Convex Optimization Algorithms for Active Balancing of Humanoid Robots

Juyong Park, Jaeyoung Haan, and F. C. Park

***Abstract*—We show that a large class of active balancing problems for legged robots can be framed as a second-order cone programming (SOCP) problem, a convex optimization problem for which efficient and numerically robust algorithms exist. We describe this general SOCP balancing framework, show that several existing optimization-based balancing strategies reduce to special cases of this more general formulation, and investigate the computational performance of our SOCP algorithms through simulation studies involving a humanoid model.**

***Index Terms*—Active balancing, convex optimization, legged robot, second-ordercone programming (SOCP).**

I. Introduction

The problem of active balancing of legged robots—also referred to variously in the literature as "whole-body cooperative balancing," "whole-body stabilization," or "posture control"—involves the correction of a robot's motion, usually in real time, to prevent the robot from tipping over while being supported by one or more legs. An input reference trajectory, not necessarily stable or even dynamically feasible (e.g., torque limits may be exceeded), is typically assumed to be given. The objective is to correct the actual motion, in real time, so as to maintain balance in the presence of external disturbances while closely following the reference trajectory.

Almost all of the existing active balancing methods involve the zero-moment point (ZMP) [9], [10], and in most cases, the center of mass (COM) as well. Among the optimization-based balancing strategies—the focus of this paper—Kagami *et al*. [2] develop an active balancing algorithm for flat-footed biped robots, in which a least squares joint tracking error is minimized while maintaining the projected COM at a predefined fixed point, and confining the ZMP to the interior of the support region. Sugihara and Nakamura [8] also propose a quadratic-programming-based method that takes into account the constraints on the COM and ZMP, but assumes as input an already balanced trajectory that is subject to small, short-term disturbances. Kudoh *et al*. [3] present a quadratic programming strategy that minimizes joint accelerations while confining the ZMP to a prescribed region, and taking into account constraints on the COM acceleration, and structural symmetry of the legs (e.g., both feet of the biped are assumed to be in contact with the ground at all times). Yamane and Nakamura [11] also present a combined optimization and potential difference (PD) feedback control approach to correcting general motions that minimizes the error between the reference and actual joint accelerations.

As evident from the existing literature, from an optimization perspective, active balancing can be formulated in numerous ways and under a variety of assumptions, e.g., minimizing a joint space tracking error while fixing the projected COM and confining the ZMP to the support polygon, minimizing the deviation of the ZMP from the support polygon center while maintaining a joint tracking error tolerance, compensating for projected COM tracking errors by a PD-type feedback control of the projected COM acceleration, etc. The primary contribution of this paper is a general *convex-optimization*-based framework for the active balancing of legged robots that includes, as special cases, many of the previous approaches proposed in the literature.

Given a (possibly unstable) input reference trajectory in the form of joint and root frame acceleration time profiles, and a desired ZMP time profile, we develop algorithms that determine, at each time step, the optimal acceleration profile subject to a general set of COM and ZMP constraints, and under the assumption that current state measurements are available. No information about the future input motion, or the nature and magnitude of external disturbances, is assumed. We do, however, assume the availability of a dynamic model of the robot, together with limitations on the joint torques, joint ranges, and any other mechanical or actuator constraints that affect the optimization. Also, the desired ZMP time profile is not essential, as it can be generated easily from the reference trajectory without affecting the structure of the ensuing optimization.

One of the distinguishing features of our contribution is the identification of a family of active balancing algorithms that can be formulated as *second-order cone programming (SOCP)* problems. SOCP problems are a special class of convex optimization problems, for which highly efficient and reliable interior point algorithms have been developed. As is well known, convex optimization problems admit global solutions, and interior point algorithms for SOCP have been reported to typically converge in 30 or so iterations regardless of the problem dimension (see, e.g., [1] and [4]). Unlike more general convex optimization problems, the algorithms developed for SOCP problems have reached a level of maturity and sophistication close to that of linear programming [1].

In this paper, we develop a general SOCP framework for the active balancing problem. Section II shows how the various physical constraints can be formulated so as to satisfy the SOCP requirements. In Section III, we briefly review the basic features of the SOCP problem, and formulate the active balancing problem in a general way such that it leads to an SOCP problem. Section IV presents several case studies involving a biped humanoid robot that assess both the qualitative and computational performance of our SOCP balancing algorithms.

II. Mathematical Representation of Balancing Constraints

We assume that the reader is familiar with the notion of the ZMP and its calculation, and also with the concept and modern notation for twists and wrenches (see [7] for details), i.e., angular and linear velocities are represented in combined form as the twist $(\omega, v) \in se(3)$, where $se(3)$ is the Lie algebra of the Euclidean group of rigid-body motions $SE(3)$. Moments and forces are also represented in combined form as the wrench $(m, f) \in se(3)^*$, where $se(3)^*$ denotes the dual of $se(3)$. Finally, we recall the adjoint matrix representation Ad_T of $T \in SE(3)$ as the 6×6 matrix

$$\mathrm{Ad}_T = \begin{bmatrix} R & 0 \\ [p]R & R \end{bmatrix} \tag{1}$$

where $R \in SO(3)$ and $p \in \Re^3$, respectively, denote the rotation and position components of $T \in SE(3)$, with $[p]$ as the 3×3 skew-symmetric matrix representation of p. The dual adjoint Ad_T^* is simply given by the transpose of Ad_T.

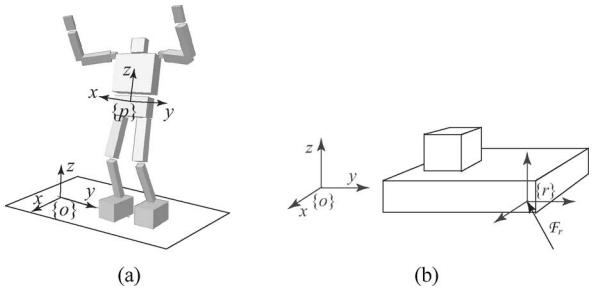
Referring to the biped humanoid model of Fig. 1, we denote the fixed reference frame by $\{o\}$, and attach a frame $\{r\}$ to the point at which the floor exerts a reaction force on the sole of the foot. The shape of each sole is assumed to be rectangular. The wrench $\mathcal{F} \in se^*(3)$ denotes the reaction moment and force expressed in the $\{r\}$ frame. Let $\mathcal{F}_o$

Manuscript received January 1, 2007. This paper was recommended for publication by Associate Editor K. Yamane and Editor H. Arai. This work was supported in part by the Frontier 21C Center for Intelligent Robotics and in part by the IAMD-SNU.

J. Park is with Samsung Advanced Institute of Technology, Gyeonggi-Do 446-712, Korea (e-mail: juyong.park@gmail.com).

J. Haan and F. C. Park are withthe School of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, Korea (e-mail: jaeyoung@robotics.snu.ac.kr; fcp@snu.ac.kr).

Digital Object Identifier 10.1109/TRO.2007.900639



(a) (b)

Fig. 1. Humanoid model: reference frames and forces. (a) Humanoid robot. (b) Contact forces at foot.

denote the wrench $\mathcal{F}$ expressed in the fixed frame, i.e., $\mathcal{F}_o = \mathrm{Ad}_T^* \mathcal{F}$. In anticipation of our later optimization formulation, we define the vector $\ddot{q} = (\dot{V}_0, \ddot{\theta})$, where $V_0 \in se(3)$ is the twist velocity of the pelvis and $\theta \in \Re^K$ is the vector of joint variables.

A. Constraints on the ZMP

The ZMP can be computed from the following set of equations:

$$p_{\mathrm{zmp}} = \frac{A_{\mathrm{zmp}} \mathcal{F}_o}{b_{\mathrm{zmp}} \mathcal{F}_o}$$

$$A_{\mathrm{zmp}} = \beta \begin{bmatrix} 0 & -1 & 0 & \\ 1 & 0 & 0 & 0_{3\times 3} \\ 0 & 0 & 0 & \end{bmatrix} \in \Re^{3\times 6}$$

$$b_{\mathrm{zmp}} = \beta [0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 1] \in \Re^{1\times 6}. \tag{2}$$

Here, the scalar β is set to either $+1$ or -1 so as to satisfy $b_{\mathrm{zmp}} \mathcal{F}_o > 0$.

The requirement that the computed ZMP lie in the interior of the support polygon $\mathcal{S}$ (i.e., the convex hull defined by all the points of contact between the floor and robot) can be expressed as a linear inequality of the form

$$A^{\mathcal{S}} p_{\mathrm{zmp}} + b^{\mathcal{S}} \leq 0 \tag{3}$$

for some matrix $A^{\mathcal{S}}$ and vector $b^{\mathcal{S}}$. From (2), the previous equation assumes a more detailed form

$$\left(A^{\mathcal{S}} A_{\mathrm{zmp}} + b^{\mathcal{S}} b_{\mathrm{zmp}} \right) \mathcal{F}_o \leq 0. \tag{4}$$

In general, this constraint will lead to a nonconvex optimization problem. One common way to remedy this situation is to provide a desired ZMP trajectory p^d_{zmp} as input, and to further require that, in addition to (3), the actual ZMP remain within some prescribed distance ϵ_{zmp} of the desired ZMP

$$\| p_{\mathrm{zmp}} - p^d_{\mathrm{zmp}} \| \leq \epsilon_{\mathrm{zmp}}. \tag{5}$$

Again taking into account (2), this equation can be more specified in more detail as

$$\| \left(A_{\mathrm{zmp}} - p^d_{\mathrm{zmp}} b_{\mathrm{zmp}} \right) \mathcal{F}_o \| \leq \epsilon_{\mathrm{zmp}} b_{\mathrm{zmp}} \mathcal{F}_o \tag{6}$$

where again $b_{\mathrm{zmp}} \mathcal{F}_o > 0$. It is also not essential to provide p^d_{zmp} as input; one possibility is to calculate the ZMP from the input reference motion, and in the event that it lies outside the support polygon, to project it to lie slightly inside the support polygon (or, even more simply, to take the center of the support polygon as the reference ZMP trajectory).

B. Constraints on the COM

As is well known, in the event that the support polygon is fixed, once the projected COM ventures outside the support polygon, gravitational roll and pitch moments that cause tipping are inevitably generated. Active balancing, thus, requires simultaneous consideration of both ZMP and COM constraints, as well as friction constraints. In [2], COM constraints are expressed as an equality constraint, in which the projected COM is fixed to a predetermined point within the support polygon. While this has the advantage of a simpler optimization formulation, it has the obvious disadvantage of excessively restricting the solution space when compared to, e.g., simply requiring the projected COM to lie within the support polygon.

Since the projected COM is completely determined by θ, in our optimization framework, we seek a $\ddot{q} = (\dot{V}_0, \ddot{\theta})$ that ensures that, at the next time step, the projected COM also remains within the support polygon. Denote the COM at the current time t by $p_{\mathrm{com}}(t)$. From the finite-difference approximation,

$$p_{\mathrm{com}}(t + \Delta t) \approx p_{\mathrm{com}}(t) + \dot{p}_{\mathrm{com}}(t) \Delta t + \frac{1}{2} \ddot{p}_{\mathrm{com}}(t) \Delta t^2 \tag{7}$$

the COM at the next time step, denoted by $p_{\mathrm{com\_next}}$, can be approximated as

$$p_{\mathrm{com\_next}} = A_{\mathrm{com\_next}} \ddot{q} + b_{\mathrm{com\_next}} \tag{8}$$

for appropriate $A_{\mathrm{com\_next}}$ and $b_{\mathrm{com\_next}}$. The requirement that the projected COM lies inside the support polygon $\mathcal{S}$ can now be expressed in linear inequality form

$$A^{\mathcal{S}}_{\mathrm{com}} P p_{\mathrm{com\_next}} + b^{\mathcal{S}}_{\mathrm{com}} \leq 0 \tag{9}$$

where P denotes the projection matrix to the ground, and $A^{\mathcal{S}}_{\mathrm{com}}$ and $b^{\mathcal{S}}_{\mathrm{com}}$ are appropriately obtained. As a matter of practical implementation, to compensate for errors in the finite-difference approximation, it is beneficial to use a slightly shrunken version of the support polygon $\mathcal{S}$. A similar strategy is suggested in [8], albeit in a somewhat different context.

An active balancing scheme is proposed in [3], in which the trajectory of the desired projected COM, denoted by $p^d_{\mathrm{pcom}}(t)$, is assumed to be given as input; the acceleration of the projected COM is then expressed in proportional derivative feedback control form, in which the proportional and differential errors between the desired and actual projected COM are multiplied by suitably small gains k_p and k_d, and summed to obtain the compensation term. Such schemes can also be included into our general convex optimization framework. Denote the acceleration of the projected COM by

$$\ddot{p}_{\mathrm{pcom}} = P \frac{\sum_i m_i a_i}{\sum_i m_i} = A_{\mathrm{ddcom}} \ddot{q} + b_{\mathrm{ddcom}} \tag{10}$$

where m_i and a_i, respectively, denote the mass and acceleration of link i, $A_{\mathrm{ddcom}}(\theta) \in \Re^{3\times N}$, and $b_{\mathrm{ddcom}}(V_0, \theta, \dot{\theta}) \in \Re^3$. The variable $\ddot{p}_{\mathrm{pcom}}$ is, then, assumed to be of the form

$$\ddot{p}_{\mathrm{pcom}} = -k_d \left(\dot{p}_{\mathrm{pcom}} - \dot{p}^d_{\mathrm{pcom}} \right) - k_p \left(p_{\mathrm{pcom}} - p^d_{\mathrm{pcom}} \right). \tag{11}$$

Assuming that the p^d_{pcom} trajectory remains inside the support polygon, (11) ensures that the robot maintains a statically balanced posture.

If p^d_{pcom} is supplied as an input, then (10) and (11) lead to an equality constraint of the form

$$A_{\mathrm{eq,static}} \ddot{q} = b_{\mathrm{eq,static}}. \tag{12}$$

Alternatively, one can make the assumption that $\dot{p}^d_{\mathrm{pcom}} = 0$, so that p^d_{pcom} is a fixed point inside the support polygon. In this case, one can

solve (10) and (11) to obtain $p^d_{\rm com}$ as a function of $\ddot{q}$. Confining $p^d_{\rm com}$ to the interior of the support polygon leads to an inequality constraint of the form

$$A^{\mathcal{S}}_{\rm com} p^d_{\rm pcom} + b^{\mathcal{S}}_{\rm com} \leq 0. \tag{13}$$

Equations (11) and (13) can be expressed collectively as

$$A_{\rm static}\ddot{q} \leq b_{\rm static}. \tag{14}$$

We remark that other linear equalities and inequalities can be obtained via alternative methods, e.g., [2].

C. Friction Constraints

To prevent the foot from slipping, both the force tangent to the ground, and the moment normal to the ground, need to be considered. In the case of a single foot contacting the floor, the basic frictional constraints can be expressed as

$$\|P_{xy}\mathcal{F}_o\| \leq \mu F_1 \tag{15}$$

$$\|P_{mz}\mathcal{F}_o - p_{\rm zmp} \times P_{xy}\mathcal{F}_o\| \leq \mu_r F_2 \tag{16}$$

$$F_1 + F_2 \leq P_z \mathcal{F}_o \tag{17}$$

where P_{xy}, P_{mz}, and P_z are the projection matrices required for obtaining the elements of force in the xy-plane, the moment normal to the floor, and the force along the z-axis, respectively. The variable μ is the frictional coefficient in the plane, μ_r is the torsional frictional coefficient normal to the floor, and F_1 and F_2 are the forces required to prevent the tangential force and normal moment from causing slippage. For the stationary foot, only the sum of these undetermined forces needs to be less than $P_z\mathcal{F}_0$.

Note that the ZMP is not known *a priori*; we can only assume $\|p_{\rm zmp} - p^d_{\rm zmp}\| \leq \epsilon_{\rm zmp}$. Since $p_{\rm zmp}$ depends on $\mathcal{F}_o$, (16) would not lead to a cone constraint, and the resulting optimization would not be convex. To circumvent this, we set $p_{\rm zmp} = p^d_{\rm zmp} + \Delta p_{\rm zmp}$; then, from the metric property, we can write the left-hand side of (16) as

$$\begin{aligned}\|P_{mz}\mathcal{F}_o - p^d_{\rm zmp} \times P_{xy}\mathcal{F}_o - \Delta p_{\rm zmp} \times P_{xy}\mathcal{F}_o\| \\ \leq \|P_{mz}\mathcal{F}_o - p^d_{\rm zmp} \times P_{xy}\mathcal{F}_o\| + \|\Delta p_{\rm zmp} \times P_{xy}\mathcal{F}_o\|.\end{aligned} \tag{18}$$

From (5) and (15), we can rewrite

$$\begin{aligned}\|P_{mz}\mathcal{F}_o - p_{\rm zmp} \times P_{xy}\mathcal{F}_o\| \\ \leq \|P_{mz}\mathcal{F}_o - p^d_{\rm zmp} \times P_{xy}\mathcal{F}_o\| + \epsilon_{\rm zmp}\mu F_1.\end{aligned} \tag{19}$$

Noting that (16) is always satisfied if

$$\|P_{mz}\mathcal{F}_o - p^d_{\rm zmp} \times P_{xy}\mathcal{F}_o\| + \epsilon_{\rm zmp}\mu F_1 \leq \mu_n F_2 \tag{20}$$

the resulting inequality is obtained as

$$\|P_{mz}\mathcal{F}_o - p^d_{\rm zmp} \times P_{xy}\mathcal{F}_o\| \leq \mu_r F_2 - \epsilon_{\rm zmp}\mu F_1. \tag{21}$$

In the course of simplifying the rotational friction constraints via (18), we have, in effect, imposed additional constraints beyond those physically required. In the event that $\epsilon_{\rm zmp}$ is set to zero, the additional constraint disappears, while for larger values of $\epsilon_{\rm zmp}$, the effects of this additional constraint increase.

In summary, given a desired ZMP trajectory $p^d_{\rm zmp}$ and threshold $\epsilon_{\rm zmp}$, and assuming multiple contacts, from the aforementioned simplification, the frictional constraints can now be expressed explicitly as follows:

$$\|P_{xy}\mathrm{Ad}^*_{T_j}\mathcal{F}_j\| \leq \lambda_j, \forall j \tag{22}$$

$$\|(P_{mz} - \left[p^d_{\rm zmp}\right] P_{xy})\mathcal{F}_o\| \leq \nu - \epsilon_{\rm zmp}\sum_j \lambda_j \tag{23}$$

$$\sum_j \frac{\lambda_j}{\mu} + \frac{\nu}{\mu_r} \leq P_z\mathcal{F}_o \tag{24}$$

where we use the notation $p_1 \times p_2 = [p_1]p_2$ for any $p_1, p_2 \in \Re^3$, j is the foot index for multiple contacts, and λ_j and ν are undetermined variables satisfying (22)–(24) (note that it is unnecessary to obtain λ_j and ν explicitly).

D. Contact Kinematic Constraints

The constraint that the supporting feet remain stationary, e.g., lifting or translating of any contacting feet is not permitted, in both static and dynamic environments, leads to an additional set of equality constraints. Denoting the body twist velocity for foot j by $V_j \in se(3)$, one then obtains the relation

$$\dot{V}_j = A_{\dot{V}_j}\ddot{q} + b_{\dot{V}_j} \tag{25}$$

where $A_{\dot{V}_j}(\theta) \in \Re^{6\times N}$ and $b_{\dot{V}_j}(V_0, \theta, \dot{\theta}) \in \Re^6$. This twist velocity is set to zero in the case of a static floor, or can be made to coincide with the velocity of the floor in the case of a dynamic environment. By stacking all such equations for each contacting foot, a linear equality constraint in $\ddot{q}$ of the form

$$A_{\dot{V}_{\rm consts}}\ddot{q} = \dot{V}_{\rm consts} - b_{\dot{V}_{\rm consts}} \tag{26}$$

is obtained. See, e.g., [5] for a systematic derivation of $\dot{V}_j$.

III. Convex Optimization Formulation

A. Second-Order Cone Programming

An SOCP problem can be expressed in the following form:

$$\min_x \quad f^T x \tag{27}$$

$$\text{subject to} \quad \|A_i x + b_i\| \leq c_i^T x + d_i, \qquad i = 1, \ldots, N \tag{28}$$

$$Fx = g \tag{29}$$

where $x \in \Re^n$ is the optimization vector $F \in \Re^{k\times n}$, and $f \in \Re^n$, $A_i \in \Re^{(n_i-1)\times n}$, $b_i \in \Re^{n_i-1}$, $c_i \in \Re^n$, and $d_i \in \Re$ are given constants. The norm is the standard Euclidean norm, i.e., $\|u\| = (u^T u)^{1/2}$. Inequality constraints of the previous form are called second-order cone constraints, since it is the same as requiring the affine function $\langle A_i x + b_i, c_i^T x + d_i\rangle$ to lie in the second-order cone in $\Re^{n_i}$. This optimization is a nonlinear convex problem that includes linear and quadratic programs; in fact, if each $c_i = 0$, the SOCP problem reduces to a quadratically constrained quadratic programming problem. Details of the SOCP problem, as well as the various interior point algorithms for their solution, can be found in [1].

B. General SOCP Formulation for Active Balancing

For the basic case of minimizing acceleration tracking errors, in which both the COM and ZMP constraints are expressed in inequality form, with $p^d_{\rm zmp}$ and $\epsilon_{\rm zmp}$ given, the resulting optimization formulation is

$$\min \tau$$

subject to

$$\|\ddot{q} - \ddot{q}_{\text{ref}}\| \le \tau$$
$$A_{\text{static}}\ddot{q} \le b_{\text{static}}$$
$$(A^{\mathcal{S}}A_{\text{zmp}} + b^{\mathcal{S}}b_{\text{zmp}})\mathcal{F}_o \le 0$$
$$\|(A_{\text{zmp}} - p^d_{\text{zmp}}b_{\text{zmp}})\mathcal{F}_o\| \le \epsilon_{\text{zmp}}b_{\text{zmp}}\mathcal{F}_o$$
$$\|P_{xy}\text{Ad}^*_{T_j}(\mathcal{F}_j)\| \le \lambda_j \qquad \forall j \in [1,m]$$
$$\|\left(P_{mz} - \left[p^d_{\text{zmp}}\right]P_{xy}\right)\mathcal{F}_o\| \le \nu - \epsilon_{\text{zmp}}\sum_j \lambda_j$$
$$\sum_j \frac{\lambda_j}{\mu} + \frac{\nu}{\mu_r} \le P_z\mathcal{F}_o$$
$$A_{\dot{V}_{\text{consts}}}\ddot{q} = \dot{V}_{\text{consts}} - b_{\dot{V}_{\text{consts}}}$$
$$\mathcal{F}_o = \sum_j \text{Ad}^*_{T_j}(\mathcal{F}_j)$$

where $\mathcal{F}_o = M\ddot{q} + C, M(\theta) \in \Re^{6\times N}, C(V_0,\theta,\dot{\theta}) \in \Re^6$ are obtained from the dynamics equations, and m denotes the number of foot-ground contacts.

To recast this as a SOCP problem, we introduce a new optimization variable

$$\mathbf{x} = (\tau, \ddot{q}, \mathcal{F}_1, \ldots, \mathcal{F}_m, \lambda_1, \ldots, \lambda_m, \nu) \in \Re^{2m+N+1}$$

and define appropriate selection matrices, e.g., $S_{\ddot{q}}\mathbf{x} = \ddot{q}, S_\nu\mathbf{x} = \nu, (S_{\boldsymbol{\lambda}}\mathbf{x})_j = \lambda_j$, etc. We then have

$$\min f^T\mathbf{x} \tag{30}$$

subject to

$$\|S_{\ddot{q}}\mathbf{x} - \ddot{q}_{\text{ref}}\| \le S_\tau\mathbf{x} \tag{31}$$
$$A_{\text{static}}S_{\ddot{q}}\mathbf{x} \le b_{\text{static}} \tag{32}$$
$$\left(A^{\mathcal{S}}A_{\text{zmp}} + b^{\mathcal{S}}b_{\text{zmp}}\right)\mathcal{F}_o \le 0 \tag{33}$$
$$\|\left(A_{\text{zmp}} - p^d_{\text{zmp}}b_{\text{zmp}}\right)\mathcal{F}_o\| \le \epsilon_{\text{zmp}}b_{\text{zmp}}\mathcal{F}_o \tag{34}$$
$$\|P_{xy}\text{Ad}^*_{T_j}(S_{\mathcal{F}}\mathbf{x})_j\| \le (S_{\boldsymbol{\lambda}}\mathbf{x})_j \qquad \forall j \in [1,m] \tag{35}$$
$$\|\left(P_{mz} - \left[p^d_{\text{zmp}}\right]P_{xy}\right)\mathcal{F}_o\| \le S_\nu\mathbf{x} - \epsilon_{\text{zmp}}\mathbf{1}_{1\times m}S_{\boldsymbol{\lambda}}\mathbf{x} \tag{36}$$
$$\frac{1}{\mu}\mathbf{1}_{1\times m}S_{\boldsymbol{\lambda}}\mathbf{x} + \frac{1}{\mu_r}S_\nu\mathbf{x} \le P_z\mathcal{F}_o \tag{37}$$
$$A_{\dot{V}_{\text{consts}}}S_{\ddot{q}}\mathbf{x} = \dot{V}_{\text{consts}} - b_{\dot{V}_{\text{consts}}} \tag{38}$$
$$\mathcal{F}_o = HS_{\mathcal{F}}\mathbf{x} \tag{39}$$

where $\mathcal{F}_o = MS_{\ddot{q}}\mathbf{x} + C$, $H = [\text{Ad}^*_{T_1}\cdots\text{Ad}^*_{T_m}] \in \Re^{6\times 6m}$, $f = [1 \quad \mathbf{0}_{1\times(2m+N)}]^T \in \Re^{1\times(2m+N+1)}$, and $\mathbf{1}$ is a matrix all of whose elements are 1.

In the case of the problem addressed in [2], in which the COM constraints are given in equality form, it can be expressed as a special case of our SOCP formulation by replacing (32) with (12), setting ϵ_{zmp} and p^d_{zmp} to zero, and eliminating (34), (35), (39), and the variable $\boldsymbol{\lambda}$. Alternatively, one can express the ZMP constraints in equality form, and the COM constraint in inequality form, to arrive at an SOCP formulation. Other objective functions subject to combinations of the aforementioned constraints (e.g., minimizing joint torques or joint accelerations subject to equality or inequality ZMP and COM constraints, and tracking the desired acceleration profile to within some prescribed bound) also admit a straightforward formulation as an SOCP problem; due to space limitations, we do not provide details of these formulations here.

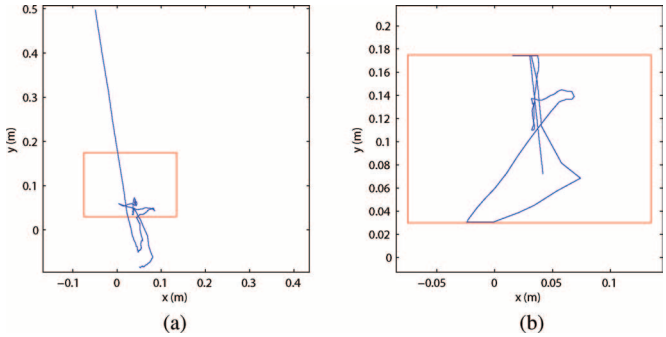


Fig. 2. Case study 1: ZMP trajectories for stabilized front kick. The dotted box represents the supporting polygon. (a) Before stabilization. (b) After stabilization.

IV. Case Studies

To assess both the computational performance and the qualitative nature of the motions generated by our SOCP active balancing algorithms, several case studies with the biped humanoid robot of Fig. 1 are conducted. The robot has 31 DOFs and 25 actuated joints: six at each limb and one at the rotational DOF about the waist. Values for the kinematic and inertial parameters are set to approximate a typical full-size biped humanoid. All simulations are performed in Matlab v6.5 running on a Pentium 1.4-GHz laptop. Each simulation time interval is 0.025 s. For the SOCP optimization, we use the Matlab toolbox provided by MOSEK.

For the input motions and environmental disturbances used in our case studies, we assume that the maximum joint torques are sufficient for the robot to maintain active balance and set the friction coefficient to be 0.6.

A. Simulation Results

As our first case study, we consider the problem of having the robot imitate human motions, i.e., kicking in a static environment. Obviously, because of structural and inertial differences between the human and robot, the input motions need to be corrected in order for the robot to perform them without tipping over. It should also be apparent that some motions may be inherently unstabilizable because of, e.g., insufficient torque capacity, lack of friction, the motions may be excessively fast, etc.

Computational experiments involving a front kick and side kick motion are performed for the most general SOCP optimization, as described in (30); this optimization is repeatedly performed at each time step over the entire duration of the motion. Setting ϵ_{zmp} to $0.5d_{\max}$ for both motions, where $d_{\max}$ denotes the diameter of the smallest circle enclosing the support polygon, the computation time for both motions, which involves computing the COM and ZMP, calculating the dynamics, and performing the SOCP optimization, is 0.4 s.

For the front kick motion, the ZMP trajectories before and after stabilization are shown in Fig. 2. The small rectangle denotes the support polygon, and the x-axis denotes the forward direction along which the kick is directed. The figure clearly shows the ZMP always confined to the interior of the support polygon following our optimization procedure.

As our second case study, we consider the problem of maintaining a desired posture in a dynamically changing environment. A biped

TABLE I
CASE STUDY 2: POSTURE STABILIZATION IN A DYNAMIC ENVIRONMENT

Acceleration profile ($\lambda = 2\pi/t_f$)	α	ϵ_{zmp}	*Time/step* (sec)
$\alpha \sin \lambda t$ (translation along x axis)	$1.3m/s^2$	0	0.40
		$0.1d_{max}$	0.41
		$0.5d_{max}$	0.42
$-\alpha\lambda^2 \sin \lambda t$ (rotation along y axis)	$7\text{deg}/s^2$	0	0.41
		$0.1d_{max}$	0.42
		$0.5d_{max}$	0.42

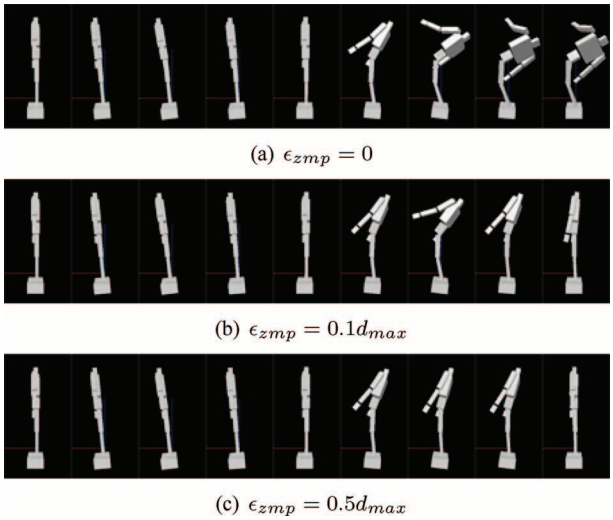


Fig. 3. Case study 2: stabilized motions on rotating ground. The floor is rotating along the lateral axis of the robot. With smaller ϵ_{zmp}, i.e., narrower region of available ZMP, the robot bends more in order to balance itself. (a) $\epsilon_{\text{zmp}} = 0$. (b) $\epsilon_{\text{zmp}} = 0.1d_{\text{max}}$. (c) $\epsilon_{\text{zmp}} = 0.5d_{\text{max}}$.

robot standing with both feet planted to the floor is subject to two separate accelerations (the floor is given by the x–y plane, with the robot facing the x-direction): 1) a linear acceleration in the x-direction and 2) a rotational acceleration about the y-axis. The exact form of the acceleration profiles for the two cases are given in Table 1, together with the average computation times for the optimization. Note that the computation times are very close to the previous case study, i.e., on the order of 0.4 s.

Given the standing posture as the desired posture, and adopting the same SOCP formulation as (30), the resulting stabilized motions are shown in Figs. 3 and 4 for various values of ϵ_{zmp}. Recall that larger values of ϵ_{zmp} result in closer tracking of the desired projected COM (set to be the center of the support polygon); from our simulations, we can observe that larger values of ϵ_{zmp} result in more upright motions, while smaller values of ϵ_{zmp} result in larger upper body motions for maintaining balance.

B. Discussion

One of the notable features of our algorithm is that throughout our case studies (including those not reported here), and over a wide range of parameter values and scenarios, the SOCP optimization procedure shows remarkable consistency in terms of both computation times (around 0.4 s, with a standard deviation of less than 0.2 s) and the number of iterations per optimization (less than 30).

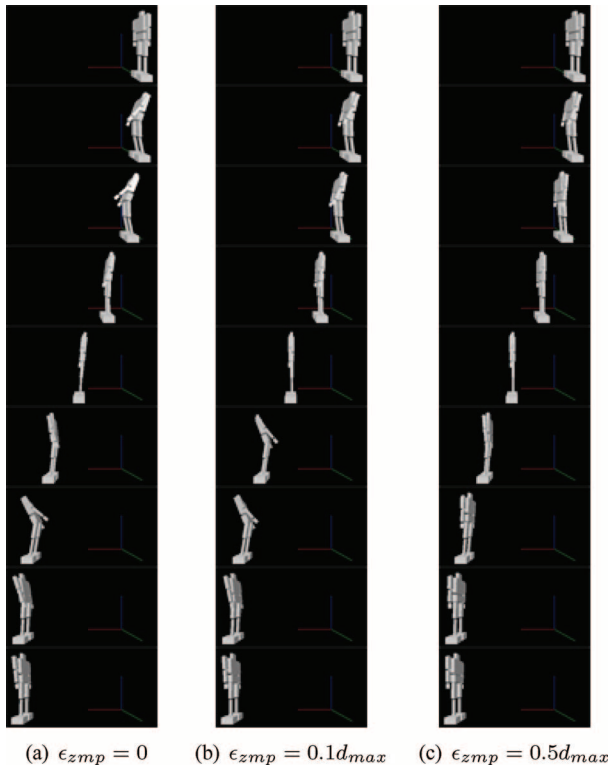


Fig. 4. Case study 2: stabilized motions on translating ground. Similar to Fig. 3, when the floor accelerated and decelerated, the robot curves its back for maintaining the balance. (a) $\epsilon_{\text{zmp}} = 0$. (b) $\epsilon_{\text{zmp}} = 0.1d_{\text{max}}$. (c) $\epsilon_{\text{zmp}} = 0.5d_{\text{max}}$.

We now discuss the effects of our choice of parameters on the resulting motion. The friction coefficient, as expected, significantly impacts the active balancing algorithm. In cases where the friction coefficient is less than 0.5, with complex input motions and large disturbances, solutions can fail to exist. That the algorithm is more effective with greater friction is not surprising, since balancing is far more difficult on slippery floors.

Our motions are also affected by the choice of ϵ_{zmp}. As mentioned in our previous discussion on ZMP constraints, larger values of ϵ_{zmp} tend to magnify the effects of the extraneous rotational friction constraint. If ϵ_{zmp} is set to $0.5d_{\text{max}}$ (i.e., the diameter of the smallest circle enclosing the support polygon), and p^d_{zmp} is fixed to be the center of this circle, then any point inside the support polygon is a candidate ZMP. (If $\epsilon_{\text{zmp}} = d_{\text{max}}$, then regardless of the choice of p^d_{zmp}, any point inside the support polygon is a candidate ZMP; values of ϵ_{zmp} larger than d_{max} are, therefore, superfluous.) While choice of ϵ_{zmp} will, to some extent, require tuning for different platforms and application scenarios, in our case studies, we have found $\epsilon_{\text{zmp}} \leq 0.5d_{\text{max}}$ effectively allows nearly all points within the support polygon to be a candidate ZMP.

Another choice affecting the stabilized motions is the actual support polygon $\mathcal{S}$ used in the optimization. Since during the course of the stabilization, the projected COM approaches $p^d_{\text{pcom}} \in \mathcal{S}$, a smaller $\mathcal{S}$ tends to lead to more stable motions. Also, the faster the convergence toward p^d_{pcom} (achieved by increasing the values for k_d and k_p, for example), the more quickly and efficiently the robot is able to react to dynamically changing environments and other disturbances. The obvious cost with a smaller $\mathcal{S}$ is that the solution space effectively becomes smaller as well.

V. Conclusion

This paper has presented a general convex optimization framework for the active balancing of legged robots. Assuming as given a (possibly unstable) input reference trajectory in the form of joint and root frame acceleration time profiles, and a desired ZMP time profile, we have derived SOCP algorithms that determine, at each time step, the optimal acceleration profile subject to more general COM and ZMP constraints, and under the assumption that current state measurements (i.e., the joint positions and velocities) are available. SOCP formulations are achieved by: 1) introducing a user-supplied parameter ϵ_{zmp} that reflects the allowed tolerance between the actual and desired ZMP trajectories and 2) expressing the rotational friction constraints in a simplified but slightly more restrictive form. Even with these restrictions, our framework is shown to be more general than are the existing methods of, e.g., [2], [3], and [8].

More importantly, because the optimization formulations now admit the structure of an SOCP problem, solutions can be obtained quickly and reliably: our case studies involving a biped robot, which has 25 joints, indicate that over a wide range of scenarios and parameter values, solutions are consistently obtained within 30 iterations, and that with current desktop computing technology (our simulations were performed in Matlab on a notebook computer), real-time solutions are within reach. Efforts toward both improving the computational aspects of the optimization, and exploring alternative convex optimization formulations for active balancing, are currently underway.

References

[1] S. Boyd and L. Vandenverghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[2] S. Kagami, F. Kanehiro, Y. Tamiya, M. Inaba, and H. Inoue, "AutoBalancer: An online dynamic balance compensation scheme for humanoid robots," *Proc. Int. Workshop Algorithmic Found. Robot. (WAFR)*, Hanover, NH, 2000, pp. 329–340.

[3] S. Kudoh, T. Komura, and K. Ikeuchi, "The dynamic postural adjustment with the quadratic programming method," *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2002, vol. 3, pp. 2563–2568.

[4] M. S. Lobo, L. Vandenverghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra Appl.*, vol. 284, pp. 193–228, 1998.

[5] F. C. Park, J. Choi, and S. R. Ploen, "Symbolic formulation of closed chain dynamics in independent coordinates," *Mech. Mach. Theory*, vol. 34, pp. 731–751, 1999.

[6] J. Park and F. C. Park, "A convex optimization algorithm for stabilizing whole-body motions for humanoid robots, presented at the 10th Symp. Adv. Robot Kinemat., Ljubljana, Slovenia," 2006.

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

[8] T. Sugihara and Y. Nakamura, "Whole-body cooperative balancing of humanoid robot using COG Jacobian," *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2002, vol. 3, pp. 2575–2580.

[9] M. Vukobratovic and D. Juricic, "Contribution to the synthesis of biped gait," *IEEE Trans. Biomed. Eng.*, vol. BME-16, no. 1, pp. 1–6, Jan. 1969.

[10] M. Vukobratovic and Y. Stepanenko, "On the stability of anthropomorphic systems," *Math. Biosci.*, vol. 15, pp. 1–37, 1972.

[11] K. Yamane and Y. Nakamura, "Dynamics filter—Concept and implementation of on-line motion generator for human figures," *IEEE Trans. Robot. Autom.*, vol. 19, no. 3, pp. 421–432, Jun. 2003.

Gyrobot: Control of Multiple Degree of Freedom Underactuated Mechanisms Using a Gyrating Link and Cyclic Braking

James M. Gilbert

***Abstract*—A novel control strategy is introduced in which a multiple degree of freedom passive joint mechanism is augmented with a single continuously gyrating link. The gyrating link introduces coupling torques to the remaining joints, and the effect of these torques is controlled by cyclically applying holding brakes mounted on the passive joints. This allows position and trajectory control of the mechanism.**

***Index Terms*—Nonholonomic, space robots, switching systems, underactuated systems, variable structure systems.**

I. Introduction

Underactuated mechanisms, in which there are fewer actuators than degrees of freedom, have attracted considerable research interest because they have the potential to reduce the cost and weight associated with actuators while maintaining a high degree of dexterity. The key challenge in controlling such mechanisms is to find a method of generating the desired motion in unactuated or passive links. The dynamics of typical multiple degree of freedom mechanisms contain four effects which introduce coupling between links: inertia coupling, centripetal forces, coriolis forces, and gravitational forces [1], [2]. A number of authors have considered these coupling terms, either singly or in combination, to produce desirable motion in the passive joints of underactuated mechanisms, [2]–[5]. The control of such mechanisms is not trivial, and a number of approaches have been studied including feedback linearization [3], passivity based control [6], fuzzy control [7], and the use of an amplitude modulated periodic input torque to ensure attraction to an elliptical manifold [8]. An overview of controllability and stabilisability results is presented in [9]. The broader class of nonholonomic systems, to which many underactuated manipulators belong, has also been widely studied in [10] and [11].

In the aforementioned approaches, the desired motion in the passive joint(s) of an underactuated mechanism is generated by imposing a particular motion in the actuated joint(s). This may constrain the types of movements which can be achieved in the end effector. Two approaches have been considered to overcome this problem: the use of holding brakes on the passive joints, and the addition of actuated links solely for the purpose of introducing coupling effects.

The mechanisms considered in [12] and [13] include a number of passive joints with holding brakes and an equal or greater number of actuated joints. When the brakes are engaged the passive joint position is held constant, allowing the actuated joint to move as desired. Once the brakes are released the coupling torque is effective, allowing control of the passive joint. By combining these two modes, it is possible to achieve position control of both links and trajectory control of the complete mechanism. Although this approach allows the end effector to follow a desired trajectory, it is not possible to independently control its speed along the trajectory. This concept is extended in [14], to the case where the number of passive joints is greater than the number of active joints. In this case, groups of passive joints are fixed using the

Manuscript received October 27, 2006; revised January 1, 2007. This paper was recommended for publication by Associate Editor R. Roberts and Editor H. Arai upon evaluation of the reviewers' comments.

The author is with the Department of Engineering, University of Hull, Hull HU6 7RX, U.K., (e-mail: j.m.gilbert@hull.ac.uk).

Digital Object Identifier 10.1109/TRO.2007.900632