12

Unification of Bipedal Robotic Walking using Quadratic Program-based Control Lyapunov Function

Applications to Regulation of ZMP and Angular Momentum

Pilwon Hur,* *Kenneth Chao* and *Victor Christian Paredes Cauna*

ABSTRACT

This chapter presents how robotic bipedal walking control can unify locomotion, manipulation, and force-based tasks into a single framework via quadratic programs utilizing control Lyapunov functions. We introduce two common examples where the unification can be applied: (1) ZMP-based pattern generation and locomotion, and (2) nonlinear dynamics with push recovery. Specifically, for ZMP-based pattern generation and locomotion, the proposed controller unifies both model predictive control-based center of mass planning and locally exponentially stabilizating control Lyapunov functions. For nonlinear dynamics with push recovery, the proposed controller unifies exponentially stabilizating control Lyapunov functions with centroidal angular momentum regulators. In both examples, unification of multiple control tasks into a single framework via quadratic program has benefits in terms of feasibility of constraints, simultaneous treatment of control objectives, and robustness and extensibility of this approach to various applications. In both examples, the end result was a single quadratic program that can dynamically balance all of these disparate objectives.

Department of Mechanical Engineering, Texas A&M University, College Station, TX, USA.
* Corresponding author: pilwonhur@tamu.edu

This approach can be implemented as a single framework and allow for more holistic implementation of controllers on bipedal robotic walking.

Keywords: Locomotion, control lyapunov function, quadratic program, angular momentum, unification, model predictive control

INTRODUCTION

Control of bipedal robotic walking is one of the main research interests in robotic literature. Among various approaches, two control methods have been used predominantly: (1) zero-moment point (ZMP)-based control (Kajita et al. 2003a, Kuindersma et al. 2014, Stephens and Atkeson 2010, Wieber 2006) and (2) nonlinear control methods that can enable dynamic walking (Ames 2014, Grizzle and Westervelt 2008, Morris and Grizzle 2005). Let's briefly look at both ZMP and nonlinear dynamic methods, respectively. ZMP is a point where influence of all generalized forces (e.g., forces, torques) acting on the foot of a robot can be replaced by one single force (Vukobratovic and Borovac 2004). ZMP-based methods usually include a simplified linear inverted pendulum model (LIPM) with the robot's center of mass (COM) and its ZMP. ZMP-based methods are simple enough that the algorithms can be implemented and computed in real time. Even though the ZMP-based methods can provide the dynamic balance all the time by keeping the ZMP within the support polygon under the foot, the resulting walking on the full-dimensional robot is usually limited to a quasi-static behavior. Nonetheless, various implementations of ZMP-based methods have been proposed and realized: kinematics-based control (Kajita et al. 2003a), force/torque-based control (Herzog et al. 2014, Kajita et al. 2003b, Macchietto et al. 2009) and whole body control with manipulation (Saab et al. 2013) to name a few. Experimental success of ZMP-based methods has motivated further attempts to unify ZMP-based methods with nonlinear control methods into a single-unified framework.

On the other end of the spectrum, several nonlinear control methods have been proposed to enable dynamic walking for bipedal robots. McGeer (1990) showed that exploiting the dynamics of a simple robotic system can achieve more agile and human-like walking even without any controls compared to ZMP-based methods. Several nonlinear control methods could fully utilize the benefits of the robot dynamics via Poincare analysis for the stability of the hybrid dynamical system. Efforts have been made for deriving proper initial conditions for passive robots, or driving a robot actively to the region of attraction within an invariant set where a stable periodic orbit exists. Examples of nonlinear control include direct trajectory optimization (Posa and Tedrake 2013), hybrid zero dynamics (Grizzle and Westervelt 2008), human-inspired control with partial hybrid zero dynamics (Ames 2014, Zhao et al. 2014). Even though these methods utilizing full-order dynamics of the system can provide natural benefits of human-like and energy-efficient robotic walking, generating gaits patterns that can simultaneously consider whole-body manipulation and robust walking in the presence of mild external perturbations is still a challenging control problem due to the high complexity of the dynamics and uncertainties of the environment.

In both methods, whether it is ZMP-based or nonlinear control methods, there are several criteria that need to be considered simultaneously when generating control laws. These criteria include stability, posture specification (e.g., trajectory tracking for each joint), other walking performance (e.g., control efforts, angular momentum, torque saturation, ZMP constraints). Kuindersma et al. (2014) proposed a quadratic program that combines MPC of walking pattern generation and locomotion control of real-time implementation. Ames and Powell (2013) introduced a unifying method of robotic locomotion and manipulation using control Lyapunov function (CLF) and quadratic programming (QP). The CLF-QP method was later experimentally realized with human-inspired walking control and MPC for full dynamics of robot (Powell et al. 2015). This unification method (CLF-QP) could combine the aforementioned various criteria into a single unified and implementable framework. The CLF-QP framework could dynamically balance all of various disjoint controller designs through weighted inequality constraints, which provide a holistic implementation of controllers on the robotic system. Even though some drawbacks on QP problems including real-time implementation and possible infeasibility may exist, the benefits of CLF-QP should not be undermined.

In our paper, we will demonstrate how CLF-QP framework can be used to unify locomotion controller design with two examples: (1) controller design using ZMP-based method and (2) controller design using nonlinear control method for push recovery. The structure of this paper is as follows. In Section **Control Lyapunov Function-Based Quadratic Program**, we will introduce and summarize the mathematical framework of CLF-QP that will be used in the two examples in Sections **Controlling Robot Locomotion With Zmp Constraints** and **Unification of Controller Design Using Nonlinear Control Methods for Push Recovery**, respectively. Section **Controlling Robot Locomotion With ZMP Constraints** will handle the unification of ZMP-based control using CLF-QP. ZMP constraints, linear inverted pendulum model (LIPM) for planning of COM trajectory, and a model predictive control (MPC) approach for pattern generation of LIPM will also be introduced. Section **Unification of Controller Design Using Nonlinear Control Methods for Push Recovery** will illustrate the unification of controller design using nonlinear control methods for push recovery. Specifically, human-inspired control, partial hybrid zero dynamics, and angular momentum regulation will be introduced. Finally, conclusion and future directions will be discussed in Section **Conclusion**.

CONTROL LYAPUNOV FUNCTION-BASED QUADRATIC PROGRAM

In this section, we summarize the framework of control Lyapunov function-based quadratic program (CLF-QP). This framework will be used in Sections **Controlling Robot Locomotion With Zmp Constraints** and **Unification of Controller Design Using Nonlinear Control Methods for Push Recovery** as a tool for unifying various control methodologies. Readers who want to know the detailed theoretical framework for CLF-QP are advised to refer to Ames and Powell (2013).

We assume that the equations of motion for a robot can be represented by the following form of the Euler-Lagrange equations:

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

where $q \in Q \subset \mathbb{R}^{n\times 1}$, Q is the configuration space of a robot with degrees of freedom, 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 torque map that determines which torque inputs actuate the robot. By reformulating (1), the following affine control system in the form of ODE can be obtained:

$$\dot{x} = f(x) + g(x)u. \tag{2}$$

where $x = (q, \dot{q}) \in TQ \subset \mathbb{R}^{2n}$, and TQ represents the tangent bundle of Q.

Since robotic walking involves discrete impacts, the discrete behavior in conjunction with the continuous dynamics can be formulated in terms of hybrid systems:

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

where $\mathcal{D} \subset TQ \subset \mathbb{R}^{2n}$ is the domain of a smooth submanifold of the continuous dynamics, $\mathcal{U} \subset \mathbb{R}^{n}$ is the set of all admissible control, $\mathcal{S}$ is a proper subset of $\mathcal{D}$ and is called the guard or switching surface at which discrete changes in the dynamics happen, Δ is a smooth map called the reset map and determines the discrete changes in dynamics, and (f, g) is the affine control system governing the continuous dynamics. The details of reset map can be found in Ames (2012).

Suppose that the nonlinear robot dynamics in the form of (2) is associated with an output y_2 of relative degree 2. Then, the input/output relation can be expressed as follows:

$$\ddot{y}_2(q) = L_f^2 y_2(q) + L_g L_f y_2(q) u = L_f + Au, \tag{4}$$

where L_f is the *Lie* derivative and A denotes the decoupling matrix which is always invertible since outputs can be chosen mutually exclusive (Zhao et al. 2014). If (2) is associated with both output y_1 of relative degree 1 and output y_2 of relative degree 2, the input/output relation can be expressed as follows:

$$\begin{bmatrix} \dot{y}_1(q, \dot{q}) \\ \ddot{y}_2(q) \end{bmatrix} = \begin{bmatrix} L_f y_1(q, \dot{q}) \\ L_f^2 y_2(q) \end{bmatrix} + \begin{bmatrix} L_g y_1(q, \dot{q}) \\ L_g L_f y_2(q) \end{bmatrix} u = L_f + Au, \tag{5}$$

The control law of feedback linearization for (4) or (5) is then given as follows:

$$u = A^{-1}(-L_f + \mu) \tag{6}$$

Applying the feedback law in (6) back to (4) or (5) will result in $\ddot{y}_2(q) = \mu$ or $[\dot{y}_1(q, \dot{q}),\ \ddot{y}_2(q)]^T = \mu$, respectively. Designing appropriate control μ can drive the outputs y_1 and y_2 to zero exponentially. For, $\eta \in H$ where H is a controlled space with appropriate dimension, defining the output coordinate $\eta = [y_2, \dot{y}_2]^T$ or $\eta = [y_1, y_2, \dot{y}_2]^T$ for (4) or (5) yields the following dynamics:

$$\dot{\eta} = F\eta + G\mu \tag{7}$$

where

$$F = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}, G = \begin{bmatrix} 0 \\ I \end{bmatrix} \quad \text{for dynamics (4)}$$

or

$$F = \begin{bmatrix} 0 & 0 \\ 0 & I \\ 0 & 0 \end{bmatrix}, G = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & I \end{bmatrix} \quad \text{for dynamics (5)} \tag{8}$$

Please note that the original affine nonlinear system (2) was turned into a linear system (7) as a result of input/output feedback linearization (Isidori 1995, Sastry 1999).

However, input/output linearization also produces an uncontrolled nonlinear subsystem such that the original system (2) can be decomposed as follows:

$$\begin{aligned} &\dot{\eta} = F\eta + G\mu \\ &\dot{z} = Q(\eta, z) \end{aligned} \tag{9}$$

where $z \in Z$ are the uncontrolled states, η are the controlled (or output) states as defined before, and a vector field Q is assumed to be locally Lipschitz continuous. With some appropriate control u (i.e., μ from (6)), η, becomes zero and the uncontrolled nonlinear dynamics (9) becomes $\dot{z} = Q(0, z)$, which is called a zero dynamics. In this case, the manifold z becomes invariant.

To ensure the stability of the continuous dynamics (9) (or equivalently (2)), for ***exponentially stabilizing control Lyapunov function*** (ES-CLF) V: $H \rightarrow \mathbb{R}$ which is a continuously differentiable function, there exist c_1, c_2, and c_3 are some positive real numbers such that (Ames et al. 2014, Blanchini and Miani 2008),

$$c_1\|\eta\|^2 \leq V(\eta) \leq c_2\|\eta\|^2 \tag{10}$$

$$\inf_{\mu \in U}[L_f V(\eta, z) + L_g V(\eta, z)\mu + c_3 V(\eta)] \leq 0 \tag{11}$$

Note that if there are no positive real c_1, c_2, and c_3, exponential stability of (9) is not guaranteed. When $V(\eta)$ is an ES-CLF, the solution of (9) satisfies the following:

$$\|\eta\| \leq \sqrt{\frac{c_2}{c_1}} e^{-\frac{c_3}{2}t} \|\eta(0)\|. \tag{12}$$

However, the definition of ES-CLF is not enough to handle the stability of the hybrid dynamics (3). That is, the convergence rate of ES-CLF is not fact enough to handle the hybrid dynamics. Therefore, ES-CLF is extended to have sufficiently rapid convergence rate to overcome the repulsion due to impact from the discrete dynamics. For ***rapidly exponentially stabilizing control Lyapunov function*** (RES-CLF), a one-parameter family of continuously differentiable functions V_ϵ: $H \rightarrow \mathbb{R}$, there exist c_1, c_2, and c_3 are some positive real numbers such that

$$c_1\|\eta\|^2 \leq V_\epsilon(\eta) \leq c_2\|\eta\|^2 \tag{13}$$

$$\inf_{\mu \in U}[L_f V_\epsilon(\eta, z) + L_g V_\epsilon(\eta, z)\mu + \frac{c_3}{\epsilon} V_\epsilon(\eta)] \leq 0 \tag{14}$$

for all $0 \leq \epsilon \leq 1$, where c_1, c_2, and c_3 are some positive real numbers that are related to the convergence rate to the origin. When u_ϵ is any Lipschitz continuous feedback control law and $V_\epsilon(\eta)$ is an RES-CLF, the (13) and (14) imply that the solution of (9) satisfies the following:

$$V_\epsilon(\eta(t)) \leq e^{-\frac{c_3}{\epsilon}t} V_\epsilon(\eta(0)) \tag{15}$$

$$\|\eta\| \leq \frac{1}{\epsilon}\sqrt{\frac{c_2}{c_1}} e^{-\frac{c_3}{2\epsilon}t} \|\eta(0)\|. \tag{16}$$

Therefore, the rate of exponential convergence can be directly controlled by ϵ (Ames et al. 2014). Finally, the following optimization problem can be configured such that control effort can be minimized using quadratic program (QP):

$$u^* = \arg\min_{\mu} \mu^T \mu \tag{17}$$

$$L_f V_\epsilon(\eta, z) + L_g V_\epsilon(\eta, z)\mu + \frac{c_3}{\epsilon} V_\epsilon(\eta) \le 0 \tag{18}$$

Then, we can eventually come to the following QP-based CLF formulation in terms of actual input u.

$$u^* = \arg\min_{u} u^T A^T A u + 2L_f^T A u \tag{19}$$

$$L_f V_\epsilon(\eta, z) + \frac{c_3}{\epsilon} V_\epsilon(\eta) + L_g V_\epsilon(\eta, z)\,(L_f + Au) \le 0 \tag{20}$$

It should also be noted that the advantage of using QP based controller is that various constrains which are affine in the input can be imposed in addition to the stability constraints as opposed to min-norm control or LQR. In the next section, the other constrains that will be used for ZMP-based walking will be briefly introduced.

CONTROLLING ROBOT LOCOMOTION WITH ZMP CONSTRAINTS

To control robotic locomotion using ZMP methods, we can consider two methods which will be briefed here: (1) nonlinear robot control with ZMP constraints using RES-CLF and (2) linear inverted pendulum model for COM trajectory generation using MPC. And finally, later in this section, we will introduce how these two methods can be unified. The details can be found in (Chao et al. 2016).

Nonlinear Robot Control with ZMP Constraints

We assume that all movements of the robot happen only in the sagittal plane. Then, the ZMP constrains for dynamic balance can be described in terms of the ground reaction force (GRF), as shown in the following:

$$x_z^- \le x_z \left(= -\frac{\tau_y}{F_z}\right) \le x_z^+ \tag{21}$$

where x_z^- and x_z^+ are the lower and upper ends of support polygon, x_z is the ZMP position, τ_y is an ankle torque in the sagittal plane and F_z is the vertical GRF. To achieve robotic bipedal locomotion while satisfying the constraint of (21) and minimizing the control effort, CLF-QP with RES-CLF that was introduced in the previous section can be used. CLF-QP can then be formulated in the following way:

$$\bar{u}^* = \arg\min_{\bar{u}} \bar{u}^T H_{CLF}\, \bar{u} + f_{CLF}^T\, \bar{u} \tag{22}$$

$$\text{s.t. } \dot{V}(x) \le -\epsilon V(x) \tag{23}$$

$$x_z^- \le -\frac{\tau_y}{F_z} \le x_z^+. \tag{24}$$

where H_{CLF} is the quadratic objective function, f_{CLF} is the linear objective function, (23) is RES-CLF condition, (24) is ZMP constraint and $x = [q\ \dot{q}]^T$ and $\bar{u}$ is the control input from the extended robot dynamics (extended from Eq. (1)) $D(q)\,\ddot{q} + H(q,\ \dot{q}) = [B\ J_h^T]\begin{bmatrix} u \\ F \end{bmatrix} \equiv \bar{B}\bar{u}$, where J_h is Jacobian matrix of the contact constraint $h(q)$, and F is the ground reaction force vector.

However, even though the inequality (24) is satisfied, the robot can still enter the states for which there is no feasible solution to the ZMP constraints (24). In the next subsection, we will introduce another method using linear inverted pendulum model and MPC.

Linear Inverted Pendulum Model for COM Trajectory Generation

To simplify the ZMP tracking, one can generate a COM trajectory with the linear inverted pendulum model (LIPM) that tracks the desired ZMP trajectory (or walking pattern generation). For this pattern generation, model predictive control (MPC) has been used in the literature. Since LIPM assumes a constant COM height, we have the following relation:

$$\ddot{x}_c = \frac{g}{z_0}(x_c - x_z) \equiv \omega^2(x_c - x_z) \tag{25}$$

where z_0 is the constant COM height, x_c is the horizontal component of the COM, x_z is the ZMP. The continuous equation (25) is then discretized to implement MPC as following:

$$x_{t+1} = \begin{bmatrix} 1 & \Delta T & 0 \\ \omega^2 \Delta T & 1 & -\omega^2 \Delta T \\ 0 & 0 & 1 \end{bmatrix} x_t + \begin{bmatrix} 0 \\ 0 \\ \Delta T \end{bmatrix} u_t \tag{26}$$

where $x_t = [x_{ct}\ \dot{x}_{ct}\ x_{zt}]^T$, and ΔT is the sampling time. If N time step horizon is assumed, we have the following linear equation:

$$\bar{X} = \bar{A}x_{t_0} + \ \bar{B}\bar{U} \tag{27}$$

where $\bar{X} = [x^T_{t_0+1} \dots x^T_{t_0+N}]^T$ and $\bar{U} = [u^T_{t_0+1} \dots u^T_{t_0+N}]^T$ are the sequence of states and control inputs, respectively, and $\bar{A}$ and $\bar{B}$ are derived from (26). The predictive control can be computed from the following optimization problem:

$$\bar{U}^* = \arg\min_{\bar{U}} \ \bar{U}^T H_P \bar{U} + f_P^T \bar{U} \tag{27}$$

$$\text{s.t. } C\bar{U} \le \bar{d} \tag{28}$$

where ZMP constraint is expressed in the linear constraint (28). The details of constraints and objective function will be explained in the next subsection where the unification will be introduced. Even though the pattern generation for LIPM via MPC has advantage of real-time implementation, the oversimplification of LIPM may generate control sequence that may not be feasible for the full nonlinear dynamics. Also, the ZMP trajectory generated for LIPM may not result in a feasible ZMP trajectory for the full nonlinear dynamics. To overcome these issues for both nonlinear robot control with ZMP constraints and LIPM with MPC, we will introduce the unification of both methods in the next subsection.

Unification of Pattern Generation and Locomotion via CLF-QP

Setup for Nonlinear CLF-QP

In this subsection, we will combine nonlinear CLF-QP with LIPM-MPC. The setup of CLF-QP for ZMP-based walking was already introduced in Section **Control Lyapunov Function-Based Quadratic Program**. For ZMP-based walking, we need relative degree two output function (4) for the input/output linearization. The RES-CLF for the linearized dynamics (7) is used as the following form (Ames et al. 2014),

$$V(x) = V(\eta) = \eta^T P \eta \tag{29}$$

where P can be obtained from the solution of the continuous time algebraic Riccati equation (CARE):

$$F^T P + PF - PGG^T P + Q = 0 \tag{30}$$

where $Q = Q^T > 0$, and $P = P^T > 0$. As mentioned in Section **Control Lyapunov Function-Based Quadratic Program**, exponentially stabilizing η with the convergence rate $\epsilon > 0$, η must satisfy the following condition:

$$\dot{V}(\eta) = L_f V(\eta) + L_g V(\eta) u \leq -\epsilon V(\eta) \tag{31}$$

where

$$L_f V(\eta) = \eta^T (F^T P + PF)\eta$$

$$L_g V(\eta) = 2\eta^T PG \tag{32}$$

In this setup, we assume the following for the output functions: (1) the height of COM is set to initial height, $y_{z_{COM}} = z_{COM}(q_0)$, (2) torso angle is fixed with respect to the inertial frame, $y_{\theta_{torso}}(q) = 0$, (3) during the single support phase, the orientation of swing foot, $y_{\theta_{foot}}(q)$, is zero (i.e., horizontal to the ground), and the horizontal and vertical positions of swing foot $y_{x_{foot}}(q)$ and $y_{z_{foot}}(q)$ are smooth polynomial functions where the velocities and accelerations at the beginning and at the end of swing phase are zero.

Here, we introduce all constraints used for CLF-QP. For exponential stability (31), we relaxed the constraint to improve the feasibility when there are multiple active constraints. The relaxation is penalized in the objective function.

$$\dot{V}(\eta) = L_f V(\eta) + L_g V(\eta) u \leq -\epsilon V(\eta) + \delta \tag{33}$$

Torque for each motor can exert was limited such that $|u| \leq u_{max}$. For ZMP constraint, we used (21). We also made sure that normal force at the foot is always positive, $F_z \geq 0$. The contact constraints are imposed in the equation of motion as follows (Murray et al. 1994),

$$\begin{aligned}\ddot{q} = J_h^T \left[\left(J_h D(q)^{-1} J_h^T \right)^{-1} - I \right] \times \left(J_h D(q)^{-1} H(q, \dot{q}) - J_h^T \dot{J}_h \dot{q} \right) \\ - J_h^T \left[\left(J_h D(q)^{-1} J_h^T \right)^{-1} - I \right] J_h D(q)^{-1} u\end{aligned} \tag{34}$$

where $h(q)$ is the holonomic constraint of contact points, J_h is the Jacobian matrix of $h(q)$. The detailed derivation of (34) can be found in (Chao et al. 2016).

The objective function for nonlinear CLF-QP is similar to (19). One modification to (19) is that minimization for relaxation is added with some weighting as following:

$$u^* = \arg \min_{\bar{u}} \bar{u}^T A^T A \bar{u} + 2\, L_f^T A \bar{u} + p\delta^2 \tag{35}$$

Where $\bar{u}$ is defined the same as in (22), and p is the weighting factor for the relaxation δ.

Setup for LIPM via MPC

Since robotic walking alternates single support (SS) and double support (DS) phases, the target duration of MPC horizon has N discrete points where $N = (T_{SS} + T_{DS})/\Delta T$, and T_{SS}, T_{DS}, ΔT are single support time, double support time, and MPC sampling time, respectively. Therefore, the horizon naturally consists of three phases of either SS-DS-SS or DS-SS-DS depending on the current phase. For the LIPM-MPC constraints, one of the goals is to enforce the COM to reach the position x_c^{goal} at the end of the trajectory. Also, ZMP over the horizon has to remain within the support polygon. The details can be found in (Chao et al. 2016, Stephens and Atkeson 2010, Stephens 2011, Wieber 2006).

The objective function for MPC is to minimize the control effort, while achieving ZMP trajectory tracking and driving the COM position to the desired location for the next step. The objective function is given as follows:

$$\bar{U}^* = \arg\min_{\bar{U}^*} \omega_1 \bar{U}^T \bar{U} + \omega_2 \left|\bar{X}_z - \bar{X}_z^{goal}\right|^2 \tag{36}$$

where ω_1, ω_2 are the weighting factors, $\bar{X}_z^{goal}$ is the desired ZMP trajectory, and x_c^{goal} is the desired COM terminal position at $t = t_0 + N$.

Unified QP combining Pattern Generation and ZMP-based Walking Control

The QP which combines both pattern generation and ZMP-based locomotion with RES-CLF is given as follows:

$$\begin{aligned}
&\arg\min_{\bar{u}^*, \bar{U}^*, \delta^*} \frac{1}{2} \begin{bmatrix} \bar{u} \\ \bar{U} \\ \delta \end{bmatrix}^T \begin{bmatrix} H_{CLF} & 0 & 0 \\ 0 & H_p & 0 \\ 0 & 0 & p \end{bmatrix} \begin{bmatrix} \bar{u} \\ \bar{U} \\ \delta \end{bmatrix} + \begin{bmatrix} f_{CLF} \\ f_p \\ 0 \end{bmatrix}^T \begin{bmatrix} \bar{u} \\ \bar{U} \\ \delta \end{bmatrix} \\
&\quad \text{s.t.} \begin{bmatrix} A_{eq,CLF} & 0 & 0 \\ 0 & A_{eq,p} & 0 \end{bmatrix} \begin{bmatrix} \bar{u} \\ \bar{U} \\ \delta \end{bmatrix} = \begin{bmatrix} b_{eq,CLF} \\ b_{eq,p} \end{bmatrix} \\
&\qquad \begin{bmatrix} A_{iq,CLF} & 0 & -1 \\ 0 & A_{iq,p} & 0 \end{bmatrix} \begin{bmatrix} \bar{u} \\ \bar{U} \\ \delta \end{bmatrix} \leq \begin{bmatrix} b_{iq,CLF} \\ b_{iq,p} \end{bmatrix} \\
&\qquad (L_f^2 x_c + L_f L_g x_c \bar{u}) \frac{z_0}{g} - x_c = -x_z
\end{aligned} \tag{37}$$

where the last equality constraint is from (25). As a synthesis constraint, it aims to equate the COM acceleration, $\ddot{x}_c$, in full dynamics to the $\ddot{x}_c$ in LIPM so that the pattern generation will plan the COM with the acceleration feedback from full dynamics. By solving the QP for each time step, the instantaneous torque input for ZMP-based locomotion considering both output tracking and COM planning on-the-fly can be derived.

Simulation Result and Future Direction

The unified controller introduced in the previous subsection is implemented on the model of AMBER 3[1] which is a human-sized, planar, and fully actuated bipedal robot with 6 degrees of freedom. For the simulation, the following parameters were implemented (Table 1).

With these parameters, the unified controller was implemented combining nonlinear CLF-QP and LIPM-MPC using MATLAB (v2015a, MathWorks, Natick, MA). The results are shown in Figs. 1, 2, and 3. As the figures show, tracking performance seems satisfactory. However, compared to the controller solving LIPM-QP and CLF-QP separately, several adjustments of the unified QP controller has to be made to ensure

Table 1. Important simulation parameters.

Parameter	Value	Parameter	Value
T_{ss}	2s	T_{Ds}	1s
MPC sampling time ΔT	0.1s	Length of MPC horizon	3s
Step Length	10 cm	Stride Height	5 cm

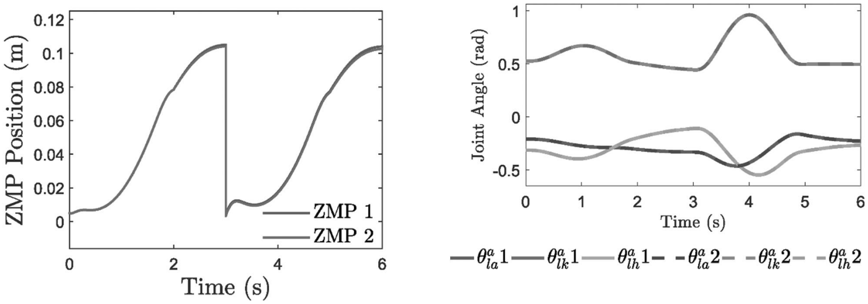

Figure 1. Comparison of ZMP trajectories (left) and joint tracking profiles (right) from two different simulations of the proposed method: (1) unified QP with terminal constraints on the COM, (2) unified QP without the terminal constraints.

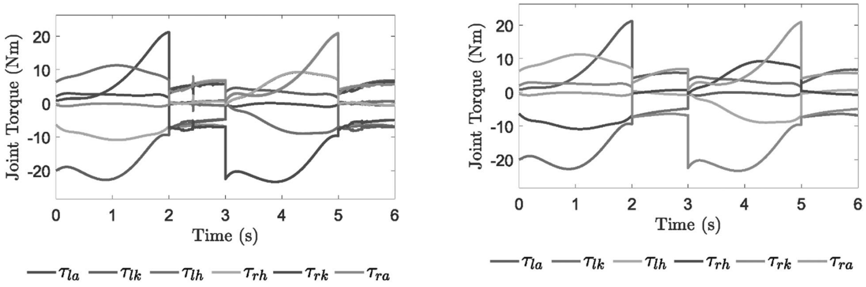

Figure 2. Joint torques from simulation of the proposed unified RES-CLF QP with (left) and without (right) terminal constraints on the COM.

[1] AMBER 3 was built in AMBER Lab led by Dr. Aaron Ames at Texas A&M University. Please note that AMBER Lab moved to Georgia Tech in Atlanta, GA in July 2015. Since then, AMBER 3 has been maintained, operated, and redesigned by Human Rehabilitation (HUR) Group led by Dr. Pilwon Hur at Texas A&M University.

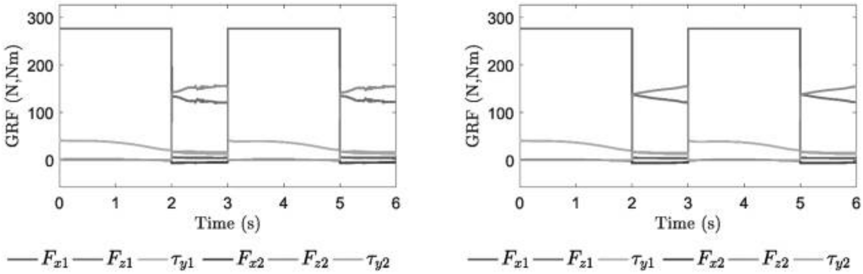

Figure 3. Ground reaction forces from simulation using the proposed unified RES-CLF QP with (left) and without (right) terminal constraints on the COM.

that the system would not be over-constrained. The first change was to remove x_c from the output vector in the unified framework because the resolved control input also minimized the cost function in unified pattern generation. Dropping of x_c could provide greater flexibility for integration with other tasks. Second, the COM terminal constraints were removed in the unified framework, since this would make the system over-constrained and the control input would lose continuity and cause chattering as shown in Figs. 2, and 3, although the derived joint trajectories and ZMP patterns (Fig. 1) are similar. Lastly, the direct ZMP feedback for updating x_{t_0} in real-time COM planning would easily cause the resolved x_c to diverge in the beginning of single support due to the impact in the full dynamics. As a result, only partial state feedback of the COM position and velocity from the full dynamics were applied. Future implementations will reason about hybrid system elements, such as impact equations, for a more general pattern generation scheme.

UNIFICATION OF CONTROLLER DESIGN USING NONLINEAR CONTROL METHODS FOR PUSH RECOVERY

As was mentioned in the Introduction section, nonlinear dynamic walking has benefits over the ZMP-based walking in terms of agileness, efficiency, and resemblance of the human walking. However, it lacks the robustness which is defined as the capacity to react to or the ability to keep balance against external perturbations. The causes of disturbances in robotic walking can be various, e.g., slippery surface, external push and uneven terrain. The importance of robust stabilization of the perturbed bipedal walking gait cannot be overemphasized in the situations where the gait stability and task performance of rescue robots, space robots, and any other service robots can be easily threatened via unexpected disturbances. However, despite these importance, robust stabilization of the perturbed bipedal walking is still a challenging research topic. In this section, we will consider unification of controller design via CLF-QP for push-recovery scenario.

In the biomechanics literature, humans are known to regulate the whole body angular momentum during walking (Herr and Popovic 2008, Popovic et al. 2004a, 2004b). Neptune and McGowan (2011) also reported that the regulation of whole-body angular momentum is essential to restoring and maintaining dynamic balance during walking. In the robotics community, several researchers have proposed control schemes exploiting angular momentum to recover balance of robots from external perturbations. Kajita et al. (2003b) presented a controller that regulated whole-body linear and angular momenta to the desired momenta. Goswami (2004) proposed to regulate the rate of angular momentum to zero in order to maintain stability.

As mentioned in Section **Control Lyapunov Function-Based Quadratic Program**, CLF-QP can be used to generate optimal controller in real time with guaranteed exponential stability. Furthermore, CLF-QP combined with the human-inspired control from the human walking data could provably result in stable and human-like robotic walking (Ames 2014). In this section, we introduce a control framework that unifies CLF-QP with human-inspired control and regulation of centroidal angular momentum, which we call CLF-K-QP. In the following subsections, we will show that the inclusion of centroidal angular momentum regulation in the CLF-QP with human-inspired control can robustly reject disturbance due to external perturbations while successfully tracking joint trajectories.

Angular Momentum in Bipedal Walking

In this subsection we briefly present angular momentum calculation. Detailed explanation can be found in (Orin and Goswami 2008). For each link *i*, linear and angular momenta about its COM can be computed as follows:

$$h_i = \begin{bmatrix} l_i \\ k_i \end{bmatrix} = \begin{bmatrix} m_i v_{c_i} \\ I_i^{COM}\,\omega_i \end{bmatrix} \tag{38}$$

where v_{c_i} and ω_i are linear velocity of link *i* COM and angular velocity of link *i*, respectively. m_i and I_i^{COM} are mass and inertia tensor with respect to the link *i* COM, respectively. Translating each momentum of link *i* to the origin of the local reference frame, with a distance from the origin to its center of mass as $\vec{p}_{c_i}$, and expressing the cross product using a skew symmetric matrix, $S(a)b = a \times b$, the momentum can be derived in matrix form as shown in the following equation:

$$h_i^O = \begin{bmatrix} l_i^O \\ k_i^O \end{bmatrix} = \begin{bmatrix} m_i I_{3\times 3} & m_i S(\vec{p}_{c_i})^T \\ \hat{I}_i & m_i S(\vec{p}_{c_i}) \end{bmatrix} \begin{bmatrix} \omega_i \\ v_i \end{bmatrix} = A_i^O \vec{V}_i \tag{39}$$

where $\vec{V}_i$ is the system velocity which contains ω_i and v_i of the link *i*, and $\hat{I}_i = I_i^{COM} + m_i S(\vec{p}_{c_i}) S(\vec{p}_{c_i})^T$.

Then, the collection of momentum for all links of the system can be given as follows:

$$h^O = \begin{bmatrix} h_1^O \\ \vdots \\ h_n^O \end{bmatrix} = diag[A_1^O \quad \ldots \quad A_n^O]\vec{V} = A^O \vec{V} \tag{40}$$

where

$$\vec{V} = [(\vec{V}_1)^T \cdots (\vec{V}_n)^T]^T \tag{41}$$

The velocity at the origin of the local coordinate system can be computed using a Jacobian with respect to the link's reference frame as follows:

$$\vec{V}_i = J_i \dot{q} \tag{42}$$

where q is the joint angle vector, and J_i is the Jacobian of the reference frame on link i. Combining (41) with (42) can yield the following equation:

$$\vec{V} = [(J_1)^T \ldots (J_n)^T]^T \dot{q} = J\dot{q} \tag{43}$$

Then, both (40) and (43) leads to:

$$h^O = A^O J\dot{q} = H^O\dot{q} \tag{44}$$

By projecting each momentum onto the COM, we get the following equation:

$$h^G = \sum_{i=1}^{n} X_i^{COM} h_i^O = \sum_{i=1}^{n} X_i^{COM} H_i^O \dot{q} \tag{45}$$

$$h^G = [X_1^{COM} \quad \ldots \quad X_n^{COM}] H_i^O \dot{q} = X^{COM} H_i^O \dot{q} = A^G \dot{q} \tag{46}$$

CLF-QP and Human-Inspired Control

Robot Dynamics

Figure 4 gives the coordinate system and definition of joint angles: $\theta = [\theta_{sa}, \theta_{sk}, \theta_{sh}, \theta_{nsh}, \theta_{nsk}, \theta_{nsa}]^T$. The equation of motion for the robot using Euler-Lagrange formula is given as follows:

$$D(\theta)\ddot{\theta} + H(\theta, \dot{\theta}) = Bu \tag{47}$$

where $D(\theta) \in \mathbb{R}^{6\times6}$ is the inertial matrix and $H(\theta, \dot{\theta}) \in \mathbb{R}^{6\times1}$ contains the terms resulting from the Coriolis effect and the gravity vector. The torque map $B = I_{6\times6}$ under the assumption that the robot is fully-actuated and the control input, u, is the vector of torque inputs. With the notation $x = [\theta; \dot{\theta}]$, we assumed to have the following affine control system:

$$\dot{x} = f(x) + g(x)u \tag{48}$$

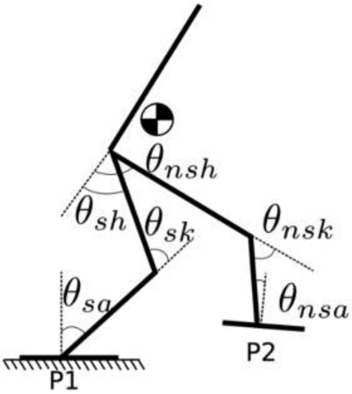

Figure 4. Flat footed robot model.

Human-Inspired Outputs

With the goal of achieving human-like robotic walking, we need to drive the actual robot output $y^a(\theta)$ to the desired human output $y^d(t, \alpha)$ that are represented by the canonical walking function (CWF):

$$y_{CWF} = e^{\alpha_4 t}(\alpha_1 \cos(\alpha_2 t) + \alpha_3 \sin(\alpha_2 t)) + \alpha_5 \tag{49}$$

Instead of using experimental human walking data directly, we are interested in seeking a low dimensional representation of human walking. It is known that hip position increases linearly with time whereas other human walking data can be fitted by CWF in the form as in (49) (Ames 2014, Sinnet et al. 2014). For our model (see Fig. 4), a total of 6 outputs are of interest for the fully-actuated 7-link bipedal robot. Therefore, we introduce human-inspired outputs as follows:

$$y(\theta, \dot{\theta}, \alpha) = \begin{bmatrix} y_1(\theta, \dot{\theta}, \alpha) \\ y_2(\theta, \alpha) \end{bmatrix} = \begin{bmatrix} y_1^a(\theta, \dot{\theta}) - v_{hip} \\ y_2^a(\theta) - y_2^d(\rho(\theta), \alpha) \end{bmatrix} \tag{50}$$

where $y_1(\theta, \dot{\theta}, \alpha)$ is the relative degree one output which is the difference between the actual hip velocity $y_1^a(\theta, \dot{\theta})$ and the desired hip velocity v_{hip}. The vector $y_2(\theta, \alpha)$ contains the 5 relative degree two human-inspired outputs which are the differences between the actual outputs $y_2^a(\theta)$ and the desired output $y_2^d(\rho(\theta), \alpha)$. Based on the linearity of hip position over time, $\rho(\theta)$ is utilized as a time parameterization of the given human walking gait.

Utilizing these outputs, the human-inspired controller can be utilized to drive both $y_1 \rightarrow 0$ and $y_2 \rightarrow 0$ in a provably exponentially stable fashion for the continuous dynamics. However, the robot may be "thrown-off" from the designed trajectory when impacts occur. This motivates the introduction of the partial hybrid zero dynamics (PHZD) constraints aiming to yield a parameter set α that ensures the tracking of relative degree two outputs will remain invariant even after the impacts. In particular, with the partial zero dynamics (PZD) surface defined as:

$$\mathbf{PZ}_\alpha = \{(\theta, \dot{\theta}) \in TQ : y_2(\theta, \alpha) = 0, L_f y_2(\theta, \alpha) = 0\} \tag{51}$$

The PHZD constraint can be explicitly stated as:

$$\Delta_R(S_R \cap \mathbf{PZ}_\alpha) \subset \mathbf{PZ}_\alpha \tag{52}$$

where Δ_R and S_R are the reset map and switching surface of the robot model, respectively.

Human-Inspired Control

With the human-inspired output functions defined in the previous subsubsection, we can perform input/output linearization. The details of this process are already introduced in Section **Control Lyapunov Function-Based Quadratic Program**. Since we have both relative degree one output and relative degree two output vectors, the input/output relation is the same as (5) and the feedback control is the same as (6). The linearized equation (7) has Lyapunov function defined as in (29) that is related to the solution of CARE (30). The Lyapunov function (29) is a RES-CLF once it satisfies the constraint (14). Finally, CLF-QP is defined with the cost function (17) or (19).

Unification of CLF-QP and Angular Momentum Regulator

In the previous subsection, we have already defined CLF-QP. In this subsection, angular momentum regulator will be combined via a QP. That is, the new controller (CLF-K-QP) adds a Lyapunov constraint to track the desired values of centroidal angular momentum as another constraint in the QP. As derived before, the centroidal angular momentum at the COM is expressed as follows:

$$h^G = A^G \dot{q} \tag{53}$$

Since the centroidal angular momentum is of interest in this application, the linear momentum will be discarded from now on.

$$k^G = [0_{3\times3} \;\; I_{3\times3}] h^G = Kx \tag{54}$$

As usual, the dynamics of the system is expressed as in (48). To track a desired angular momentum R, it is possible to construct a Lyapunov function based on the actual angular momentum as follows:

$$V_k = (k^G - R)^T (k^G - R) = (Kx - R)^T (Kx - R) \tag{55}$$

To ensure exponential convergence, the following condition must be met.

$$L_f V_k(x) + L_g V_k(x) u \leq -\frac{\gamma_k}{\epsilon_k} V_k(x) \tag{56}$$

where γ_k and ϵ_k are both positive and determine the rate of convergence. γ_k is a constant while ϵ_k can be changed by the control designer. A smaller ϵ_k would increase the convergence rate but it may affect feasibility of the optimization.

Since the system is already fully-actuated, we relax the tracking of CWF with the insertion of δ_1 and δ_2 as shown in (57, 58, 59) to compensate for disturbances in angular momentum at the cost of losing tracking performance. Therefore, an optimal $u*$ that tracks the CWF and the centroidal angular momentum is given by:

$$u* = \arg\min_u u^T A^T A u + 2 L_f^T A u + p_1 \delta_1^2 + p_2 \delta_2^2 \tag{57}$$

s.t.

$$\varphi_0(q, \dot{q}) + \varphi_1(q, \dot{q})\,(Au + L_f) \leq \delta_1 \tag{58}$$

$$\varphi_0^k(q, \dot{q}) + \varphi_1^k(q, \dot{q})^T u \leq \delta_2 \tag{59}$$

where

$$\begin{aligned} \varphi_0(\eta) &= L_f V(\eta) = \frac{\gamma}{\varepsilon} V(\eta) \\ \varphi_1(\eta) &= L_g V(\eta)^T \end{aligned} \tag{59}$$

$$\varphi_0^k = L_f V_k(x) + \frac{\gamma_k}{\epsilon_k} V_k(x) \tag{60}$$

$$\varphi_1^k = L_g V_k(x)^T \tag{61}$$

and δ_1 and δ_2 are positive slack variables inserted into constraints (58) and (59) to relax the conditions on tracking and angular momentum regulation in order to find feasible solution.

To compute the desired centroidal angular momentum R, it is possible to compute the ideal states from the PHZD surface which represents a lower dimensional presentation of the system. These ideal states can be plugged into (46) to compute R which will then be plugged into the optimization shown in (59).

Simulation Result and Future Direction

In the simulation, we assumed that the robotic model is perturbed by an external and impulsive force $F_{ext} = 700N$ horizontally at the torso in the direction of movement for $t = 0.05s$ and was applied when $\rho(\theta) = 0.21$. The proposed controller CLF-K-QP is compared with CLF-QP. In order to have specific metrics of performance we use the root-mean-square (RMS) values to compare the tracking performance for the different quantities we are interested in.

The results are shown in Figs. 5, 6 and Table 2. The principal differences found during an external push include torso angle, angular momentum and hip velocity (Fig. 6). After the disturbance, all of torso angle, centroidal angular momentum and hip velocity suffered from a significant deviations from their nominal trajectories. The CLF-QP controller tried to recover joint trajectory tracking for the torso back to its desired value faster compared to CLF-K-QP. However, CLF-K-QP did not immediately drive the torso angle to its desired trajectory as shown in Fig. 6a. As expected, CLF-K-QP performed better in tracking of angular momentum to its desired value compared to CLF-QP (Fig. 6b). Interestingly, CLF-K-QP showed better performance in hip velocity

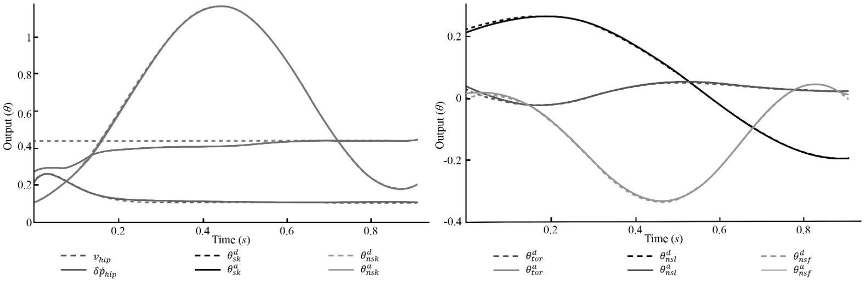

Figure 5. Outputs of unperturbed system. Note that superscript *d* represents desired trajectory whereas superscript *a* represents actual trajectory.

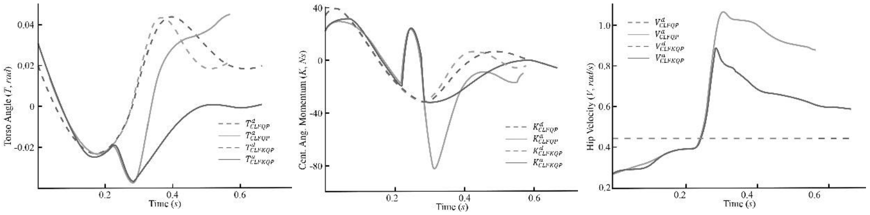

Figure 6. Torso angle evolution for one perturbed step. Note that superscript *d* represents desired trajectory whereas superscript *a* represents actual trajectory.

Table 2. Controller comparison.

RMS Values	**CLF-K-QP**			**CLF-QP**		
	Angular Momentum (Ns)	Hip Velocity (m/s)	Torso Angle (rad)	Angular Momentum (Ns)	Hip Velocity (m/s)	Torso Angle (rad)
Push	12.919	0.289	0.008	24.965	0.380	0.021
Norm	3.496	0.064	0.004	5.225	0.075	0.005

compared to CLF-QP (Fig. 6c). When perturbation as applied, hip velocity increased drastically for both controllers. However, CLF-K-QP outperformed CLF-QP in that CLF-K-QP had smaller peak hip velocity and kept the hip velocity significantly close to the desired hip velocity compared to CLF-QP. What is more inspiring is that even though CLF-QP was better at tracking torso angle, hip velocity tracking became worse. Based on these observations, we may summarize the following findings: (1) when the system is perturbed, CLF-K-QP put more effort in regulating angular momentum, (2) CLF-K-QP relaxes tracking of torso angle, and (3) the system can maintain its walking speed more robustly with CLF-K-QP. These three observations may not seem to work independently. For example, relaxed torso angle due to CLF-K-QP may have counteracted the undesired and excessive angular momentum, and eventually the system could maintain the walking speed better, whereas CLF-QP made the system stiffer. This may be similar to the natural behavior of human beings how protract their torso and outstretch their arms out to counteract the debilitating angular moments.

CONCLUSION

This chapter presented how robotic bipedal walking control can use the framework developed for locomotion and manipulation to unify ZMP-based tasks, force-based tasks, and angular momentum regulation into the same single framework via quadratic programs utilizing control Lyapunov functions. We introduced two possible examples where the unification can be applied: (1) ZMP-based pattern generation and locomotion, and (2) nonlinear dynamics with push recovery. In both examples, the end result was a single quadratic program that can dynamically balance all of these disparate objectives. This approach can be implemented as a single algorithm and allow for more holistic implementation of controllers on bipedal robotic walking. Practically, CLF-QP has been implemented in real-time to experimentally achieve 2D bipedal robotic walking. However, the speed of the QP depends on the feasibility of the constraints, so future study investigating the relation between feasibility of constraints and QP computation will be an interesting problem.

REFERENCES

Ames, A. 2011. First steps toward automatically generating bipedal robotic walking from human data. pp. 89–116. *In*: K. Kozłowski (ed.). Robot Motion and Control 2011, Vol. 422. London: Springer London, 2012.

Ames, A. and M. Powell. 2013. Towards the unification of locomotion and manipulation through control lyapunov functions and quadratic programs. pp. 219–240. *In*: D. C. Tarraf (ed.). Control of Cyber-Physical Systems. Heidelberg: Springer International Publishing, 2013.

Ames, A. 2014. Human-inspired control of bipedal walking robots. IEEE Trans. Automat. Contr., 59(5): 1115–1130, May 2014.

Ames, A., K. Galloway, K. Sreenath and J. Grizzle. 2014. Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics. IEEE Trans. Automat. Contr., 59(4): 876–891, 2014.

Blanchini, F. and S. Miani. 2008. Set-Theoretic Methods in Control. Basel: Birkhäuser, 2008.

Chao, K., M. Powell, A. Ames and P. Hur. 2016. Unification of locomotion pattern generation and control lyapunov function-based quadratic programs. pp. 3910–3915. *In*: Boston, M.A. American Control Conference, 2016.

Goswami, A. and V. Kallem. 2004. Rate of change of angular momentum and balance maintenance of biped robots. pp. 3785–3790. *In*: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04, 2004, Vol. 4.

Grizzle, J. W. and E. R. Westervelt. 2008. Hybrid zero dynamics of planar bipedal walking. pp. 223–237. *In*: Analysis and Design of Nonlinear Control Systems, Springer, 2008.

Herr, H. and M. Popovic. 2008. Angular momentum in human walking. J. Exp. Biol., 211(Pt 4): 467–81, Feb. 2008.

Herzog, A., L. Righetti, F. Grimminger, P. Pastor and S. Schaal. 2014. Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics. pp. 981–988. *In*: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014.

Isidori, A. 1995. Nonlinear Control Systems, 3rd ed. London: Springer, 1995.

Kajita, S., F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi and H. Hirukawa. 2003a. Resolved momentum control: humanoid motion planning based on the linear and angular momentum. pp. 1644–1650. *In*: Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No. 03CH37453), 2003, Vol. 2.

Kajita, S., F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi and H. Hirukawa. 2003b. Biped walking pattern generation by using preview control of zero-moment point. pp. 1620–1626. *In*: 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422), 2003, Vol. 2.

Kuindersma, S., F. Permenter and R. Tedrake. 2014. An efficiently solvable quadratic program for stabilizing dynamic locomotion. pp. 2589–2594. *In*: 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014.

Macchietto, A., V. Zordan and C. R. Shelton. 2009. Momentum control for balance. *In*: ACM SIGGRAPH 2009 papers on - SIGGRAPH '09, 2009, 28(3): 1.

McGeer, T. 1990. Passive dynamic walking. Int. J. Rob. Res., 9(2): 62–82, April 1990.

Morris, B. and J. W. Grizzle. 2005. A restricted poincaré map for determining exponentially stable periodic orbits in systems with impulse effects: Application to bipedal robots. pp. 4199–4206. *In*: Proceedings of the 44th IEEE Conference on Decision and Control, 2005.

Murray, R. M., Z. Li and S. S. Sastry. 1994. A Mathematical Introduction to Robotic Manipulation, 1st ed. Boca Raton, 1994.

Neptune, R. R. and C. P. McGowan. 2011. Muscle contributions to whole-body sagittal plane angular momentum during walking. J. Biomech., 44(1): 6–12, Jan. 2011.

Orin, D. E. and A. Goswami. 2008. Centroidal momentum matrix of a humanoid robot: Structure and properties. pp. 653–659. *In*: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008.

Popovic, M., A. Hofmann and H. Herr. 2004a. Zero spin angular momentum control: definition and applicability. *In*: 4th IEEE/RAS International Conference on Humanoid Robots, 2004, 1: 478–493.

Popovic, M., A. Hofmann and H. Herr. 2004b. Angular momentum regulation during human walking: biomechanics and control. *In*: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04, 2004, 3: 2405–2411.

Posa, M. and R. Tedrake. 2013. Direct trajectory optimization of rigid body dynamical systems through contact. pp. 527–542. *In*: E. Frazzoli, T. Lozano-Perez, N. Roy and D. Rus (eds.). Algorithmic Foundations of Robotics X. Vol. 86. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

Powell, M., E. Cousineau and A. Ames. 2015. Model predictive control of underactuated bipedal robotic walking. pp. 5121–5126. *In*: 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015.

Saab, L., O. E. Ramos, F. Keith, N. Mansard, P. Soueres and J. -Y. Fourquet. 2013. Dynamic whole-body motion generation under rigid contacts and other unilateral constraints. IEEE Trans. Robot., 29(2): 346–362, April 2013.

Sastry, S. 1999. Nonlinear Systems—Analysis, Stability, and Control. New York: Springer, 1999.

Sinnet, R., S. Jiang and A. Ames. 2014. A human-inspired framework for bipedal robotic walking design. Int. J. Biomechatronics Biomed. Robot., Feb. 2014.

Stephens, B. J. and C. G. Atkeson. 2010. Push recovery by stepping for humanoid robots with force controlled joints. pp. 52–59. *In*: 2010 10th IEEE-RAS International Conference on Humanoid Robots, 2010.

Stephens, B. 2011. Push Recovery Control for Force-Controlled Humanoid Robots, Carnegie Mellon University, 2011.

Vukobratovic, M. and B. Borovac. 2004. ZERO-MOMENT POINT—THIRTY FIVE YEARS OF ITS LIFE. Int. J. Humanoid Robot., 01(01): 157–173, March 2004.

Wieber, P. 2006. Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. pp. 137–142. *In*: 2006 6th IEEE-RAS International Conference on Humanoid Robots, 2006.

Zhao, H., M. Powell and A. Ames. 2014. Human-inspired motion primitives and transitions for bipedal robotic locomotion in diverse terrain. Optim. Control Appl. Methods, 35(6): 730–755, Nov. 2014.

Zhao, H., W. -L. Ma, A. Ames and M. Zeagler. 2014. Human-inspired multi-contact locomotion with AMBER2. pp. 199–210. *In*: 2014 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS), 2014.