An efficiently solvable quadratic program for stabilizing dynamic locomotion

Citation: Kuindersma, Scott, Frank Permenter, and Russ Tedrake. "An Efficiently Solvable Quadratic Program for Stabilizing Dynamic Locomotion." 2014 IEEE International Conference on Robotics and Automation (ICRA) (May 2014).

As Published: http://dx.doi.org/10.1109/ICRA.2014.6907230

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Persistent URL: http://hdl.handle.net/1721.1/90913

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

An Efficiently Solvable Quadratic Program for Stabilizing Dynamic Locomotion

Scott Kuindersma, Frank Permenter, and Russ Tedrake

***Abstract*— We describe a whole-body dynamic walking controller implemented as a convex quadratic program. The controller solves an optimal control problem using an approximate value function derived from a simple walking model while respecting the dynamic, input, and contact constraints of the full robot dynamics. By exploiting sparsity and temporal structure in the optimization with a custom active-set algorithm, we surpass the performance of the best available off-the-shelf solvers and achieve 1kHz control rates for a 34-DOF humanoid. We describe applications to balancing and walking tasks using the simulated Atlas robot in the DARPA Virtual Robotics Challenge.**

I. Introduction

Achieving dynamically-stable locomotion in complex legged systems is a problem at the heart of modern robotics research. For humanoid systems in particular, nonlinear, underactuated, and high-dimensional dynamics conspire to make the control problem challenging. Optimization-based techniques must simultaneously reason about the dynamics, actuation limits, and contact constraints of the walking system. Model predictive control (MPC) is a popular approach to performing this type of constrained optimization iteratively over fixed horizons, but its computational complexity has hindered applications to high-dimensional systems. Furthermore, the hybrid dynamics of walking robots makes multi-step optimization difficult [1]. Successful examples of using MPC for humanoid control have therefore relied upon the use of low-dimensional linear models [2], [3] or relaxation of constraints to permit smooth optimization through discontinuous dynamics [4].

Several researchers have recently explored using quadratic programs (QPs) to control bipedal systems by exploiting the fact that the *instantaneous* dynamics and contact constraints can be expressed linearly (effectively solving a horizon-1 MPC problem) [5], [6], [7], [8], [9], [10], [11], [12]. A key observation about these approaches in the context of balancing and locomotion tasks is that, during typical operation, the set of active inequality constraints changes very infrequently between consecutive control steps. We give a problem formulation and solution technique that explicitly take advantage of this observation.

We describe a QP that exploits optimal control solutions for a simple unconstrained model of the walking system. Using time-varying LQR design, we compute the optimal cost-to-go for the simple model and use it as part of the objective function in a constrained optimization to compute inputs for the full robot. We describe the approach concretely in terms of a simulated bipedal system and zero-moment point (ZMP) dynamics. In addition to providing a principled and reliable way to stabilize walking trajectories, we show the resulting QP cost function contains low-dimensional structure that can be exploited to reduce solution time.

To achieve real-time control rates, we designed a custom active-set solver that exploits consistency between subsequent solutions and outperforms the best available off-the-shelf solvers such as CVXGEN and Gurobi by a factor of 5 or more. Our analysis of solver performance during typical walking experiments suggests that the active set remains constant between consecutive control steps approximately 97% of the time, requiring only a *single linear system solve per step*. In our tests, we were able to achieve average control rates of 1kHz for a 34-DOF humanoid. We briefly summarize extensive simulation testing done with the Atlas robot as part of the DARPA Virtual Robotics Challenge.

II. LQR Design for ZMP Dynamics

The planar center of mass (COM) and ZMP dynamics of a fully actuated rigid body system can be written in state space form as

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ &= \begin{bmatrix} 0 & \mathbf{I} \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix} \mathbf{u} \qquad (1) \\ \mathbf{y} &= \mathbf{C}\mathbf{x} - b(\mathbf{x}, \dot{\mathbf{x}})\mathbf{u} \\ &= \begin{bmatrix} \mathbf{I} & 0 \end{bmatrix} \mathbf{x} - \frac{z_{\mathrm{com}}}{\ddot{z}_{\mathrm{com}} + g} \mathbf{I}\mathbf{u}, \qquad (2) \end{aligned}$$

where $\mathbf{x} = [x_{\mathrm{com}}, y_{\mathrm{com}}, \dot{x}_{\mathrm{com}}, \dot{y}_{\mathrm{com}}]^T$, $\mathbf{u} = [\ddot{x}_{\mathrm{com}}, \ddot{y}_{\mathrm{com}}]^T$, $\mathbf{y} = [x_{\mathrm{zmp}}, y_{\mathrm{zmp}}]^T$, g is a constant gravitational acceleration, and z_{com} is the COM height. The ZMP is a well-studied quantity in the bipedal walking literature that defines the point on the ground plane at which the moment produced by inertial and gravitational forces is parallel to the surface normal (i.e., the robot is not tipping) [13]. Since dynamic balance is achieved when the contact forces directly oppose the gravitational and inertial forces, maintaining the ZMP within the contact support polygon can be an effective strategy for maintaining dynamic stability in legged locomotion.

Given desired ZMP trajectory, $\mathbf{y}^d(t)$, we would like to compute an optimal tracking controller that takes into account the time- and state-varying constraints on $\mathbf{u}$ imposed by the dynamics, input limits, and contacts of the full walking system. Due to the prohibitive computational requirements of

This work was supported by AFRL contract FA8750-12-1-0321 and NSF contract ERC-1028725, IIS-1161909, and IIS-0746194.

The authors are with the Computer Science and Artificial Intelligence Laboratory at the Massachusetts Institute of Technology, Cambridge, MA, USA. {scottk,fpermenter,russt}@csail.mit.edu

solving nonlinearly constrained optimal control problems of this scale, we instead solve an unconstrained time-varying LQR problem to compute the optimal cost-to-go, J^*, which provides a control-Lyapunov function (CLF) for the ZMP dynamics. On each iteration, we select the control inputs to descend this ZMP CLF while reasoning about the instantaneous constraints of the full system.

We begin by specifying a cost functional of the form

$$J = \bar{\mathbf{y}}(t_f)^T \mathbf{Q}_f \bar{\mathbf{y}}(t_f) + \int_0^{t_f} \bar{\mathbf{y}}(t)^T \mathbf{Q} \bar{\mathbf{y}}(t) dt, \tag{3}$$

where the coordinates $\bar{\mathbf{y}}(t) = \mathbf{y}(t) - \mathbf{y}^d(t)$, $\mathbf{Q} \succ 0$, and $\mathbf{Q}_f \succ 0$. In practice the COM height, z_{com}, is often assumed to be constant, making the ZMP dynamics (2) linear [14]. More generally, if the COM height trajectory is constrained to be a known function of time, $(z_{\text{com}}(t), \dot{z}_{\text{com}}(t), \ddot{z}_{\text{com}}(t))$, the ZMP dynamics are time-varying linear,

$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t), \tag{4}$$

and therefore amenable to TVLQR design without explicit linearization.

Solving the Riccati equation yields the optimal cost-to-go for the time-varying linear system,

$$J^*(\bar{\mathbf{x}}, t) = \bar{\mathbf{x}}^T \mathbf{S}(t) \bar{\mathbf{x}} + \mathbf{s}_1(t)^T \bar{\mathbf{x}} + s_0(t),$$

and the linear optimal controller,

$$\begin{aligned} \bar{\mathbf{u}}^* &= -\mathbf{K}(t)\bar{\mathbf{x}} \\ &= \arg\min_{\bar{\mathbf{u}}} \bar{\mathbf{y}}(t)^T \mathbf{Q} \bar{\mathbf{y}}(t) + \left.\frac{\partial J^*}{\partial \bar{\mathbf{x}}}\right|_{\bar{\mathbf{x}}} \dot{\bar{\mathbf{x}}}, \end{aligned} \tag{5}$$

where $\bar{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}^d(t)$ and $\bar{\mathbf{u}}(t) = \mathbf{u}(t) - \mathbf{u}^d(t)$. In general, achieving $\bar{\mathbf{u}}^*$ is not possible due to constraints imposed by the robot dynamics. For example, actuator saturations and contact friction properties can limit the possible magnitudes and directions of COM accelerations. Therefore, to compute control inputs we perform a constrained minimization using

$$V(\bar{\mathbf{x}}, \bar{\mathbf{u}}, t) = \bar{\mathbf{y}}(t)^T \mathbf{Q} \bar{\mathbf{y}}(t) + \left.\frac{\partial J^*}{\partial \bar{\mathbf{x}}}\right|_{\bar{\mathbf{x}}} \dot{\bar{\mathbf{x}}} \tag{6}$$

as a surrogate value function.

III. QP Formulation

Given the stabilizing solution for the ZMP dynamics, we design a QP to solve for control inputs for the full robot dynamics that minimizes (6) and a quadratic motion cost for walking subject to the instantaneous constraints.

Consider the familiar rigid body dynamics,

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{B}(\mathbf{q}, \dot{\mathbf{q}})\boldsymbol{\tau} + \boldsymbol{\Phi}(\mathbf{q})^T \boldsymbol{\lambda}, \tag{7}$$

where $\mathbf{H}(\mathbf{q})$ is the system inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ captures the gravitational and Coriolis terms, $\mathbf{B}(\mathbf{q}, \dot{\mathbf{q}})$ is the control input map, and $\boldsymbol{\Phi}(\mathbf{q})^T$ transforms external forces, $\boldsymbol{\lambda}$, into generalized forces. In our case, $\boldsymbol{\lambda} = [\ \boldsymbol{\lambda}_1^T \quad \dots \quad \boldsymbol{\lambda}_{N_c}^T\ ]^T$ is a vector of ground-contact forces acting at N_c contact points. The set of active contacts are determined by kinematic or force measurement classification at each control step.

For floating-base systems such as humanoids, the dynamics can be partitioned into actuated and unactuated degrees of freedom [9],

$$\begin{aligned} \mathbf{H}_f\ddot{\mathbf{q}} + \mathbf{C}_f &= \boldsymbol{\Phi}_f^T \boldsymbol{\lambda} \\ \mathbf{H}_a\ddot{\mathbf{q}} + \mathbf{C}_a &= \mathbf{B}_a\boldsymbol{\tau} + \boldsymbol{\Phi}_a^T \boldsymbol{\lambda}, \end{aligned} \tag{8}$$

where we have dropped the explicit dependence on $\mathbf{q}, \dot{\mathbf{q}}$ from our notation for conciseness. This separation permits the removal of $\boldsymbol{\tau}$ as a decision variable by including (8) as a constraint expressing $\boldsymbol{\tau}$ in terms of $\ddot{\mathbf{q}}$ and $\boldsymbol{\lambda}$:

$$\boldsymbol{\tau} = \mathbf{B}_a^{-1}\left[\mathbf{H}_a\ddot{\mathbf{q}} + \mathbf{C}_a - \boldsymbol{\Phi}_a^T\boldsymbol{\lambda}\right].$$

We use a standard, conservative polyhedral approximation of the friction cone, $\hat{K}_j$, for each contact point, $\mathbf{c}_j$,

$$\hat{K}_j = \left\{ \sum_{i=1}^{N_d} \beta_{ij}\mathbf{v}_{ij} : \beta_{ij} \geq 0 \right\}. \tag{9}$$

The generating vectors, $\mathbf{v}_{ij}$, are computed as $\mathbf{v}_{ij} = \mathbf{n}_j + \mu_j \mathbf{d}_{ij}$, where $\mathbf{n}_j$ and $\mathbf{d}_{ij}$ are the contact-surface normal and i^{th} tangent vector for the j^{th} contact point, respectively, μ_j is the Coulomb friction coefficient, and N_d is the number of tangent vectors used in the approximation [15].

Given the robot state, $\mathbf{q}, \dot{\mathbf{q}}$, at time t, we solve the following quadratic program:

Quadratic Program 1:

$$\min_{\ddot{\mathbf{q}}, \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\eta}} V(\bar{\mathbf{x}}, \bar{\mathbf{u}}, t) + w_{\ddot{\mathbf{q}}}||\ddot{\mathbf{q}}_{\text{des}} - \ddot{\mathbf{q}}||^2 + \varepsilon \sum_{ij} \beta_{ij}^2 + ||\boldsymbol{\eta}||^2 \tag{10}$$

subject to

$$\begin{aligned} \mathbf{H}_f\ddot{\mathbf{q}} + \mathbf{C}_f &= \boldsymbol{\Phi}_f^T\boldsymbol{\lambda} && (11) \\ \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}} &= -\alpha\mathbf{J}\dot{\mathbf{q}} + \boldsymbol{\eta} && (12) \\ \mathbf{B}_a^{-1}(\mathbf{H}_a\ddot{\mathbf{q}} + \mathbf{C}_a - \boldsymbol{\Phi}_a^T\boldsymbol{\lambda}) &\in [\boldsymbol{\tau}_{\min}, \boldsymbol{\tau}_{\max}] && (13) \\ \forall_{j=\{1\dots N_c\}}\ \boldsymbol{\lambda}_j &= \sum_{i=1}^{N_d} \beta_{ij}\mathbf{v}_{ij} && (14) \\ \forall_{i,j}\beta_{ij} &\geq 0 && (15) \\ \boldsymbol{\eta} &\in [\boldsymbol{\eta}_{\min}, \boldsymbol{\eta}_{\max}]. && (16) \end{aligned}$$

The constraints (11) and (13) ensure that the dynamics and input limits are respected, (12) is a no-slip constraint on the foot contacts requiring that their acceleration be negatively proportional to the velocity, and the constraints (14,15) together ensure that contact forces remain within $\hat{K}$. The parameter vector $\boldsymbol{\eta}$ allows bounded violations of the no-slip constraint to reduce the likelihood of infeasibility, ε is a regularization constant typically set to a small value, e.g., $\varepsilon = 10^{-8}$, and $\mathbf{J} = \partial\mathbf{c}/\partial\mathbf{q}$ is the Jacobian matrix for the vector of all contact points, $\mathbf{c} = [\ \mathbf{c}_1^T \quad \dots \quad \mathbf{c}_{Nc}^T\ ]^T$.

The weight parameter, $w_{\ddot{\mathbf{q}}}$, is used to balance the relative contribution of the desired motion cost with the ZMP tracking cost. To respect joint limits, the bounds $\ddot{q}_i \geq 0$ and $\ddot{q}_i \leq 0$ are added for all i such that $q_i = q_i^{\text{MIN}}$ and $q_i = q_i^{\text{MAX}}$, respectively.

IV. Optimization

We solve QP 1 at each control step using a simple active-set method. The method assumes the set of active inequality constraints remains constant for consecutive solutions. It then produces a candidate solution by solving a partial set of optimality conditions derived from the assumed active set. If the candidate solution satisfies the full set of optimality conditions, the assumption is correct and the algorithm terminates. Otherwise, the method updates the active set and repeats until a solution is found or a maximum number of iterations is reached.

On rare occasions when no solution is found, the algorithm fails over to a more reliable (but on average slower) interior point solver. In our experiments, this lead to infrequent single-step input delays on the order of 3ms, which had no significant effect on walking performance. This contingency is required since finite termination cannot be guaranteed for the proposed method. In practice, however, instances of QP 1 are almost always solved in one iteration. The computational cost of each iteration is also very small. A candidate solution is produced by solving a structured system of linear equations and constraints are evaluated only once.

A. Active-set method

The QP solved at each control step can be written in the standard form,

$$\begin{array}{ll} \min\limits_{\mathbf{z}} & \frac{1}{2}\mathbf{z}^T\mathbf{W}\mathbf{z} + \mathbf{g}^T\mathbf{z} \\ \text{subject to} & \mathbf{A}\mathbf{z} = \mathbf{b} \\ & \mathbf{P}\mathbf{z} \leq \mathbf{f}, \end{array} \tag{17}$$

where the inequalities are defined by $\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n)^T$ and $\mathbf{f} = (f_1, f_2, \ldots, f_n)^T$. To solve this problem, it is assumed that $\mathbf{p}_i^T\mathbf{z} = f_i$ at optimality for each i in a subset $\mathcal{A} \subseteq \{1 \ldots n\}$ called the *active set*. For $t > 0$, this subset equals the indices of the active inequalities from time $t-1$. With this assumption, the KKT conditions for the QP can be written in terms of $\mathbf{z}$, $\boldsymbol{\gamma}$, and $\boldsymbol{\alpha}$:

$$\begin{array}{rcll} \mathbf{W}\mathbf{z} + \mathbf{A}^T\boldsymbol{\alpha} + \sum_{i\in\mathcal{A}} \gamma_i\mathbf{p}_i &=& -\mathbf{g} & \\ \mathbf{A}\mathbf{z} &=& \mathbf{b} & \\ \mathbf{p}_i^T\mathbf{z} &=& f_i & \forall i \in \mathcal{A} \\ \gamma_i &=& 0 & \forall i \neq \mathcal{A} \end{array} \tag{18}$$

$$\begin{array}{rcll} \mathbf{P}\mathbf{z} &\leq& \mathbf{f} & \\ \gamma_i &\geq& 0 & \forall i \in \mathcal{A}. \end{array} \tag{19}$$

Our method solves the linear equations (18) and checks if the solution $(\mathbf{z}, \boldsymbol{\gamma}, \boldsymbol{\alpha})$ satisfies the inequalities (19). If the inequalities are satisfied, $\mathbf{z}$ solves the QP and the algorithm terminates. Otherwise, the algorithm adds index i to $\mathcal{A}$ if $\mathbf{p}_i^T\mathbf{z} > f_i$ or removes index i if $\gamma_i < 0$ and resolves (18). The algorithm repeats this process until the inequalities (19) are satisfied or a until a specified maximum number of iterations is reached. The method is outline in Algorithm 1.

B. Efficiently computing a candidate solution

The structure of QP 1 admits an efficient solution of the linear system (18). In particular, one can cheaply compute $\mathbf{W}^{-1}$ and construct a smaller system for $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$. Using a solution to this smaller system, one can then easily recover $\mathbf{z}$. To see this, first let $\mathbf{P}_{act}$ and $\mathbf{f}_{act}$ denote the rows of $\mathbf{P}$ and $\mathbf{f}$ indexed by $\mathcal{A}$ and let $\mathbf{R} = [\mathbf{A}^T \quad \mathbf{P}_{act}^T]^T$ and $\mathbf{e} = [\ \mathbf{b}^T \quad \mathbf{f}_{act}^T\ ]^T$. A solution to (18) can be found by first solving the following system of equations for $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$:

$$-\mathbf{R}\mathbf{W}^{-1}\mathbf{R}^T \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\gamma} \end{bmatrix} = \mathbf{e} + \mathbf{R}\mathbf{W}^{-1}\mathbf{g} \tag{20}$$

Using a solution to this system, $\mathbf{z}$ can be recovered via

$$\mathbf{z} = -\mathbf{W}^{-1}\left(\mathbf{g} + \mathbf{R}^T \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\gamma} \end{bmatrix}\right). \tag{21}$$

Efficient computation of $\mathbf{W}^{-1}$ arises from its block diagonal structure,

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{11} & 0 \\ 0 & \mathbf{W}_{22} \end{bmatrix}, \tag{22}$$

where $\mathbf{W}_{22}$ is diagonal and $\mathbf{W}_{11} = w_{\ddot{\mathbf{q}}}\mathbf{I} + \mathbf{U}^T\mathbf{Q}\mathbf{U}$. For the ZMP dynamics, $\mathbf{U} = \mathbf{D}(t)\mathbf{J} \in \mathbb{R}^{2\times n}$, where $\mathbf{J}$ is the COM(x, y) Jacobian and $\mathbf{D}(t)$ is the input mapping defined in (4). Applying the matrix inversion lemma yields an expression for $\mathbf{W}_{11}^{-1}$ that involves computing the inverse of 2×2 matrices:

$$\mathbf{W}_{11}^{-1} = \frac{1}{w_{\ddot{\mathbf{q}}}}\mathbf{I} - \frac{1}{w_{\ddot{\mathbf{q}}}^2}\mathbf{U}^T(\mathbf{Q}^{-1} + \frac{1}{w_{\ddot{\mathbf{q}}}}\mathbf{U}\mathbf{U}^T)^{-1}\mathbf{U}.$$

It should also be noted that $\mathbf{W}^{-1}$ is independent of $\mathcal{A}$ and thus only needs to be computed once per control step even if multiple solver iterations are required. The same holds for various sub-matrices in the expressions (20) and (21).

Data: A QP of form (17) where the cost matrix $\mathbf{W}$ has the structure (22). A set of constraints $\mathcal{A}$ assumed to be active at optimality.
Result: An optimal solution $\mathbf{z}$ with active set $\mathcal{A}$ or a flag indicating failure.

```
iter ← 0
repeat
    Compute candidate solution z, γ, α from (20,21)
    if p_i^T z > f_i then
        add i to A
    end
    if γ_i < 0 then
        remove i from A
    end
    iter ← iter + 1
    if iter > iter_MAX then
        return Failure
    end
until z and γ satisfy (19)
return A and z
```

Algorithm 1: Active-set method for solving (17). The set $\mathcal{A}$ passed to the algorithm at time t equals the set of constraints active at optimality for time $t-1$.

V. Application

We implemented our controller using the 34-DOF Atlas humanoid model developed for the DARPA Virtual Robotics Challenge. Our evaluation of the controller included a variety of balancing and locomotion tasks using two independent simulation environments: Drake [16] and Gazebo [17]. As part of MIT's entry into the DARPA Virtual Robotics Challenge (VRC), the controller was used to walk reliably over uneven terrain, through simulated knee-deep mud, and while carrying an unmodeled multi-link hose, all using imperfect state and terrain estimation (Figure 1).[1]

To design the balancing controller, we solved an infinite horizon LQR problem to regulate the ZMP at $(0, 0)$. The cost functional took the form

$$\begin{aligned} J &= \int_0^\infty \mathbf{y}^T \mathbf{Q} \mathbf{y} dt, \\ &= \int_0^\infty \left[\mathbf{x}^T \mathbf{C}^T \mathbf{C} \mathbf{x} + \mathbf{u}^T \mathbf{D}^T \mathbf{D} \mathbf{u} + 2 \mathbf{x}^T \mathbf{C}^T \mathbf{D} \mathbf{u} \right] dt, \end{aligned}$$

where $\mathbf{Q} = \mathbf{I}$. We assumed the COM height was constant while standing, thus making the ZMP dynamics linear. This had the advantage that it only required us to solve the LQR problem once. To see this, note that $J^*(\bar{\mathbf{x}}) = \bar{\mathbf{x}}^T \mathbf{S} \bar{\mathbf{x}}$, where $\mathbf{S}$ is the solution of the algebraic Riccati equation. Thus the QP cost had the form,

$$\bar{\mathbf{y}}^T \bar{\mathbf{y}} + 2 \bar{\mathbf{x}}^T \mathbf{S} (\mathbf{A} \bar{\mathbf{x}} + \mathbf{B} \mathbf{u}) + w_{\ddot{\mathbf{q}}} ||\ddot{\mathbf{q}}_{\text{des}} - \ddot{\mathbf{q}}||^2 + \varepsilon \sum_{ij} \beta_{ij}^2,$$

where new desired ZMP locations $\mathbf{k} = \begin{bmatrix} k_x & k_y \end{bmatrix}^T$ could be achieved by a change in coordinates, $\bar{\mathbf{y}} = \mathbf{y} - \mathbf{k}$, $\bar{\mathbf{x}} = \mathbf{x} - \mathbf{k}$, and $\mathbf{k}$ is, e.g., the point at the center of the foot support polygon. In practice, we found the constant COM height assumption has minimal practical effect on balancing performance, even when recovery motions included significant hip bends and arm motion. We computed $\ddot{\mathbf{q}}_{\text{des}}$ via a simple PD control rule, $\ddot{\mathbf{q}}_{\text{des}} = K_p(\mathbf{q}_{\text{des}} - \mathbf{q}) - K_d(\dot{\mathbf{q}})$, using either a fixed nominal posture, $\mathbf{q}_{\text{des}}$, for standing or a time-varying configuration trajectory for manipulation. We used the same scalar gains, K_p and K_d, for all joints.

Our planning implementation took desired foot trajectories as input and computed a ZMP plan, $\mathbf{y}^d(t)$, by linear interpolation between step locations. The footstep planner combined terrain map information with heuristics to select reasonable step locations and timing. We solved the TVLQR problem (3) for the linear ZMP dynamics using the Riccati solution for balancing as the final cost, $\mathbf{Q}_f = \mathbf{S}$. The corresponding COM$(x, y)$ trajectory, $\mathbf{x}^d(t)$, can be computed by simulating the COM dynamics (1) in a closed loop from time $t = 0$ to $t = t_f$ with the optimal controller, $\bar{\mathbf{u}}^* = -\mathbf{K}(t)\bar{\mathbf{x}}$. In practice, we were able to compute both $J^*(\bar{\mathbf{x}}, t)$ and $\mathbf{x}^*(t)$ for a 10m walking plan in approximately $1/4$s using an unoptimized MATLAB implementation of the explicit ZMP Riccati solution described by Tedrake et al. [18].

The desired configuration, $\mathbf{q}_{\text{des}}(t)$, was computed via inverse kinematics with constraints on the foot pose and COM position. Computation of $\mathbf{q}_{\text{des}}(t)$ was done either offline for open-loop trajectory following or reactively inside the control loop by linearizing the forward kinematics at the current configuration and solving a second small QP to minimize the weighted ℓ^2 distance to a nominal configuration while respecting foot pose, COM, and joint limit constraints. Qualitatively different motions could be achieved by varying the relative weights assigned to joints in the cost. For example, a smaller cost on back joints would tend to produce more torso sway to track the desired COM trajectory.

We used a simplified 4-point contact representation for each foot. Active contacts were determined by a combination of the desired footstep plan and the estimated distance between the foot and terrain. If and only if the foot is perceived to be in contact and the plan agreed did we include the corresponding foot contact in the optimization. The requirement that both conditions be true was essential for breaking contact with the ground while walking. As with balancing, footstep and ZMP plans could be translated in three dimensions without additional computation by a simple change in coordinates in the QP cost.

A. Solver Performance

We compared the solve time of our active-set algorithm against two general-purpose QP solvers, Gurobi [19] and CVXGEN [20]. For the Gurobi solver, we used the barrier (B) algorithm and dual simplex (DS) algorithm with both active constraint and solution warm-starting. Our CVXGEN problem formulation omitted the input saturation inequalities (13) to fit within the problem size requirements. These experiments were done on an i7 2.1GHz quad-core laptop. A comparison of average solve times while executing a fixed flat ground pattern is given in Table I.

TABLE I

Comparison of average QP solve times while walking.

	Algorithm 1	Gurobi (DS)	CVXGEN	Gurobi (B)
Solve time	0.2 ms	1.0 ms	2.2 ms	3.1 ms

The custom active-set method outperforms the next best solver by a factor of 5. The significant performance advantage of Algorithm 1 can be understood by considering the histogram in Figure 2. For an overwhelming percentage of control steps, the active set does not change and the solver succeeds in a single iteration. Thus, most of the time control inputs are computed by solving a single linear system of equations.

For the active-set algorithm, the total controller computation time is largely spent setting up the QP, which involves computing the manipulator dynamics, contact surface normals, and kinematic quantities such as the COM and contact Jacobians. In our implementation, the average QP setup time is approximately 0.8ms for the 34-DOF Atlas model, giving us a total control step time of 1ms.

The performance of the solver does have a subtle dependency on the problem formulation. We found that using the

[1] Example simulation code is available at `http://people.csail.mit.edu/scottk`.

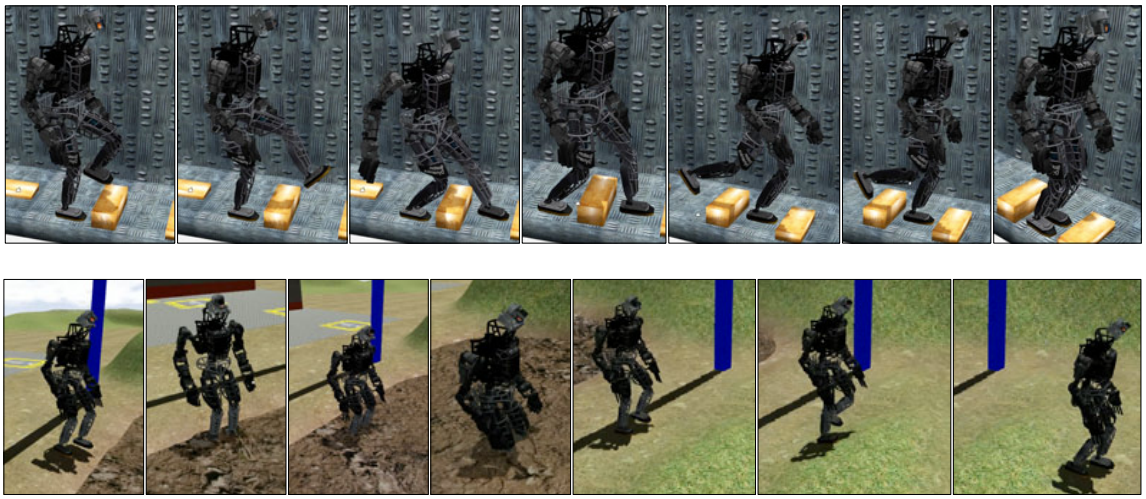

Fig. 1. Walking in simulation over obstacles, through simulated mud, and over rolling hills using state and terrain estimation.

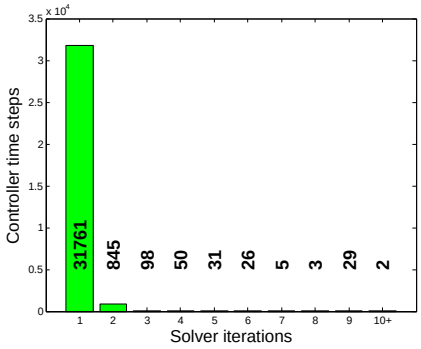

Fig. 2. Histogram of iterations needed to solve Quadratic Program 1 during a walking task. The method requires only one iteration approximately 97% of the time.

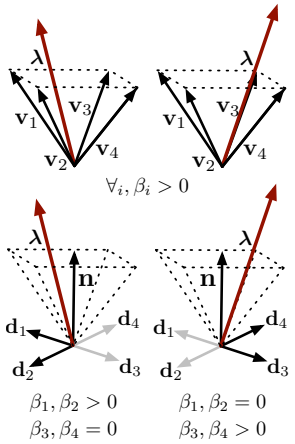

Fig. 3. An illustration showing how different approximate friction cone parameterizations can affect active set stability.

parameterization of the approximate friction cone (9) lead to fewer active set changes than the commonly used Stewart and Trinkle [21] parameterization,

$$\hat{K}_{\mathrm{ST}} = \left\{ z\mathbf{n} + \sum_{i=1}^{N_d} \beta_i \mathbf{d}_i : z \geq 0, \beta_i \geq 0, \sum_{i=1}^{N_d} \beta_i \leq \mu z \right\}, \quad (23)$$

where we have dropped the explicit contact point index, j. The parameterization (23) lead to approximately 50% more control steps requiring 2 iterations or more. Intuitively, this is a result of the fact that the active inequalities constraints, $\{i : \beta_i = 0\}$, under parameterization (23) can change when forces inside the approximate friction cone change direction. By contrast, when using (9), the constraints on β_i only become active on the surface of the polyhedron. This idea is illustrated in Figure 3.

VI. Related Work

The controller design we proposed shares some properties with other horizon-1 MPC implementations. For example, the same flavor of dynamic, friction, and foot motion constraints have appeared in other QP formulations [5], [9], [11], [12]. Herzog et al. [9] proposed the idea of separating the manipulator equation into floating-base and actuated DOFs to remove $\boldsymbol{\tau}$ as a decision variable, which enabled them to achieve control rates of 1kHz for a 14-DOF biped. Polyhedral approximations are frequently used to linearize friction constraints, but to our knowledge no prior connection has been made between different parameterizations and solver performance.

Ames et al. [22], [23] used CLFs for walking control design by solving QPs that minimize the input norm, $||\mathbf{u}||$, while satisfying constraints on the negativity of $\dot{V}_{\mathrm{clf}}$. By contrast, we placed no constraint on $\dot{V}_{\mathrm{clf}}$ and instead minimized an objective of the form $\ell(\mathbf{x}, \mathbf{u}) + \dot{V}_{\mathrm{clf}}$, where $\ell(\mathbf{x}, \mathbf{u})$ is an instantaneous cost on $\mathbf{x}$ and $\mathbf{u}$. This approach gave us the significant practical robustness while making the QP less prone to infeasibilities.

Other uses of active-set methods for MPC have exploited the temporal relationship between the QPs arising in MPC.

Bartlett et al. compared active-set and interior-point strategies for MPC [24]. The described an active-set approach based on Schur complements for efficiently resolving KKT conditions after changes are made to the active set. This framework is analogous to the solution method we discuss in Section IV-B. In the discrete time setting, Wang and Boyd [25] describe an approach to quickly evaluating control-Lyapunov policies using explicit enumeration of active sets in cases where the number of states is roughly equal to the square of the number of inputs.

Ferreau et al. [26] consider the MPC problems where the cost function and dynamic constraints are the same at each time step; i.e., the QPs solved at iteration differ only by a single constraint that enforces initial conditions. By smoothly varying the initial conditions from the previous to the current state, they were able to track a piecewise linear path traced by the optimal solution, where knot points in the path correspond to changes in the active set. Since the controller we considered had changing cost and constraint structure, this method would have been difficult to apply.

VII. Conclusion

We described a stabilizing QP controller formulation for dynamic walking and solution technique that exploits consistency between active inequality constraints in subsequent control steps. In our experiments with a simulated Atlas robot, we were able to efficiently compute control inputs while walking by solving a single system of linear equations a high-percentage of the time, hence outperforming several popular general-purpose solvers used frequently in the literature. Although we have focused on humanoids and ZMP dynamics in this paper, the QP formulation we described is equally applicable to more general floating-base systems and other types of simple system models. Similarly, the active-set method used in this work could easily be applied to the various MPC formulations that exist in the literature. Our current efforts are focused on adapting this approach to achieve stable walking, climbing, and manipulation with a physical Atlas humanoid robot at MIT.

Acknowledgments

We would like to thank the members of the MIT VRC team for their contributions to the perception and estimation algorithms that made walking in the simulation challenge possible. We thank Robin Deits for designing the footstep planner used by the controller described in this paper.

References

[1] M. Posa and R. Tedrake, "Direct trajectory optimization of rigid body dynamical systems through contact," in *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*, Cambridge, MA, 2012.

[2] B. Stephens and C. Atkeson, "Push recovery by stepping for humanoid robots with force controlled joints," in *Proceedings of the International Conference on Humanoid Robots*, Nashville, TN, 2010.

[3] D. Dimitrov, A. Sherikov, and P.-B. Wieber, "A sparse model predictive control formulation for walking motion generation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, USA, Sept. 2011, pp. 2292–2299.

[4] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[5] Y. Abe, M. da Silva, and J. Popović, "Multiobjective control with frictional contacts," in *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Aire-la-Ville, Switzerland, 2007, pp. 249–258.

[6] C. Collette, A. Micaelli, C. Andriot, and P. Lemerle, "Dynamic balance control of humanoids for multiple grasps and non coplanar frictional contacts," in *Proceedings of the IEEE/RAS International Conference on Humanoid Robots*, 2007, pp. 81–88.

[7] A. Macchietto, V. Zordan, and C. R. Shelton, "Momentum control for balance," in *Transactions on Graphics/ACM SIGGRAPH*, 2009.

[8] A. D. Ames, "First steps toward underactuated human-inspired bipedal robotic walking," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, St. Paul, MN, 2012.

[9] A. Herzog, L. Righetti, F. Grimminger, P. Pastor, and S. Schaal, "Momentum-based balance control for torque-controlled humanoids," *CoRR*, vol. abs/1305.2042, 2013.

[10] S. Kudoh, T. Komura, and K. Ikeuchi, "The dynamic postural adjustment with the quadratic programming method," in *International Conference on Intelligent Robots and Systems (IROS)*, October 2002, pp. 2563–2568.

[11] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Souères, and J.-Y. Fourquet, "Dynamic whole-body motion generation under rigid contacts and other unilateral constraints," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 346–362, April 2013.

[12] T. Koolen, J. Smith, G. Thomas, S. Bertrand, J. Carff, N. Mertins, D. Stephen, P. Abeles, J. Englsberger, S. McCrory, J. van Egmond, M. Griffioen, M. Floyd, S. Kobus, N. Manor, S. Alsheikh, D. Duran, L. Bunch, E. Morphis, L. Colasanto, K.-L. H. Hoang, B. Layton, P. Neuhaus, M. Johnson, and J. Pratt, "Summary of team IHMC's virtual robotics challenge entry," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, Atlanta, GA, Oct 2013.

[13] P. Sardain and G. Bessonnet, "Forces acting on a biped robot. Center of pressure-zero moment point," *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 34, no. 5, pp. 630–637, 2004. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TSMCA.2004.832811

[14] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, September 2003.

[15] N. S. Pollard and P. S. A. Reitsma, "Animation of humanlike characters: Dynamic motion filtering with a physically plausible contact model," in *Yale Workshop on Adaptive and Learning Systems*, 2001.

[16] "Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems," http://drake.mit.edu, September 2013. [Online]. Available: http://drake.mit.edu

[17] "Gazebo," http://gazebosim.org, September 2013.

[18] R. Tedrake, S. Kuindersma, R. Deits, and K. Miura, "An explicit solution for the ZMP planning problem with quadratic cost," *In Prep*, 2013.

[19] "Gurobi optimizer," http://www.gurobi.com, September 2013.

[20] J. Mattingley and S. Boyd, "CVXGEN: a code generator for embedded convex optimization," in *Optimization Engineering*, vol. 13, no. 1, 2012, pp. 1–27.

[21] D. E. Stewart and J. C. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction," *International Journal for Numerical Methods in Engineering*, vol. 39, no. 15, pp. 2673–2691, 1996.

[22] A. D. Ames, K. Galloway, and J. W. Grizzle, "Control Lyapunov functions and hybrid zero dynamics," in *Proceedings of the 51st IEEE Conference on Decision and Control*, Maui, HI, 2012.

[23] A. D. Ames, "Human-inspired control of bipedal robotics via control Lyapunov functions and quadratic programs," in *Hybrid Systems: Computation and Control*, 2013.

[24] R. A. Bartlett, A. Wächter, and L. T. Biegler, "Active set vs. interior point strategies for model predictive control," in *Proceedings of the American Control Conference*, Chicago, IL, June 2000.

[25] Y. Wang and S. Boyd, "Fast evaluation of quadratic control-Lyapunov policy," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 4, pp. 939–946, 2011.

[26] H. Ferreau, H. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit MPC," *International Journal of Robust and Nonlinear Control*, vol. 18, no. 8, pp. 816–830, 2008.