defined in this formulation.
- In the case of inconsistent locomotion and manipulation tasks, there are two CLFs—one for locomotion and one for manipulation. These are each relaxed, so in the case of conflicting constraints where it is not possible to achieve both tasks at the same time, one can adjust the penalty values to prioritize one task over the other.

- Note that removing the relaxations ensures simultaneous exponential convergence of both tasks, but can result in infeasibility of the QP.
- Torque bounds can easily be added to this QP (as in (L-QP)). In this case, the constraints to (L+M-QP) become:

$$u \leq u_{max} \qquad \text{(Max Torque)}$$
$$-u \leq u_{max} \qquad \text{(Min Torque)}$$

- Finally, this representation of the controller allows for force-based tasks and multi-contact to be added to the QP as will be discussed in the next section.

5.3 Quadratic Programs with Null-Space Control

Null-space control methods can be easily subsumed into the whole-body QP presented in (L+M-QP). Using the notation of the previous section, we have:

$$u = J_\ell^T u_\ell + N_\ell^T J_m^T u_m.$$

Therefore, the equations of motion (1) can be written as

$$D(q)\ddot{q} + H(q,\dot{q}) = \underbrace{\begin{bmatrix} BJ_\ell^T & BN_\ell^T J_m^T \end{bmatrix}}_{\widehat{B}(q)} \underbrace{\begin{bmatrix} u_\ell \\ u_m \end{bmatrix}}_{\widehat{u}}.$$

In this case, one obtains new equations of motion:

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

where f is as originally defined in (2) and

$$\widehat{g}(q,\dot{q}) = \begin{bmatrix} 0 \\ D^{-1}(q)\widehat{B}(q) \end{bmatrix}. \tag{24}$$

This results in $\widehat{A}$ as obtained in (23), but with g replaced by $\widehat{g}$. The quadratic program (L+M-QP) can thus be utilized by replacing A with $\widehat{A}$.

It is important to note that this formulation has both advantages and disadvantages. The advantage is that it limits the manipulation tasks so that they only act in the null-space of the locomotion task thus preventing conflicts. This is also the disadvantage—it does not allow for dynamic weighting of the tasks as can be achieved through control Lyapunov functions in the case when null-space projections are not used.

6 Force-Based Multi-contact Tasks

Extending beyond locomotion and manipulation, it is necessary to consider controllers that are able to accomplish force-based multi-contact tasks. In the context of traditional nonlinear control methods such as IO linearization, the fact that this results in over-actuation requires the outputs to be explicitly chosen so that they do not conflict with the external forces being applied to and by the system. In addition, since the number of outputs is necessarily less than the degrees of actuation, a priori optimization is needed to distribute the torques [16]. This is non-ideal from both a control and implementation perspective. Therefore, motivated by existing methods for force-based control [17,31,7,21], we present a method for handling force-based multi-contact tasks directly through the QP based formulation presented in this paper. In particular, rather than first constraining the dynamics based upon holonomic constraints representing mult-contact, we consider the unconstrained dynamics and allow for the dynamics to be constrained in the QP—simultaneously, we enforce CLF associated with locomotion and manipulation. This allows for a holistic approach to control (for locomotion and manipulation), multi-contact and force-based tasks, all of which can be formulated in a single QP.

6.1 Contact Constraints

Consider a vector of holonomic constraints: $h(q) = 0$, with $h(q) \in \mathbb{R}^{n_c}$. Defining the Jacobian $J_h(q) = \frac{\partial h(q)}{\partial q}$, the holonomic constraints are enforced through constraint (or contact) forces $F \in \mathbb{R}^{n_c}$ which are enforced through the dynamics via:

$$D(q)\ddot{q} + H(q,\dot{q}) = Bu + J_h^T F, \tag{25}$$

where $J_h^T F$ projects the contact wrench into joint-space coordinates. Note that in this case, the dynamics describe the "unpinned" model, i.e., they are expressed in terms of generalized coordinates (unlike the previous case in which the dynamics were implicitly given in body (or joint) coordinates where the constraint forces where a priori assumed to be satisfied).

For the constraint forces, F, to be valid, they must satisfy the following equalities and inequalities:

$$D(q)\ddot{q} + H(q,\dot{q}) = Bu + J_h^T F, \tag{26}$$

$$\dot{J}_h\dot{q} + J_h\ddot{q} = 0, \tag{27}$$

$$\mathcal{A}(F) \geq 0, \tag{28}$$

where $\mathcal{A}(F) \in \mathbb{R}^{n_a}$ is a set of *admissibility* constraints on the reaction wrench [13] which ensure physical validity of the model, e.g. positive normal force and friction constraints.

Traditionally, in modeling the robotic system, $\ddot{q}$ in (26) is explicitly solved and substituted into (27), yielding:

$$\dot{J}_h\dot{q} + J_h D(q)^{-1}(Bu + J_h^T F - H(q,\dot{q})) = 0. \tag{29}$$

Rearranging terms gives an explicit expression for the constraint forces:

$$F = (J_h D(q)^{-1} J_h^T)^{-1}(J_h D(q)^{-1}(H(q,\dot{q}) - Bu) - \dot{J}_h\dot{q}) \tag{30}$$

which can be substituted back into (25) to yield the constrained dynamical system. This method ensures that (26) and (27) are satisfied for all control inputs u; however, there is no guarantee that either (28) is satisfied, i.e., it implicitly assumes that the constraint forces are valid.

6.2 Quadratic Program Formulation

With a view toward formulating contacts and force-based tasks in the QP framework presented, the constrained equations of motion (25) can be written as:

$$D(q)\ddot{q} + H(q,\dot{q}) = \underbrace{\left[B \ J_h^T \right]}_{\overline{B}(q)} \underbrace{\begin{bmatrix} u \\ F \end{bmatrix}}_{\overline{u}}. \tag{31}$$

Noting that (31) takes the same general form of the Euler-Lagrange equations as (1), $\overline{A}$ can be calculated utilizing $\overline{g}$ obtained in a similar fashion to (24). The end result is the input/output relationship

$$\overline{A}\overline{u} = (-L_f + \mu). \tag{32}$$

This allows for the Contact Force QP formulation given by

$$\begin{aligned}
\underset{(\delta,\overline{u})\in\mathbb{R}^{n+n_c+1}}{\operatorname{argmin}} \quad & p\delta^2 + \overline{u}^T\overline{A}^T\overline{A}\overline{u} + 2L_f^T\overline{A}\overline{u} && \text{(CF-QP)} \\
\text{s.t.} \quad & \dot{J}_h\dot{q} + J_h D(q)^{-1}(\overline{B}\overline{u} - H(q,\dot{q})) = 0 && \text{(Constrained Dynamics)} \\
& \psi_0 + \psi_1^T(\overline{A}\overline{u} + L_f) \le \delta && \text{(CLF)} \\
& \mathcal{A}(F) \ge 0 && \text{(Contact Force)} \\
& F = F^d(t,q,\dot{q}) && \text{(Desired Force)}
\end{aligned}$$

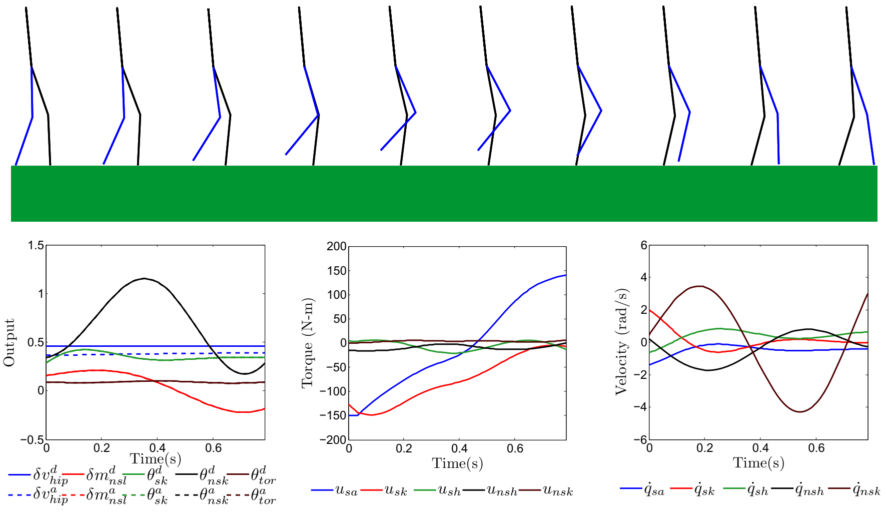
The solution to this quadratic program, $\overline{u}^*$, is a set of actuator torques u^* and manipulator contact forces F^* that satisfy the constrained dynamic equations, (Constrained Dynamics), while guaranteeing that the relative degree one and two outputs converge exponentially (CLF), *and* the relative degree zero, i.e. force-based, task F^d is performed (Desired Force) in a way that that is consistent with the required contact forces (Contact Force).

There are some important points that should be made regarding this QP for force-based multi-contact control:

- Note that one can remove the constraint (Desired Force) if the only goal is to remain in contact with the environment.
- Compliant force control can be achieved through specific choices of F^d.
- Torque constraints can easily be added to the QP through the constraints:

$$\begin{aligned}
u &\le u_{max} && \text{(Max Torque)} \\
-u &\le u_{max} && \text{(Min Torque)}
\end{aligned}$$

- Additional CLFs can be added for additional manipulation tasks as in the case of (L+M-QP).



(a) Outputs (b) Torques (c) Velocities

Fig. 1. Simulation results from one step of a steady-state robotic walking gait using human-inspired control via the quadratic program (L-QP)

7 Simulation Results

To demonstrate the results presented in this paper, we will apply them through a series of simulation results of progressive complexity. In particular, we will apply the three main QPs constructed in this paper: (L-QP), (L+M-QP) and (CF-QP). We will begin with a lower-body locomotion controller implemented through (L-QP). This controller will then be embedded into a whole-body robot model (without any knowledge of this whole-body model) and coupled with a manipulation controller through (L+M-QP). The robustness of this combined controller will be tested through rough terrain locomotion, where there is no knowledge of the terrain. Finally, these preceding controllers will be combined with a force-based contact task through (CF-QP).

Lower-body locomotion: Locomotion controllers were first obtained for the lower body using human-inspired control through the methods outlined in [1,2,4]. The walking gait, and associated outputs, were then used to find a CLF through the methods given in [5]. This was then implemented through the QP (L-QP) with torque bounds of 150 Nm. The results are shown in Fig. 1. In particular, the walking gait is shown along with the actual and desired controller output profiles (a), joint torques (b) and velocities (c). Note that the relative degree two outputs are almost exactly tracked; the exception is at the beginning of the gait when it is necessary to relax the CLF condition in order to satisfy the torque bounds.

Whole-body locomotion: The locomotion controller, and associated CLF, was then embedded on the whole-body robotic model. On the upper body, outputs were chosen that keep the arms at the robot's side, i.e., in (5) the actual (position

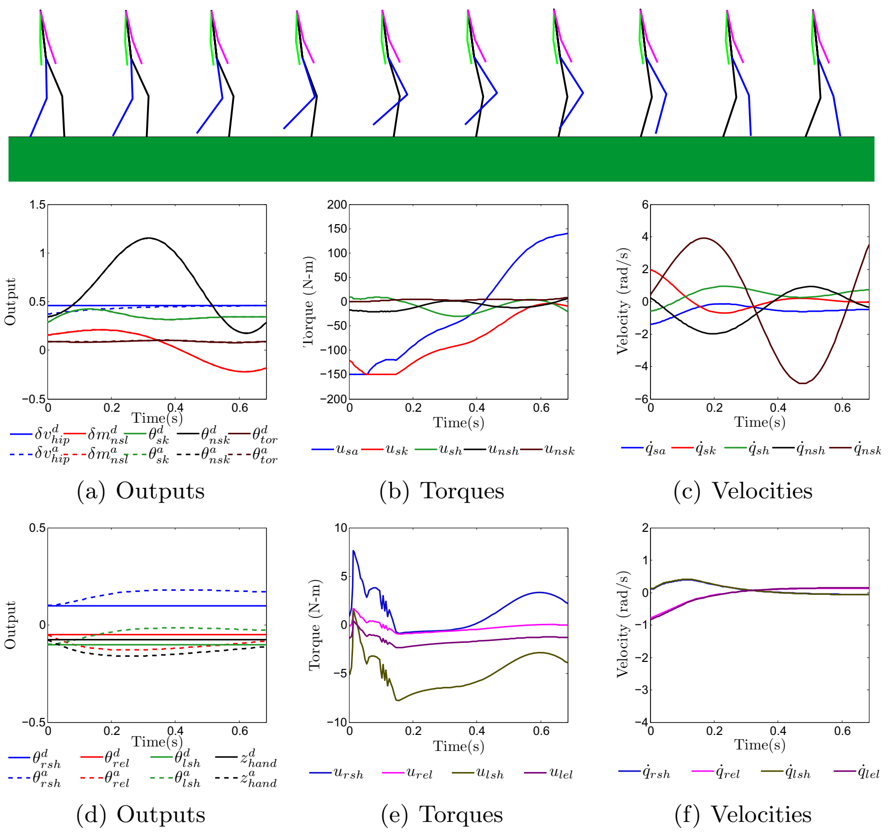


Fig. 2. Simulation results from one step of robotic walking in which a human-inspired locomotion controller is embedded in the full body of a humanoid via the quadratic program (L+M-QP)

modulating) outputs are the angles of the upper body and the desired outputs are small constant values, and the associated CLF was calculated. The controllers were then integrated through the QP (L+M-QP) in which the locomotion was given a higher priority while the manipulation task was given a low priority (through the choice of penalties p_l and p_m; that is $p_l >> p_m$). In addition, a max torque constraint of 150 Nm was enforced. The end result is that the robot walks, and the arms naturally swing to provide greater stability for the locomotion task. The results are shown in Fig. 2, including: actual and desired controller output profiles for locomotion (a), joint torques (b), and velocities (c) of the lower body; actual and desired controller output profiles for manipulation (d), joint torques (e) and velocities (f) of the upper body. In this case, slight deviations from the desired outputs are seen for the lower-body outputs while large deviations are seen for the upper-body outputs due to the prioritization of the associated CLFs.

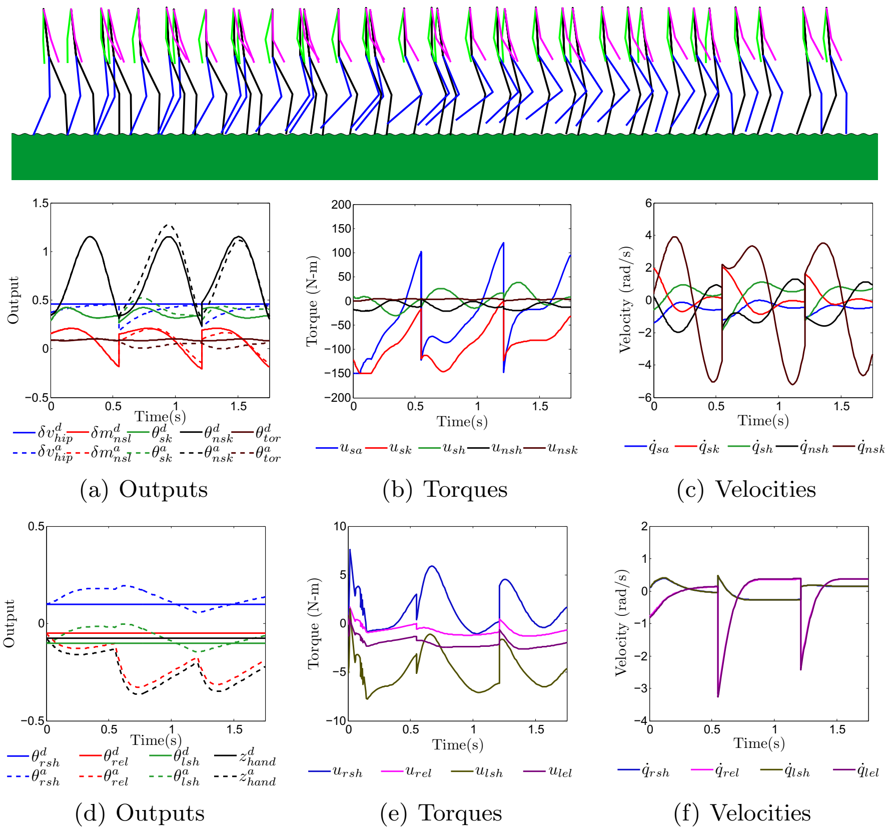
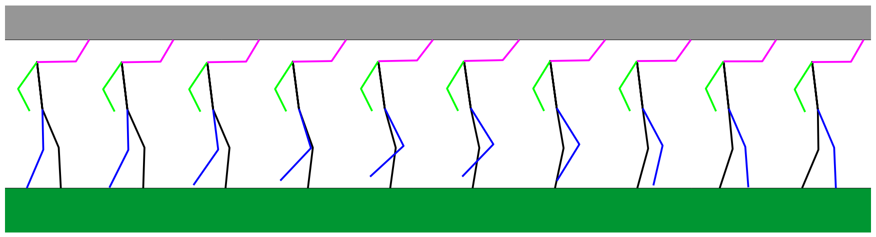
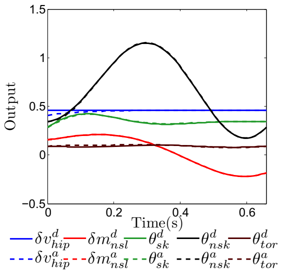


Fig. 3. Simulation results displaying robustness: three steps of robotic walking in which a human-inspired locomotion controller is embedded in the full body of a humanoid walking over—and without knowledge of—sinusoidally varying terrain (1cm peak amplitude and $\frac{2\pi}{50}$cm period)

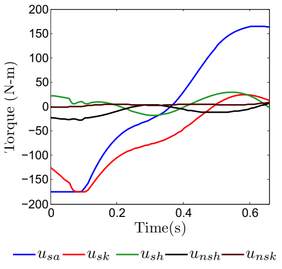
Robustness of whole-body locomotion: To demonstrate the robustness of the control method, we consider rough terrain and solve the same QP that was considered for whole-body locomotion. In this case, the robot has no knowledge of the terrain, so the controller must dynamically compensate. The robot is able to do so by swinging its arms more (as can be seen through the drift in the upper-body outputs shown in Fig. 3(d)) to maintain stability of the locomotion task; this is all done dynamically through the QP, without the user specifying this behavior. The results of this can be seen in Fig. 3 which plots the results for three steps.

Whole-body locomotion with force-based task: As a final test of the control method, we now consider the case when we wish to locomote and perform a force-based task; in this case, we want the robot to push against the ceiling with a sinusoidal force. We achieve this by solving the QP (CF-QP) with F^d being a time-based

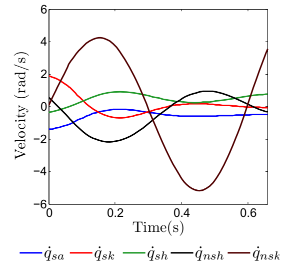



Output
Time(s)
δv^d_{hip} δm^d_{nsl} θ^d_{sk} θ^d_{nsk} θ^d_{tor}
δv^a_{hip} δm^a_{nsl} θ^a_{sk} θ^a_{nsk} θ^a_{tor}

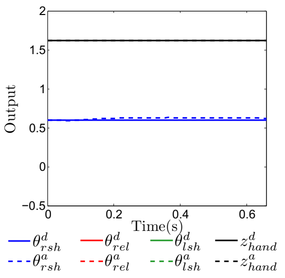
(a) Outputs



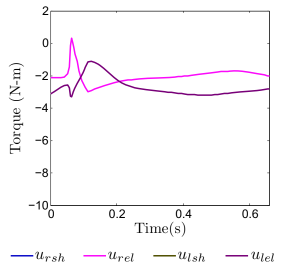
(b) Torques



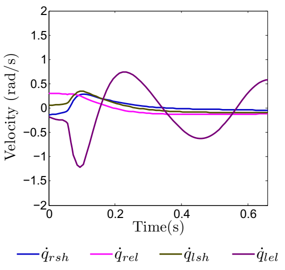
(c) Velocities



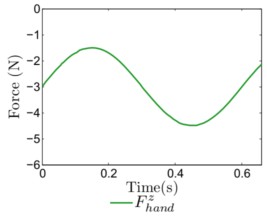
(d) Outputs



(e) Torques



(f) Velocities



(g) Force Control

Fig. 4. Simulation results demonstrating locomotion together with force-based manipulation. The manipulation controller applies a sinusoidal force (profile shown in (g)) to the ceiling while regulating the remaining arm joints. Locomotion and manipulation controllers are combined in a single quadratic program (CF-QP) with control Lyapunov function constraints enforcing contact between the left hand and the ceiling while also specifying a desired applied force to the ceiling, strict convergence for locomotion outputs and relaxed convergence in upper body joint outputs, and constraints on the admissible joint torques.

sinusoid describing the desired force to exert on the ceiling. The end results of this approach can be seen in Fig. 4 where the locomotion and force-based tasks are shown to be simultaneously achieved.

8 Conclusion

This paper presented the first step toward unifying locomotion, manipulation and force-based tasks into a single framework—quadratic programs utilizing control Lyapunov functions. The end result was a single quadratic program that can dynamically balance all of these disparate objectives through weighted inequality constraints. The construction of this QP was motivated theoretically and demonstrated through simulation, with the end result being locomotion, manipulation and force-based control on a simplified humanoid robot. The presented results potentially have important ramifications for robotic cyber-physical systems. The QP can be implemented as a single algorithm that includes both controllers and the interaction of the robot with the physical world. This could allow for more holistic implementation of controllers on physical systems, thus permitting for a more complete understanding of their behavior and proofs of their correctness.

On both a practical and theoretic level, there are numerous areas in which to further explore the concepts presented. Practically, the CLF based QP formulation has been implemented in real-time to experimentally achieve 2D bipedal robotic walking [11] (with control rates exceeding 1 kHz utilizing embedded optimization methods [18]), and similar benchmarks have been achieved using the formalism presented in this paper for 3D walking robots. Yet, the speed of the QP depends on the feasibility of the inequality and equality constraints, so understanding the interplay between computation time of the QP and feasibility of the constraints is an interesting problem. This naturally motivates theoretic research questions related to the CLF based QP formulation. In particular, the CLF inequality constraints were relaxed to allow for solvability of the QP in the presence of hard constraints like torque bounds—yet these relaxations can result in drift in the control objectives and, for aggressive torque bounds, can result in loss of convergence. Conversely, if the relaxations are removed, guarantees on convergence can be made but the QP may become infeasible. Understanding this interplay between proofs of correctness, solvability of the QP, and speed of controllers running CLF based QPs form the basis for a variety of interesting theoretic questions. All of these questions are deeply rooted in core problems related to robotic CPSs and, therefore, promise to be fruitful areas of research.

Acknowledgements. The authors would like to thank Jessy Grizzle, Kevin Galloway and Koushil Sreenath for the many discussions on CLFs, and especially for the collaboration that motivated the constructions in this paper. We would also like to thank John Yamokoski and Nicolaus Radford for encouraging the development of methods for unifying locomotion and manipulation. Finally, we would like to thank the reviewers for their helpful comments and suggestions.

References

1. Ames, A.D.: First steps toward automatically generating bipedal robotic walking from human data. In: Kozłowski, K. (ed.) Robot Motion and Control 2011. LNICS, vol. 422, pp. 89–116. Springer, Heidelberg (2012)
2. Ames, A.D.: First steps toward underactuated human-inspired bipedal robotic walking. In: IEEE International Conference on Robotics and Automation, St. Paul, MN (2012)
3. Ames, A.D.: Human-inspired control of bipedal walking robots. To appear in the IEEE Trans. Automatic Control (2013)
4. Ames, A.D., Cousineau, E.A., Powell, M.J.: Dynamically stable robotic walking with NAO via human-inspired hybrid zero dynamics. In: Hybrid Systems: Computation and Control, Beijing (2012)
5. Ames, A.D., Galloway, K., Grizzle, J.W.: Control Lyapunov functions and hybrid zero dynamics. In: Proc. 51st IEEE Conf. Decision and Control (2012)
6. Ames, A.D., Galloway, K., Grizzle, J.W., Sreenath, K.: Rapidly exponentially stabilizing control Lyapunov runctions and hybrid zero dynamics. To appear in IEEE Trans. Automatic Control (2013)
7. Anitescu, M., Potra, F.A.: Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. Nonlinear Dynamics 14, 231–247 (1997)
8. Bemporad, A., Morari, M.: Robust model predictive control: A survey. Robustness in Identification and Control 245, 207–226 (1999)
9. Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.N.: The explicit solution of model predictive control via multiparametric quadratic programming. In: Proceedings of the American Control Conference (2012)
10. Freeman, R.A., Kokotović, P.V.: Robust Nonlinear Control Design. Birkhäuser (1996)
11. Galloway, K., Sreenath, K., Ames, A.D., Grizzle, J.W.: Torque saturation in bipedal robotic walking through control lyapunov function based quadratic programs. CoRR, abs/1302.7314 (2013)
12. Grizzle, J.W., Abba, G., Plestan, F.: Asymptotically stable walking for biped robots: Analysis via systems with impulse effects. IEEE Transactions on Automatic control 46(1), 51–64 (2001)
13. Grizzle, J.W., Chevallereau, C., Ames, A.D., Sinnet, R.W.: 3D bipedal robotic walking: models, feedback control, and open problems. In: IFAC Symposium on Nonlinear Control Systems, Bologna (September 2010)
14. Khatib, O.: A unified approach for motion and force control of robot manipulators: The operational space formulation. IEEE Journal of Robotics and Automation 3, 43–53 (1987)
15. Khatib, O., Sentis, L., Park, J., Warren, J.: Whole-body dynamic behavior and control of human-like robots. International Journal of Humanoid Robotics 1, 29–43 (2004)
16. Kolavennu, S., Palanki, S., Cockburn, J.C.: Nonlinear control of nonsquare multivariable systems. Chemical Engineering Science 56, 2103–2110 (2001)
17. Lee, S.H., Goswami, A.: A momentum-based balance controller for humanoid robots on non-level and non-stationary ground. Autonomous Robots 33(4), 399–414 (2012)
18. Mattingley, J., Boyd, S.: Cvxgen: a code generator for embedded convex optimization. Optimization and Engineering 13(1), 1–27 (2012)

19. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: Stability and optimality. Automatica 36, 789–814 (2000)
20. Murray, R.M., Li, Z., Sastry, S.S.: A Mathematical Introduction to Robotic Manipulation. Boca Raton (1994)
21. Oppenheimer, M.W., Doman, D.B., Bolender, M.A.: Dynamic balance force control for compliant humanoid robots. In: 14th Mediterranean Conference on Control and Automation, MED 2006, pp. 1–6 (2006)
22. Powell, M., Hereid, A., Ames, A.D.: Speed regulation in 3D robotic walking through motion transitions between human-inspired partial hybrid zero dynamics. To appear in the IEEE International Conference on Robotics and Automation (2013)
23. Powell, M.J., Zhao, H., Ames, A.D.: Motion primitives for human-inspired bipedal robotic locomotion: Walking and stair climbing. In: IEEE International Conference on Robotics and Automation, St. Paul, MN (2012)
24. Saab, L., Ramos, O.E., Keith, F., Mansard, N., Soueres, P., Fourquet, J.-Y.: Dynamic whole-body motion generation under rigid contacts and other unilateral constraints. IEEE Transactions on Robotics 29(2), 346–362 (2013)
25. Salini, J., Padois, V., Bidaud, P.: Synthesis of complex humanoid whole-body behavior: A focus on sequencing and tasks transitions. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 1283–1290 (2011)
26. Sastry, S.S.: Nonlinear Systems: Analysis, Stability and Control. Springer (1999)
27. Siciliano, B., Slotine, J.J.E.: A general framework for managing multiple tasks in highly redundant robotic systems. In: Fifth International Conference on Advanced Robotics, ICAR (1991)
28. Sontag, E.: A 'universal' contruction of Artstein's theorem on nonlinear stabilization. Systems & Control Letters 13, 117–123 (1989)
29. Srinivasan, S., Raptis, I.A., Westervelt, E.R.: Low-dimensional sagittal plane model of normal human walking. ASME Journal of Biomechanical Engineering 130(5) (2008)
30. Stephens, B.J., Atkeson, C.G.: Push recovery by stepping for humanoid robots with force controlled joints. In: IEEE International Conference on Humanoid Robots (2010)
31. Stephens, B.J., Atkeson, C.G.: Dynamic balance force control for compliant humanoid robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS (2010)
32. Tedrake, R., Manchester, I.R., Tobenkin, M., Roberts, J.W.: LQR-trees: Feedback motion planning via sums of squares verification. International Journal of Robotics Research 29, 1038–1052 (2010)
33. Wang, Y., Boyd, S.: Fast model predictive control using online optimization. IEEE Transations on Control Systems Technology 18(2), 267–278 (2010)
34. Westervelt, E.R., Grizzle, J.W., Chevallereau, C., Choi, J.H., Morris, B.: Feedback Control of Dynamic Bipedal Robot Locomotion, Boca Raton (June 2007)
35. Nadubettu Yadukumar, S., Pasupuleti, M., Ames, A.D.: From formal methods to algorithmic implementation of human inspired control on bipedal robots. In: Tenth International Workshop on the Algorithmic Foundations of Robotics (WAFR), Boston, MA (2012)