Chapter 4.5

Template Models for Control

Patrick M. Wensing[¶] **and Shai Revzen**[‖]

[¶]*Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, IN, United States* [‖]*Electrical Engineering and Computer Science and Ecology and Evolutionary Biology, University of Michigan, Ann Arbor, MI, United States*

4.5.1 INTRODUCTION

Within integrative biology, template models have enabled us to gain insight and understanding into a variety of complex motor control tasks. Indeed, we see that despite the relative complexity of our musculoskeletal systems, locomotion is often well modeled conceptually with relatively few degrees of freedom. Species from crabs to kangaroos bounce in a dynamically similar fashion, well described by the Spring-Loaded Inverted Pendulum (SLIP) template model. While the SLIP only captures the role of spring-leg operation in governing the center of mass (CoM) dynamics, in principle, increasingly complex template models may be sought and developed to *describe* increasingly rich aspects of motion. However, the importance of conceptual models is hypothesized to extend well beyond merely these *descriptive* capabilities.

In addition, template models may be used as *prescriptive* models of locomotion, providing low dimensional dynamic targets for closed-loop control in biological and robotic systems alike. It has been hypothesized that the cognitive processes regulating locomotion may include reasoning centered around reduced dimensional subsets of our full dynamics – template dynamics that capture the most salient aspects of a locomotory behavior (Full and Koditschek, 1999). This conjecture is a central tenant of the templates and anchors hypothesis from Chapter 3. These principled reductions capture universal characteristics of locomotion and may play an important role in enabling animals to generalize dynamic performance across such a wide range of scenarios in nature. This section aims to illuminate how conceptually similar reductions may be applied within control of our robots, enabling versatile locomotion with template models for control.

Before we begin, why should one consider the use of these reductive models for control when performance guarantees in the full state space may be provided by other control techniques in this chapter? At present, many methods with performance guarantees are not yet computationally viable for real-time application in the high-dimensional state space of our robots. Some methods, for instance those in Subchapter 4.7, provide certificates of stability and robustness through offline analysis. However, these guarantees are often only valid in a narrow region of the state space. The hybrid and nonlinear dynamics in our high-degree-of-freedom (DoF) robots challenge the development of guarantees that hold more broadly. These challenges collectively motivate the use of template models for control: by addressing an important subset of the dynamics, template-based control provides computational and analytical advantages that enable real-time computation and simplify control system analysis.

As we start to view template models for control, it is important to understand the fundamental differences from the use of template models in biology. Within biology, it is a role of the integrative biologist to *discover* reduced dimensional template dynamics embedded in human or animal motion. These template dynamics may take the form

$$\dot{\mathbf{x}}_T = \mathbf{f}(\mathbf{x}_T)$$

where $\mathbf{f}(\cdot)$ captures the effects of closed-loop sensorimotor control, and $\mathbf{x}_T$ represents the state of the template. Rather than discover existing dynamics, within robotics, it is instead the goal to *synthesize* closed-loop dynamics through real-time control. For an uncontrolled robot, it is possible that no template dynamics exist in advance. As a result, the specific aim of template-based control is to *realize* the behavior of a template through feedback in the full model. To guide the design of this feedback, we view templates themselves as controlled dynamic systems

$$\dot{\mathbf{x}}_T = \mathbf{f}(\mathbf{x}_T, \mathbf{u}_T) \tag{4.5.1}$$

with template control inputs $\mathbf{u}_T$ used to shape the closed-loop response. By strategically crafting closed-loop template controls that achieve a high-level goal (a desired running speed, or recovery from a push disturbance, for instance), anchoring a template imparts satisfaction of these performance objectives in the full model.

4.5.1.1 A Design Process for Template-Based Control

Despite the wide range of applications for template models in control, the development of template-based controllers generally follows a common workflow. This design process can be broken down into three rough steps as depicted in Fig. 4.5.1: template selection, template control, and establishing the template/anchor relationship. Template selection entails the choice or design of a reduced dimensional control dynamic system that captures the challenges of a motion control task while respecting the limitations of any target hardware. Following this selection, reduced dimensional control strategies may be designed for the template. Properties of the closed-loop dynamics for this template may

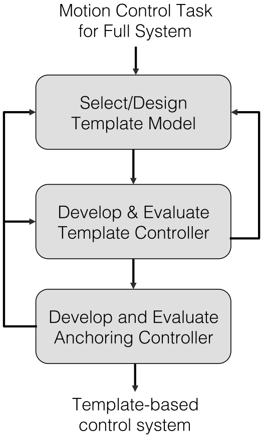
Motion Control Task for Full System
Select/Design Template Model
Develop & Evaluate Template Controller
Develop and Evaluate Anchoring Controller
Template-based control system

FIGURE 4.5.1 Three step design process to employ template models for control. The steps proceeding top to bottom represent a linear progression of designing a template-based control system. At each stage in the process, design decisions from earlier steps should be refined, as represented by the upward flows.

then be used to guide control in a high-DoF robot towards establishing a template/anchor relationship.

We further detail each of these steps in the sections that follow. Examples of developing walking and running controllers for a humanoid are presented to clarify the design steps. Despite the rather linear progression in the presentation of the examples, we note that the design process in general should be iterative. In practice, insights gained at each step should be used to continually inform refinements to decisions made earlier in the process. For example, following the development of a SLIP-based template controller in step (2), attempted application to a robot with high-impedance transmissions in step (3) may require redesign of the template model selected from step (1). At the conclusion of the process, the final output is a real-time control system, which may be used in a physical or simulated robot.

4.5.2 TEMPLATE MODEL SELECTION

In the first step of the process, an appropriate template for the motion control task must be selected or designed. This selection may come from study into motor control for biological systems, from previously applied models in robotics, or from personal insight into the fundamental physics of the motor control task. There is no explicitly right or wrong template model for a given task, and within robotics one need not be confined to those models that have appeared in biology. Indeed, fundamental differences between mechanical actuators and materials

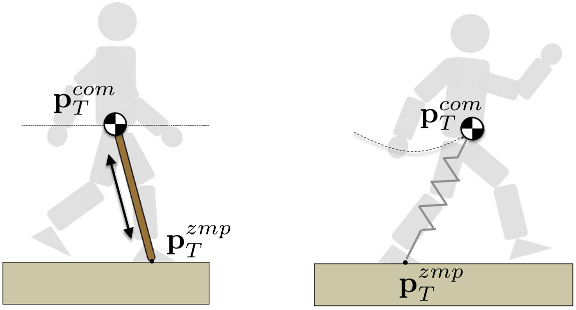

FIGURE 4.5.2 The linear inverted pendulum (LIP) and spring loaded inverted pendulum (SLIP) models are commonly employed templates for walking and running control in legged robots.

compared to biological muscles and tissues necessitates strategically principled bio-inspiration. As a result, the validity of a template model in robotics may only be judged based on the results of the final template-based control system. This may be viewed as a downside. However, we see a similar story for the use of template models in biology. Across both realms, a template model can never explicitly be proven as *the* correct template, but instead must be judged based on its usefulness to understand or control new physical behaviors.

Within the control of locomotion to date, popular template models have generally been simple *physical* dynamic systems. That is to say, systems whose dynamics follow the laws of Newtonian physics. Such models include the linear inverted pendulum model (LIP) as commonly used for walking control (Kajita et al., 2001, 2003; Herdt et al., 2010), the SLIP model used in running control (Blickhan, 1989; Seipel and Holmes, 2005; Garofalo et al., 2012; Wensing and Orin, 2013b), and many others. Examples below detail these common center of mass (CoM) template models further, highlighting their benefits to control dynamic walking (Kuindersma et al., 2015) and running (Wensing and Orin, 2013b). We will follow these examples though the design process of template models for control applied to a high-DoF robot in the sections to come. Through this development $\mathbf{x}_T \in \mathbb{R}^{n_T}$ and $\mathbf{u}_T \in \mathbb{R}^{m_T}$ will denote the state and controls for the template.

4.5.2.1 Linear CoM Models for Walking

Example 1 (LIP Model for Walking)**.** The Linear Inverted Pendulum (LIP) Model, shown in Fig. 4.5.2, captures the connection between the Center of Mass (CoM) and ZMP dynamics under an assumption of constant walking height. It is a commonly selected template to develop walking controllers. The model consists of a point mass m located at a position $\mathbf{p}_{com} = [x, y, z]^T \in \mathbb{R}^3$. The position of the mass is intended to represent the CoM of a system with more degrees of freedom. The model evolves according to forces $\mathbf{f}_{zmp} = [f_x, f_y, f_z]^T \in \mathbb{R}^3$ that act at a ZMP $\mathbf{p}_{zmp} = [p_x, p_y, p_z]^T \in \mathbb{R}^3$. For walking on level ground, p_z remains constant. As is standard with point-mass template models, the LIP assumes that ground reaction forces (GRFs) create no moment about the CoM. This condition defines the line of action for the GRF and requires $f_x/f_z = (x - p_x)/(z - p_z)$ with similar conditions for f_y. As a result, the template dynamics take the form:

$$m\ddot{x} = \frac{f_z}{z - p_z}(x - p_x), \tag{4.5.2}$$

$$m\ddot{y} = \frac{f_z}{z - p_z}(y - p_y), \tag{4.5.3}$$

$$m\ddot{z} = f_z - mg, \tag{4.5.4}$$

where g is the gravitational constant. Under the assumption of a constant height, $\dot{z} = \ddot{z} \equiv 0$. Thus, letting $\omega = \sqrt{g/(z - p_z)}$, we obtain

$$\ddot{x} = \omega^2(x - p_x), \tag{4.5.5}$$

$$\ddot{y} = \omega^2(y - p_y)\,. \tag{4.5.6}$$

These dynamics are unstable, with poles on the real axis at $s = \pm\omega$. The ZMP positions in the plane can be viewed as the control input $\mathbf{u}_T = [p_x, p_y]^T \in \mathbb{R}^2$ for this system, with state $\mathbf{x}_T = [x, y, \dot{x}, \dot{y}]^T$. For this template, a constraint that the ZMP must remain in a support polygon, discussed in Section 4.1.2.2, can be expressed directly through constraints on $\mathbf{u}_T(t)$.

For use in the next section, we note that the LIP dynamics can be expressed in state space:

$$\dot{\mathbf{x}}_T(t) = \mathbf{A}\,\mathbf{x}_T(t) + \mathbf{B}\,\mathbf{u}_T(t) \tag{4.5.7}$$

$$= \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \omega^2\mathbf{I} & \mathbf{0} \end{bmatrix} \mathbf{x}_T(t) + \begin{bmatrix} \mathbf{0} \\ -\omega^2\mathbf{I} \end{bmatrix} \mathbf{u}_T(t), \tag{4.5.8}$$

with the CoM accelerations considered as an output $\mathbf{y}_T = [\ddot{x}, \ddot{y}]^T \in \mathbb{R}^2$, where

$$\mathbf{y}_T(t) = \mathbf{C}\,\mathbf{x}_T(t) + \mathbf{D}\,\mathbf{u}_T(t) \tag{4.5.9}$$

$$= \begin{bmatrix} \omega^2\mathbf{I} & \mathbf{0} \end{bmatrix} \mathbf{x}_T(t) + \begin{bmatrix} -\omega^2\mathbf{I} \end{bmatrix} \mathbf{u}_T(t)\,. \tag{4.5.10}$$

Example 2 (ZMP/CoM Dynamics for Walking with Nonconstant Height)**.** In some cases, such as walking up stairs, the assumption of a fixed CoM height may be prohibitive within a template model for walking control. In the case that a nonconstant desired CoM height $z(t)$ is known in advance, the remaining dynamic equations for $x(t)$ and $y(t)$ are no longer time invariant. We again assume that the forces create no moment about the CoM. Considering that $f_z(t) = m\,g + m\,\ddot{z}(t)$, redefinition of $\omega := \omega(t)$ to be time varying according to

$$\omega(t) = \sqrt{\frac{g + \ddot{z}(t)}{z(t) - p_z}}$$

results in an linear time varying (LTV) system for the lateral (x, y) dynamics:

$$\ddot{x} = \omega(t)^2\,(x - p_x), \tag{4.5.11}$$

$$\ddot{y} = \omega(t)^2\,(y - p_y). \tag{4.5.12}$$

Despite the additional complexity, constraints on the ZMP are still readily addressed through constraints directly on $\mathbf{u}_T(t) = [p_x, p_y]^T$.

4.5.2.2 SLIP Models for Running

Example 3 (A Passive SLIP Model for Humanoid Running)**.** The Spring-Loaded Inverted Pendulum (SLIP) model, described in Subchapter 3.3, is a commonly selected template for control of running and hopping robots. In the SLIP, a point mass m alternates between periods of flight and stance. In flight, the point mass experiences ballistic physics $\ddot{\mathbf{p}}_T^{com} = \mathbf{g}$, where $\mathbf{g} \in \mathbb{R}^3$ is the gravity vector.

During flight, the leg may be repositioned through touchdown angles (θ, ϕ) in the forward and lateral directions. For application to humanoid control, these touchdown angles may be given with respect to an estimated hip position, $\mathbf{p}_T^{hip}$, at a fixed offset from the CoM as shown in Fig. 4.5.3. The hip offset may change from step to step to account for the leg in stance. In comparison to defining touchdown angles in spherical coordinates relative to the CoM, this alternate touchdown angle definition provides closer correspondence with the virtual leg angles in a robot that includes hip separation. As an alternative, the Cartesian position of the foot may be used directly as a control input in flight. This requires more careful consideration of virtual leg length constraints, but provides a close correspondence between the LIP and SLIP in terms of their states and controls.

SLIP stance begins at touchdown (TD) of its virtual leg, wherein a Hookean spring of stiffness k_s and initial length ℓ_0 imparts conservative forces on the mass. The stance dynamics follow

$$m\,\ddot{\mathbf{p}}_T^{com} = k_s(\ell_0 - \|\boldsymbol{\ell}\|)\,\hat{\boldsymbol{\ell}} + m\,\mathbf{g} \tag{4.5.13}$$

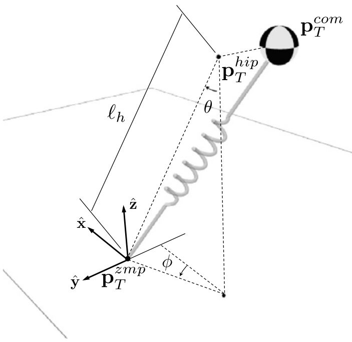

FIGURE 4.5.3 Leg angle definitions for 3D-SLIP applied to humanoid control. Leg angles are given with respect to a virtual hip position that exists at a fixed offset from the CoM during flight.

where $\boldsymbol{\ell} := \mathbf{p}_T^{com} - \mathbf{p}_T^{zmp} \in \mathbb{R}^3$ represents the virtual leg, $\hat{\boldsymbol{\ell}} \in \mathbb{R}^3$ the unit vector along the leg, and ℓ_0 the rest length computed at touchdown. Often, the ZMP position $\mathbf{p}_T^{zmp}$ is fixed following touchdown for simplicity even when the SLIP acts as a template for a robot with flat feet. Stance ends at liftoff (LO), wherein the model transitions back to flight. This energetically passive SLIP model can be controlled through touchdown angle inputs $\mathbf{u}_T[n] = [\theta, \phi]^T$ at each step n.

A unique event for each step (e.g. apex during flight) may be used to define a Poincaré section for study of the step-to-step dynamics. Letting $\mathbf{x}_T[n] = [z, \dot{x}, \dot{y}]^T$ at the nth apex, we define the step-to-step dynamics through

$$\mathbf{x}_T[n+1] = \mathbf{f}(n, \mathbf{x}_T[n], \mathbf{u}_T[n]) \tag{4.5.14}$$

where $\mathbf{f}(n, \mathbf{x}_T, \mathbf{u}_T)$ is a Poincaré return map.

Example 4 (An Active SLIP Model to Enable Energetic Transitions)**.** As one of a variety of potential extensions for this model, the examples in this chapter will consider modulating the stiffness of the SLIP leg once per step. Other valid actuation schemes for the SLIP may consider changes to its rest leg length, which are equivalent to the addition of a linear actuator in series with the spring. For simplicity, we consider a fixed stiffness k_{s_1} before maximum compression of the spring in stance, and a stiffness k_{s_2} following maximum compression. This consideration modulates the total energy E by $\Delta E = \frac{1}{2}(k_{s_2} - k_{s_1})(\ell_0 - \|\boldsymbol{\ell}\|)^2$, enabling changes in speed and height from step to step. Considering these stiffnesses as control variables along with the leg touchdown angles, the active SLIP may be controlled by selecting

$$\mathbf{u}_T[n] = [\theta, \phi, k_{s_1}, k_{s_2}]^T \tag{4.5.15}$$

at each step n. As in the LIP, a key aspect of the template is that the forces emanate from a well defined point $\mathbf{p}_T^{zmp}$ that must reside under the supporting contacts.

4.5.2.3 Perspectives of Template Model Selection

Each of these template models have a notable characteristic in that they follow the laws of Newtonian physics. Templates that follow the laws of Newtonian physics have benefits when targeted to legged robots, which themselves must follow the laws of physics. Physical template models may often easily be restricted to operate in a regime that is dynamically feasible for the target system. For instance, control inputs across each of these pendular models directly influence the center of pressure. Constraints for this point to remain within a support polygon are easy to formulate, which can be used to simplify template planning and control. To contrast, for a target system such as a humanoid, constraints on *its* control inputs (joint torques) to satisfy center of pressure constraints are much more complex, and in general are nonlinearly dependent on state.

Despite these benefits afforded in physical template models, such physicality is not a requirement. With such freedom, template models may be judiciously crafted to possess linear (Kajita et al., 2003; Kuindersma et al., 2014), integrable (Mordatch et al., 2010), polynomial (Park et al., 2015), or other simplified dynamics. These simplifications may be sought to further facilitate analysis and control. We again stress that, within robotics, the applicability of template models need not be grounded in biological or exact physical plausibility. Instead templates should ultimately be assessed based on the additional performance that they bring to motion control in physical robots.

Selections of control inputs $\mathbf{u}_T$ also have important reachability implications for the template. For instance, the selection of controls for a passive SLIP precludes the possibility to reach any states at higher or lower total energy levels. As template controls are designed to improve reachability properties, they need not match those controls available the full system. Design choices should be made, however, such that the resultant template dynamics are able to be replicated in the full model given its associated control authority.

4.5.3 TEMPLATE MODEL CONTROL

After selecting a candidate template for a motor control task, the second step in the process is to develop a control system for the template itself. The key idea in using template models for control is that the solution of this template control problem can provide guiding principles to solve the control problem for the full system. General methods to design control systems for template models are the same as those that might otherwise applied for the full model. As a benefit, approaches that might not scale (in time or space complexity) to the full model may become practical when considered in the scope of a template. Trajectory optimization for optimal control (Diehl et al., 2006) may offer applicability for real-time model-predictive control, feedback motion planning libraries (Tedrake et al., 2010) may scale to the dimensionality of template models, and feedforward knowledge-bases may be developed through offline computation (Wu and Geyer, 2013). Template model control has enabled many contemporary systems (Kuindersma et al., 2015; Feng et al., 2013; Pratt et al., 2012; Takenaka et al., 2009; Rezazadeh et al., 2015) to skirt Bellman's curse of dimensionality (Bellman, 1957) while maintaining real-time computation.

4.5.3.1 Control of Linear CoM Models for Walking

Many center of mass (CoM) templates have led to template controllers realized in modern walking humanoids. CoM templates driven by the zero moment point (ZMP) in discrete time (Kajita et al., 2003; Dimitrov et al., 2011) and continuous time (Tedrake et al., 2015) enable real-time optimal control solutions, highlighted in the example below. Whether in the LIP model, or its extension with a fixed CoM height trajectory, linear template dynamics simplify the development of optimal controllers. Other methods have used differential dynamic programming (DDP) to solve the optimal control problem locally when vertical CoM dynamics are allowed to vary but are not fixed a priori (Feng et al., 2013).

Example 5 (ZMP Preview Control Through LQR)**.** A common method of walking control for humanoid robots attempts to control the ZMP position underneath the feet. The linear inverted pendulum template model from Example 1 can enable efficient computation of ZMP controllers for use in a high-DoF robot. Let us assume that a desired ZMP trajectory $\mathbf{u}_T^d(t)$ has been generated in advance and that an assumption of constant CoM height is reasonable. Due to the unstable pole in the LIP dynamics, blindly using this ZMP trajectory as input to the LIP template will result in an unstable CoM motion, which must be handled through feedback in a ZMP controller.

ZMP Preview control attempts to predict (and minimize) future CoM motions and ZMP errors through model predictive control (MPC). This main idea has been pursued by many authors (Kajita et al., 2003; Dimitrov et al., 2011; Tedrake et al., 2015) with different solution methods and formulations. Following the techniques in Kuindersma et al. (2015) and Tedrake et al. (2015), this problem can be formulated as one of continuous time linear optimal control

$$J^*(t_0, \mathbf{x}_0) = \min_{\mathbf{u}(t)} \int_{t_0}^{\infty} \|\mathbf{y}_T(t)\|_{\mathbf{R}}^2 + \|\mathbf{u}_T(t) - \mathbf{u}_T^d(t)\|_{\mathbf{Q}}^2 \, \mathrm{d}t \tag{4.5.16}$$

$$\text{s.t. } \dot{\mathbf{x}}_T(t) = \mathbf{A}\,\mathbf{x}_T(t) + \mathbf{B}\,\mathbf{u}_T(t), \tag{4.5.17}$$

$$\mathbf{y}_T(t) = \mathbf{C}\mathbf{x}_T(t) + \mathbf{D}\mathbf{u}_T(t), \tag{4.5.18}$$

$$\mathbf{x}_T(t_0) = \mathbf{x}_0, \tag{4.5.19}$$

where $J^*(t_0, \mathbf{x}_0)$ represents the optimal cost-to-go for an optimal ZMP preview controller. The matrices $\mathbf{R} = \mathbf{R}^T > 0$ and $\mathbf{Q} = \mathbf{Q}^T > 0$ are positive definite matrices that encode the relative importance of minimizing CoM accelerations and ZMP tracking errors through the weighted ℓ_2-norms $\|\mathbf{y}\|_{\mathbf{R}} = \sqrt{\mathbf{y}^T\mathbf{R}\mathbf{y}}$ and $\|\mathbf{u}\|_{\mathbf{Q}} = \sqrt{\mathbf{u}^T\mathbf{Q}\mathbf{u}}$, respectively.

We note that due to the invertibility of $\mathbf{D}$, $\mathbf{u}_T$, or $\mathbf{y}_T$ could either be viewed as the control for optimization. Regardless, this is a standard LQR problem (Tedrake et al., 2015) with an optimal cost-to-go of the form:

$$J^*(t, \mathbf{x}_T) = \mathbf{x}_T^T\mathbf{S}_1(t)\mathbf{x}_T + \mathbf{x}_T^T\mathbf{s}_2(t) + s_3(t). \tag{4.5.20}$$

Details to analytically form $\mathbf{S}_1(t)$, $\mathbf{s}_2(t)$, and $s_3(t)$ are provided in (Tedrake et al., 2015) for the interested reader and may be derived as an exercise. The optimal control law is given by the Hamilton–Jacobi–Bellman (HJB) equation (Bertsekas, 2005) for $\mathbf{u}_T$ or $\mathbf{y}_T$ as

$$\begin{aligned}
&\mathbf{u}_T^*(t, \mathbf{x}_T(t)) \\
&\quad = \underset{\mathbf{u}_T}{\operatorname{argmin}}\left(\|\mathbf{C}\mathbf{x}_T(t) + \mathbf{D}\mathbf{u}_T\|_{\mathbf{R}}^T + \|\mathbf{u}_T - \mathbf{u}_T^d(t)\|_{\mathbf{Q}}^2 + \frac{\mathrm{d}}{\mathrm{d}t}J^*(t, \mathbf{x}_T(t))\right), \text{ or}
\end{aligned} \tag{4.5.21}$$

$$\mathbf{y}_T^*(t, \mathbf{x}_T(t)) = \underset{\mathbf{y}_T}{\operatorname{argmin}}\, L(t, \mathbf{x}_T(t), \mathbf{y}_T) \tag{4.5.22}$$

$$\quad = \underset{\mathbf{y}_T}{\operatorname{argmin}}\left(\|\mathbf{y}_T(t)\|_{\mathbf{R}}^T + \|\mathbf{D}^{-1}(\mathbf{y}_T - \mathbf{C}\mathbf{x}_T(t)) - \mathbf{u}_T^d(t)\|_{\mathbf{Q}}^2 + \frac{\mathrm{d}}{\mathrm{d}t}J^*(t, \mathbf{x}_T(t))\right), \tag{4.5.23}$$

which balances instantaneous costs with long-term costs encoded in the optimal-cost-to-go (Kuindersma et al., 2015; Tedrake et al., 2015). The next subsection will explore how this optimal control solution can be lifted into a more complex robot model.

Capture point methods provide a different perspective to controlling the CoM. The capture point (CP) was originally introduced as a point on the ground where a robot would have to step to bring its CoM to a complete stop (Pratt et al., 2006, 2012; Koolen et al., 2012). This concept is illustrated in Fig. 4.5.4. For the linear inverted pendulum model, the CP $\boldsymbol{\xi} = [\xi_x, \xi_y, 0]^T$ is a composite variable in a sense that it incorporates both position and velocity information:

$$\xi_x = x + \frac{1}{\omega}\dot{x} \tag{4.5.24}$$

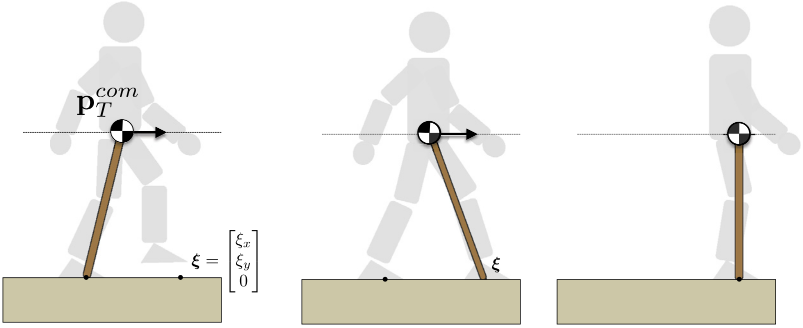

FIGURE 4.5.4 A capture point is a place where the foot may be placed such that the model may be brought to a complete stop. For the LIP model, the capture point is a linear combination of the CoM (x, y) position and the $(\dot{x}, \dot{y})$ velocity.

with similar definition in y (Koolen et al., 2012). Beyond seeing the capture point as a place to step, it should be noted that the capture point may be viewed as a bandwidth-tuned CoM look-ahead. Indeed, the term $1/\omega$ in (4.5.24) represents a time constant for the LIP dynamics. Performing a linear change of variables $(x, \dot{x}) \rightarrow (x, \xi_x)$ results in the system

$$\dot{x} = -\omega\,(x - \xi_x), \tag{4.5.25}$$

$$\dot{\xi}_x = \omega\,(\xi_x - p_x)\,. \tag{4.5.26}$$

In the case that the ZMP is placed at the capture point, (4.5.26) provides $\dot{\boldsymbol{\xi}} = 0$ while (4.5.25) implies that the (x, y) CoM exponentially converges to the capture point. Thus, (4.5.25) describes the system dynamics due to the *stable* pole of the LIP at $s = -\omega$, again providing an attractor for the CoM to the capture point. In contrast, (4.5.26) describes the dynamics from the *unstable* pole of the LIP at $s = +\omega$. These unstable capture point dynamics, however, represent a controllable subsystem, motivating the development of explicit capture point controllers.

As a result of the cascaded structure in the CoM/CP dynamics, control of the second-order LIP dynamics (4.5.5) may be instead pursued through control of the first-order capture point dynamics (4.5.26). Under proper tracking control of the capture point itself, stable attraction of the CoM to the capture point provides CoM tracking as a byproduct. As a result of this first-order structure, the instantaneous capture point and its 3D extension, also called the divergent component of motion (Takenaka et al., 2009; Englsberger et al., 2015), have provided an analytically clean framework to consider CoM control. Capture point-based methods are able to plan CoM trajectories directly for push recovery and walking over uneven terrains through closed-form analysis that is judiciously enabled by this reductive change of variables. Insightful geometric interpretations from the capture point and its implications for push recovery are elegantly covered in Koolen et al. (2012) with extensions to 3D in Englsberger et al. (2015).

4.5.3.2 Control for SLIP-Based Models

Within the domain of running, SLIP-based template models have provided many principles for adjusting leg behaviors to control dynamic balance. Due to the hybrid structure of the SLIP dynamics, control of the SLIP may be managed both through the selection of discrete control variables (i.e., touchdown angles) as well as continuous actions (i.e., changes in leg stiffness, nominal rest length, etc.). As in Example 4, continuous controls may often be parameterized by a discrete set of variables to simplify Poincaré analysis. Wu and Geyer (2013) developed deadbeat controllers for the 3D-SLIP using offline optimization to determine leg touchdown angles for ground height disturbance rejection. For online computation, the lack of an analytical expression for the Poincaré return map presents computational challenges. Carver (2003) and Wensing and Orin (2013b) developed locally deadbeat controllers using a linearized analysis of the Poincaré return map to reactively handle disturbances. Other authors have developed approximations to the return map to accelerate online control computations (Arslan et al., 2009; Geyer et al., 2005; Piovan and Byl, 2016). The following example highlights the ability to develop SLIP-based footstep controllers through linearized analysis.

Example 6 (Approximate Deadbeat Control of the 3D-SLIP)**.** In this example, we will develop a stabilizing controller for the actuated 3D-SLIP model running on level terrain. Following Example 4, let us assume that we have precomputed nominal controls $\mathbf{u}_T^d[n]$ and associated apex states $\mathbf{x}_T^d[n]$ that follow the discrete dynamics:

$$\mathbf{x}_T^d[n+1] = \mathbf{f}(n, \mathbf{x}_T^d[n], \mathbf{u}_T^d[n])\,. \tag{4.5.27}$$

In this example, we seek to find a control law $\mathbf{u}_T[n] = \boldsymbol{\pi}(n, \mathbf{x}_T[n])$ that provides local asymptotic tracking $\mathbf{x}_T[n] \to \mathbf{x}_T^d[n]$ to the nominal state trajectory as $n \to \infty$. As one of many possible approaches, constructing a deadbeat controller would remove all tracking error within a single step. This would require a policy such that

$$\mathbf{x}_T^d[n+1] = \mathbf{f}(n, \mathbf{x}_T[n], \boldsymbol{\pi}(n, \mathbf{x}_T[n]))\,. \tag{4.5.28}$$

Finding a policy satisfying this equation exactly requires detailed computation that may not be viable online. We demonstrate an approximate solution here.

Taking a Taylor expansion of (4.5.14) around the nominal trajectory provides

$$\tilde{\mathbf{x}}_T[n+1] = \mathbf{A}[n]\,\tilde{\mathbf{x}}_T[n] + \mathbf{B}[n]\,\tilde{\mathbf{u}}_T[n] + o\left(\left\|\left(\tilde{\mathbf{x}}_T[n], \tilde{\mathbf{u}}_T[n]\right)\right\|\right) \tag{4.5.29}$$

where $\tilde{\mathbf{x}}_T[n] = \mathbf{x}_T[n] - \mathbf{x}_T^d[n]$, $\tilde{\mathbf{u}}_T[n] = \mathbf{u}_T[n] - \mathbf{u}_T^d[n]$,

$$\mathbf{A}[n] = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}_T}\right|_{(n,\mathbf{x}_T^d[n],\mathbf{u}_T^d[n])}, \text{ and} \tag{4.5.30}$$

$$\mathbf{B}[n] = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}_T}\right|_{(n,\mathbf{x}_T^d[n],\mathbf{u}_T^d[n])}. \tag{4.5.31}$$

When the matrix $\mathbf{B}[n]$ has full row rank, the deadbeat condition (4.5.28) can be satisfied locally through the selection of any feedback law $\tilde{\mathbf{u}}_T[n] = \mathbf{K}[n]\,\tilde{\mathbf{x}}_T[n]$ satisfying

$$0 = \mathbf{A}[n] + \mathbf{B}[n]\,\mathbf{K}[n]\,. \tag{4.5.32}$$

The control policy in original coordinates

$$\boldsymbol{\pi}(\mathbf{x}_T[n], n) = \mathbf{u}_T^d[n] + \mathbf{K}[n]\,(\mathbf{x}_T[n] - \mathbf{x}_T^d[n]) \tag{4.5.33}$$

then admits local asymptotic tracking to the desired trajectory.

This simple approach can automatically capture many of the powerful heuristics that enabled dynamic gaits in Raibert's machines (Raibert, 1986). For instance, considering the actuated 3D-SLIP from Examples 1 and 2 with parameters $m = 72.5$ kg, $\ell_h = 0.97$ m, and $\mathbf{p}_T^{hip} = \mathbf{p}_T^{com} + [0,\ 12\text{ cm} \cdot (-1)^n,\ 0]^T$, the controls

$$\theta^d[n] = 0.4\text{ rad}, \qquad \phi^d[n] = 0\text{ rad}, \tag{4.5.34}$$

$$k_{s_1}^d[n] = 12.7\text{ kN/m}, \qquad k_{s_2}^d[n] = 12.7\text{ kN/m} \tag{4.5.35}$$

can be used to generate a left-right symmetric 2-step periodic running gait for a nominal speed of $\dot{x} = 3.5$ m/s (Wensing and Orin, 2013b). For such a gait, a feedback policy $\mathbf{K}[n]$ that satisfies (4.5.32) can also be developed to be 2-step periodic. For a left-foot step (i.e., $\mathbf{p}_T^{hip} = \mathbf{p}_T^{com} + [0,\ 12\text{ cm},\ 0]^T$ with coordinates given in Fig. 4.5.3), one such feedback gain is

$$\begin{bmatrix} \tilde{\theta} \\ \tilde{\phi} \\ \tilde{k}_{s_1} \\ \tilde{k}_{s_2} \end{bmatrix} = \underbrace{\begin{bmatrix} -0.51 & 0.13 & -0.01 \\ -1.95 & -0.08 & 0.90 \\ 36.9 & 13.2 & 0.86 \\ -36.9 & -13.2 & -0.86 \end{bmatrix}}_{\mathbf{K}[n]} \begin{bmatrix} \tilde{z} \\ \dot{\tilde{x}} \\ \dot{\tilde{y}} \end{bmatrix} \tag{4.5.36}$$

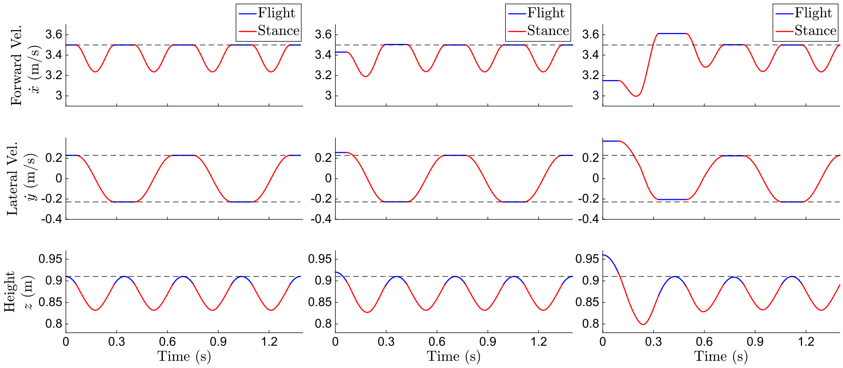

FIGURE 4.5.5 (Left) Nominal 2-step periodic 3D-SLIP gait for running at 3.5 m/s. (Center) Local deadbeat performance of the SLIP controller with small disturbances. (Right) Asymptotic tracking performance for larger disturbances with the local deadbeat SLIP controller.

where an additional constraint $\tilde{k}_{s_1} = -\tilde{k}_{s_2}$ has been employed to resolve the fact that four control variables exist to satisfy three deadbeat constraints. Looking at the bottom two rows shows the control response to a state having excess potential energy in z, or excess kinetic energy in $\dot{x}$ or $\dot{y}$. In all cases, the feedback law is one that employs a stiffer spring at touchdown, and softens at maximum compression in order to remove the excess energy. Other heuristics exist in each column. The first column shows that a running state that begins too high should be countered by placing the foot further under the CoM at touchdown. Similar leg placement heuristics are shown in the second and third columns. We note that these automatically tuned heuristics can provide guiding principles, not just for template control, but for humanoid control as we will pursue in the next subsection.

To demonstrate the performance of the control law (4.5.33), Fig. 4.5.5 shows the nominal state trajectory, the state trajectory subject to a small disturbance wherein local deadbeat behavior is approximately observed, and response to a larger perturbation wherein asymptotic tracking is recovered.

It should be noted that certain operating regimes of the 2D-SLIP model have been shown to possess so-called self-stable behavior. That is to say, for a fixed touchdown angle θ and leg stiffness k_s, open-loop stable gaits have been shown to exist (Seyfarth et al., 2002; Ghigliazza et al., 2003). While these results are an interesting finding, they do not extend to 3D (Seipel and Holmes, 2005) and represent a lower bound on the domain of attraction and robustness achievable by the addition of feedback.

More recently, the bipedal SLIP model has been proposed as an alternative walking template (Geyer et al., 2006) for the CoM. Recent work has developed controllers for this template based on analysis of its Poincaré return map

(Vejdani et al., 2015; Liu et al., 2015) similar to in the 3D-SLIP for running. This template includes rich nonlinear hybrid dynamics, due to two virtual spring legs making and breaking contact. As a result, control strategies for this model have often required offline computation with detailed knowledge bases used for real-time control.

4.5.3.3 Beyond Tracking Control for Pendular Models

Beyond these traditional CoM models, the computer graphics community has employed many other nontraditional template models for physics-based simulation of virtual characters. Mordatch et al. (2010) used a translational LIP with decoupled vertical SLIP to provide a relaxation of the SLIP model for locomotion planning with evolutionary search. da Silva et al. (2008) used iLQR with a three-link model as a template for the body and leg CoM motions. Ye and Liu (2010) applied differential dynamic programming to a CoM and angular momentum model that included an integrated centroidal angular momentum (Orin et al., 2013) as a nonphysical surrogate angular state.

Many of these cited works have focused template control on asymptotic tracking guarantees. Moving forward, template control warrants investigation to achieve other important control specifications such as robustness, viability, and yet others. Terrain robustness specifically has been a focus of much work within SLIP frameworks (Ernst et al., 2009; Wu and Geyer, 2013; Liu et al., 2016). Other work has advocated for viability (Wieber, 2008) as a more appropriate goal in legged systems. For a state to be viable intuitively means that it can be controlled to avoid any undesirable regions of the state space (such as those in which the robot has fallen to the ground). Indeed, in robotics when we say a system is “stable” it rarely is meant rigorously in a Lyapunov sense, but rather is more loosely meant in a sense of not falling down. Viability theory offers potential to bring rigor to this loose specification. Its rigorous technical definitions lead, in principle, to methods for identifying viable regions of the state space. Such verifications are beyond the reach of current computational methods for high-DoF systems. Yet, a notion of viability drives the definition of capturability (Koolen et al., 2012), and motivates the use of low-DoF template models to pursue viability more broadly (Sherikov et al., 2015).

4.5.4 ESTABLISHING A TEMPLATE/ANCHOR RELATIONSHIP

In the last step of the process, a high-DoF control system must be developed to establish the template in the anchor dynamics. Control of the template itself does not consider the issue of how the various actuators in the full system should be recruited. Thus, a high-DoF control problem exists to realize any template/anchor (T/A) relationship. Fig. 4.5.6 provides a rough diagram of how a solution to this realization fits into a full solution for real-time template-based control in a high-DoF anchor.

An initial part of this design stage entails the selection of which template/anchor notion to pursue. As alluded to in Subchapter 3.2, there are a wide range of specifications for what a proper T/A relationship may entail. At one extreme, the template states may be represented by a normally hyperbolic invariant manifold (NHIM) embedded in the state space of the anchor. To satisfy this notion of the T/A relationship, a diffeomorphic copy of the controlled template dynamics must be rendered on the NHIM in closed-loop. Towards realizing this condition, the template may be sought as the hybrid zero dynamics (HZD) (Westervelt et al., 2003; Poulakakis and Grizzle, 2009) of a judiciously crafted output regulation problem. HZD methods will be described further in Subchapter 4.7, and offer promise to bring the full scope of the templates and anchors hypothesis to bear in experimental machines.

Simpler methods may strive to replicate only select aspects of the template within the anchor dynamics. For instance, rather than the SLIP state encoding a target whole body state for the anchor (as in the NHIM notion of the T/A relationship), it may instead simply encode a target state for the anchor CoM. In this light, we will denote "anchor features" as any features of motion in the anchor whose control is informed by the template. Existing methods for controlling anchor features to match a template differ considerably in the degree of replication quality that they provide. In methods that only loosely achieve the desired template dynamics, this inaccuracy itself may represent a disturbance to any template-level controller. In methods that more precisely achieve the target template dynamics, a greater burden may exist on the template to operate in a manner that is replicable in the full system.

4.5.4.1 Realizing Template Dynamics Through Task-Space Control

Task-space or operational-space control provides a formal framework to pursue an exact realization of a template's continuous dynamics in a more complex anchor system. The state of the anchor system can be given as $[\mathbf{q}, \dot{\mathbf{q}}]^T \in \mathbb{R}^{n_A}$ with configuration $\mathbf{q}$. In legged robots, the configuration $\mathbf{q}$ generally possesses the structure $\mathbf{q} = [\mathbf{q}_b, \mathbf{q}_j]$ where $\mathbf{q}_b \in SE(3)$ is the configuration of a floating-base and $\mathbf{q}_j \in \mathbb{R}^{n_j}$ is the configuration of the internal joints. We denote $\mathbf{x}_A(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n_T}$ (same dimensions as $\mathbf{x}_T$) as the anchor feature state (i.e., the projection of the anchor state onto those features that are captured in the template). For instance, for CoM templates where $\mathbf{x}_T = [\mathbf{p}_T^{com}, \dot{\mathbf{p}}_T^{com}]$, the projection would extract the CoM position and velocity $\mathbf{x}_A := [\mathbf{p}_A^{com}, \dot{\mathbf{p}}_A^{com}]$ of the anchor

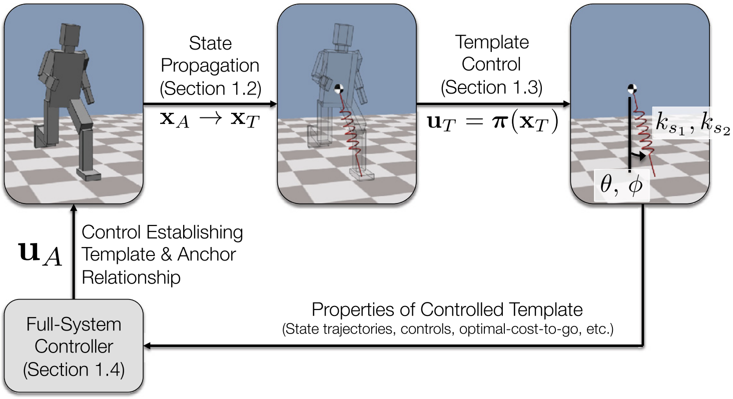

FIGURE 4.5.6 Real-time template-based control diagram. The relevant features of the anchor system $\mathbf{x}_A = \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}})$ are propagated to the template $\mathbf{x}_T$. A control system applied to the template $\mathbf{u}_T = \boldsymbol{\pi}(\mathbf{x}_T)$ gives rise to target properties of the closed-loop template dynamics. These properties are lifted into the full system by a high-dimensional controller that coordinates many actuators $\mathbf{u}_A$ to realize a template/anchor relationship.

system. For physical template models more broadly, this relationship is often clear, however, for more abstract templates, this assignment itself may present another degree of design freedom.

As shown in Fig. 4.5.6, this projected state $\mathbf{x}_A$ is propagated to the template $\mathbf{x}_A \rightarrow \mathbf{x}_T$ to begin the application of template-based control. As one of many possibilities, let us assume that such propagation happens discretely, with trajectories of the controlled template $\mathbf{x}_T(t)$ provided between updates. For instance, in the SLIP model, such propagation might only occur at the Poincaré section, with $\mathbf{x}_T(t)$ representing a controlled SLIP trajectory for the next step. Controlling the full system trajectory $\mathbf{x}_A(t)$ to asymptotically anchor the template $\mathbf{x}_A(t) \rightarrow \mathbf{x}_T(t)$ is then a well studied problem within the context of the operational-space and whole-body control literature.

Replicating the Dynamics of CoM Templates

A common whole-body control solution applied to the replication of template dynamics is to minimize the replication errors at a dynamic level through real-time optimization in closed-loop. To build towards this solution, we reintroduce the standard dynamic equations of motion

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{S}_a^T \boldsymbol{\tau} + \mathbf{J}_s(\mathbf{q})^T \mathbf{F}_s \tag{4.5.37}$$

where $\mathbf{H}$, $\mathbf{C}\dot{\mathbf{q}}$, and $\mathbf{G}$ are the familiar mass matrix, velocity product terms, and gravitational terms, respectively. Here $\mathbf{F}_s$ collects ground reaction forces (GRFs) for appendages in support and $\mathbf{J}_s$ is a combined support Jacobian. The matrix $\mathbf{S}_a = [\,\mathbf{0}_{n_j\times 6}\;\; \mathbf{1}_{n_j\times n_j}\,]$ is a selection matrix for the actuated joints.

For CoM templates with anchor features $\mathbf{x}_A = [\mathbf{p}_A^{com}, \dot{\mathbf{p}}_A^{com}]$, an optimization problem can first be formulated to partial-feedback linearize the CoM dynamics of the anchor under whole-body constraints. Given a desired acceleration for the CoM $\ddot{\mathbf{p}}_A^{com,d}$, the goal of the whole-body controller is to select joint torques $\mathbf{u}_A := \boldsymbol{\tau}$ that most closely realize this commanded acceleration. This can be formulated as an optimization problem

$$\min_{\ddot{\mathbf{q}},\boldsymbol{\tau},\mathbf{F}_s} \quad \frac{1}{2}\left\| \mathbf{J}_{com}\ddot{\mathbf{q}} + \dot{\mathbf{J}}_{com}\dot{\mathbf{q}} - \ddot{\mathbf{p}}_A^{com,d} \right\|^2 \tag{4.5.38}$$

$$\text{subject to } \mathbf{H}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{G} = \mathbf{S}_a^T\boldsymbol{\tau} + \mathbf{J}_s^T\mathbf{F}_s, \tag{4.5.39}$$

$$\mathbf{J}_s\ddot{\mathbf{q}} + \dot{\mathbf{J}}_s\dot{\mathbf{q}} = \mathbf{0}, \tag{4.5.40}$$

$$\mathbf{F}_s \in \boldsymbol{\mathcal{C}}, \tag{4.5.41}$$

where $\mathbf{J}_{com} \in \mathbb{R}^{3\times(n_j+6)}$ is the CoM Jacobian, $\mathbf{F}_s \in \mathbb{R}^{6n_f}$ are ground reaction forces for n_f feet in planar contact, and $\boldsymbol{\mathcal{C}}$ is the convex cone of forces that can be created through the available contacts (Wensing and Orin, 2013a). The constraint (4.5.40) enforces that any supporting contacts must not move. While ground forces $\mathbf{F}_s$ may often be viewed as the Lagrange multipliers associated with this constraint, their solution is not unique when multiple feet are in planar contact. As a result, the optimization enforces that at least one set of these Lagrange multipliers must satisfy frictional constraints.

If the optimization problem can be solved to an optimal objective function value of 0, then the current contacts provide the necessary control authority to exactly realize the commanded dynamics. When this is the case, solution of the optimization problem in closed-loop will provide a feedback linearization from the commanded dynamics $\ddot{\mathbf{p}}_A^{com,d}$ to the anchor feature $\mathbf{p}_A^{com}$. As a result, a common approach to achieve template tracking is to select the commanded feature dynamics according to the law

$$\ddot{\mathbf{p}}_A^{com,d} = \ddot{\mathbf{p}}_T^{com} + k_D(\dot{\mathbf{p}}_T^{com} - \dot{\mathbf{p}}_A^{com}) + k_P(\mathbf{p}_T^{com} - \mathbf{p}_A^{com}) \tag{4.5.42}$$

where k_P and k_D are positive definite gain matrices. This selection provides an asymptotically stable second-order dynamic for the anchoring error $\mathbf{e}(t) = \mathbf{p}_T^{com}(t) - \mathbf{p}_A^{com}(t)$. If the template includes only kinematic features and their associated derivatives, then the above development is general through the use of an appropriate task Jacobian. If the template includes purely velocity-dependent features, such as angular momentum, then minimal modifications to the above development can be employed (Wensing and Orin, 2013a).

Exploiting Redundancy

It should be noted that there is significant redundancy remaining after only fulfilling the commanded anchor feature dynamics. Often the use of this flexibility is needed to track aspects of the full system that are absent in the template. For instance, in running with the SLIP template, the lack of a swing leg in the template requires coordinated swing leg torques in the full model to move the swing leg into place for the following step.

These torques may be found algorithmically through solving a subsequent optimization problem that respects the optimal anchor feature dynamics from (4.5.38). Given desired accelerations $\ddot{\mathbf{p}}_1^d \in \mathbb{R}^{n_1}$ for another feature of motion, such as a swing foot position, modified torques can be derived from the solution to

$$\min_{\ddot{\mathbf{q}},\boldsymbol{\tau},\mathbf{F}_s} \quad \frac{1}{2}\left\|\mathbf{J}_1\ddot{\mathbf{q}} + \dot{\mathbf{J}}_1\dot{\mathbf{q}} - \ddot{\mathbf{p}}_1^d\right\|^2 \tag{4.5.43}$$

$$\text{subject to } \mathbf{H}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{G} = \mathbf{S}_a^T\boldsymbol{\tau} + \mathbf{J}_s^T\mathbf{F}_s, \tag{4.5.44}$$

$$\mathbf{J}_s\ddot{\mathbf{q}} + \dot{\mathbf{J}}_s\dot{\mathbf{q}} = \mathbf{0}, \tag{4.5.45}$$

$$\mathbf{J}_{com}\ddot{\mathbf{q}} + \dot{\mathbf{J}}_{com}\dot{\mathbf{q}} = \ddot{\mathbf{p}}_A^*, \tag{4.5.46}$$

$$\mathbf{F}_s \in \mathcal{C}, \tag{4.5.47}$$

where $\ddot{\mathbf{p}}_A^*$ is the optimal anchor feature acceleration resulting from first solving (4.5.38). By solving (4.5.38) and then (4.5.43) in a cascaded fashion, a strict prioritization is being given to tracking the anchor features above all else. When a strictly ordered hierarchy exists between yet further tasks, this approach may be extended through additional cascaded solves or through dedicated hierarchical solvers (Escande et al., 2014).

As an alternative, soft priorities can be implemented through solving a single optimization problem with an objective that is a weighted combination of the task command errors:

$$\min_{\ddot{\mathbf{q}},\boldsymbol{\tau},\mathbf{F}_s} \quad w_A\|\mathbf{J}_{com}\,\ddot{\mathbf{q}} + \dot{\mathbf{J}}_{com}\,\dot{\mathbf{q}} - \ddot{\mathbf{p}}_A^{com,d}\|^2 + \sum_i w_i\|\mathbf{J}_i\,\ddot{\mathbf{q}} + \dot{\mathbf{J}}_i\,\dot{\mathbf{q}} - \ddot{\mathbf{p}}_i^d\|^2 \tag{4.5.48}$$

$$\text{subject to } \mathbf{H}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{G} = \mathbf{S}_a^T\boldsymbol{\tau} + \mathbf{J}_s^T\mathbf{F}_s, \tag{4.5.49}$$

$$\mathbf{J}_s\ddot{\mathbf{q}} + \dot{\mathbf{J}}_s\dot{\mathbf{q}} = \mathbf{0}, \tag{4.5.50}$$

$$\mathbf{F}_s \in \mathcal{C}. \tag{4.5.51}$$

In comparison to using a strict prioritization, soft prioritization requires fewer invocations of an optimization solver, but can suffer from numeric conditioning issues if a large disparity in weights is desired. Overall, these optimization-based whole-body control methods perform well in practice, and have been a

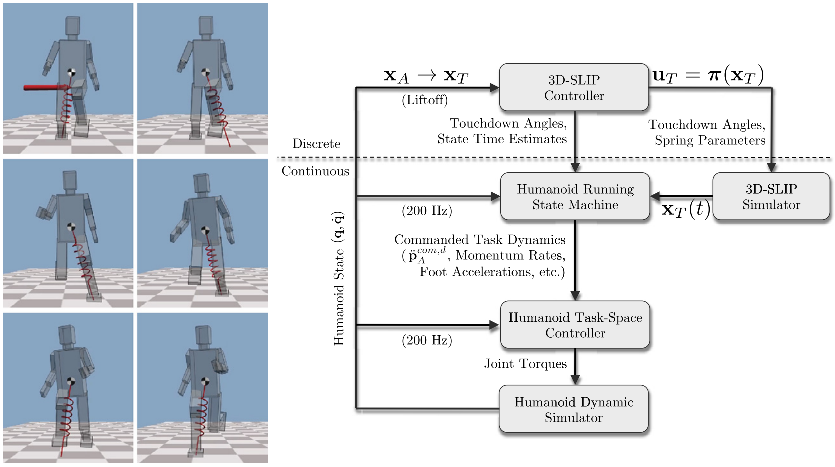
$\mathbf{x}_A \to \mathbf{x}_T$
(Liftoff)
3D-SLIP Controller
$\mathbf{u}_T = \boldsymbol{\pi}(\mathbf{x}_T)$
Touchdown Angles, State Time Estimates
Touchdown Angles, Spring Parameters
Discrete
Continuous
Humanoid Running State Machine
$\mathbf{x}_T(t)$
3D-SLIP Simulator
(200 Hz)
Humanoid State $(\mathbf{q}, \dot{\mathbf{q}})$
Commanded Task Dynamics ($\ddot{\mathbf{p}}_A^{com,d}$, Momentum Rates, Foot Accelerations, etc.)
Humanoid Task-Space Controller
(200 Hz)
Joint Torques
Humanoid Dynamic Simulator

FIGURE 4.5.7 Template-based whole-body response to a push disturbance and associated whole-body control system. Following a push (red arrow in the upper-left subfigure), the disturbed state of the humanoid is projected and propagated to the SLIP template $\mathbf{x}_A \to \mathbf{x}_T$. SLIP control $\mathbf{u}_T = \boldsymbol{\pi}(\mathbf{x}_T)$ suggests touchdown angles for the upcoming steps and a dynamically feasible CoM trajectory $\mathbf{x}_T(t)$ for a template-based recovery in the following stance. (For interpretation of the colors in this figure, the reader is referred to the web version of this chapter.)

workhorse for modern humanoids in recent years. However, they are fundamentally single-step model predictive control schemes where the actions that are instantaneously greedy are designed to play out favorably in the long term. The template controllers from Section 4.5.3 readily admit formal guarantees on their performance. However, when coupled to optimization-based whole body controllers such as (4.5.38), currently little can be proven or guaranteed about the long-term behavior of these systems.

Example 7 (Closed-Loop Control of High-Speed Humanoid Running)**.** Fig. 4.5.7 shows the results of a full template-based control system applied to humanoid push recovery while running at 3.5 m/s with the SLIP controller developed in Example 6. At each liftoff, anchor features are propagated to the SLIP template. Controlled SLIP trajectories $\mathbf{x}_T(t)$ then provide physics-based recovery motions. At each step, (4.5.33) provides touchdown angles and matched spring stiffnesses k_{s_1} and k_{s_2} to provide approximate deadbeat tracking back to the nominal gait within one step. Due to system features, such as ground/foot impacts that are not captured in the SLIP, this full template-based controller does not experience the strong deadbeat behavior shown in Example 6. Fig. 4.5.8 shows the CoM response in the simulation experiment demonstrating that the template recovery motions are realized in the anchor.

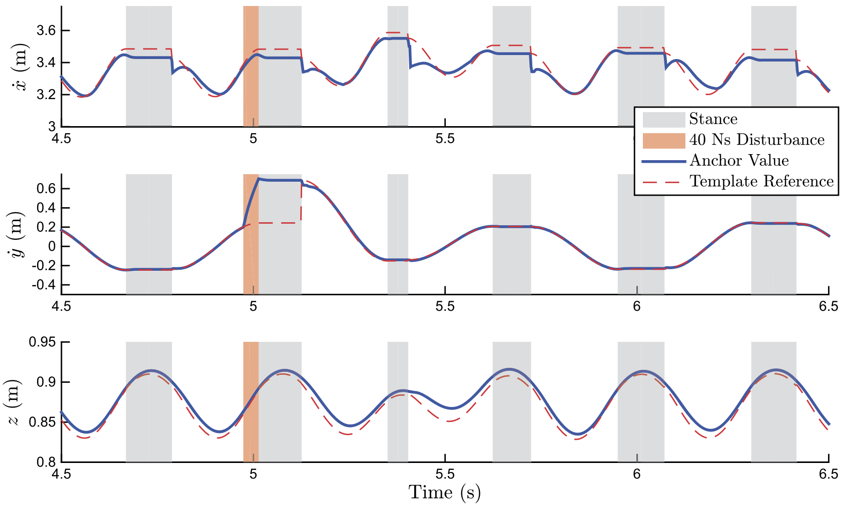

FIGURE 4.5.8 Response of a full template-based control system for a humanoid following a lateral push disturbance of 40 Ns. At each step a CoM reference trajectory is generated based on the results of the local deadbeat template controller from Example 6. At each transition from flight to stance, an impact impulse in the full system, which is due to leg mass not captured in the SLIP, provides a persistent disturbance to the full template-based control system. The system nominally recovers to its steady-state gait within 4 steps.

Since the SLIP assumes massless legs that can be instantly repositioned in flight, a state machine is used in the humanoid to coordinate leg motions. This state machine does so through providing foot acceleration commands to an optimization-based controller as in (4.5.43). Foot acceleration commands similarly take a PD form as in (4.5.42). However, rather than the template providing reference trajectories, cubic spline references are generated online to achieve the desired virtual leg configuration at touchdown. To resolve remaining redundancy, a centroidal angular momentum (Orin et al., 2013) rate of change command and pose acceleration command (Wensing and Orin, 2013b) are also provided to (4.5.43). The optimization is solved at rate of 200 Hz, and resulting joint torques $\boldsymbol{\tau}$ are applied in dynamic simulation. Further details can be found in Wensing and Orin (2013b).

4.5.4.2 Lifting Other Properties of Template Control

The previous example included feedback to the template on a discrete basis. Other methods continuously resolve the behavior of the full system with an optimal reaction from the template by lifting properties of an optimal template controller. For instance, optimal controllers for template models as in Example 5 can be used to guide CoM control in a humanoid. Following the construction of the optimal-cost-to-go from Example 5, the whole-body QP (4.5.48) from the previous subsection could be modified as (Kuindersma et al., 2015)

$$\min_{\ddot{\mathbf{q}},\boldsymbol{\tau},\mathbf{F}_s} \quad w_A L(t, \mathbf{x}_A, \ddot{\mathbf{p}}_A^{com}) + \sum_i w_i \|\mathbf{J}_i\,\ddot{\mathbf{q}} + \dot{\mathbf{J}}_i\,\dot{\mathbf{q}} - \ddot{\mathbf{x}}_i^c\|^2 \tag{4.5.52}$$

$$\text{subject to } \mathbf{H}\,\ddot{\mathbf{q}} + \mathbf{C}\,\dot{\mathbf{q}} + \mathbf{G} = \mathbf{S}_a^T\boldsymbol{\tau} + \mathbf{J}_s^T\mathbf{F}_s, \tag{4.5.53}$$

$$\mathbf{J}_s\ddot{\mathbf{q}} + \dot{\mathbf{J}}_s\dot{\mathbf{q}} = \mathbf{0}, \tag{4.5.54}$$

$$\mathbf{F}_s \in \mathcal{C}, \tag{4.5.55}$$

where again $L(t, \mathbf{x}_A, \ddot{\mathbf{p}}_A^{com})$ from (4.5.22) (Kuindersma et al., 2015) balances instantaneous template control costs with long-term costs encoded in its optimal-cost-to-go. The benefit of this formulation over (4.5.48) comes from the fact that when the optimal template dynamics are not instantaneously feasible, deviations from the optimal template dynamics are not created equal. Some deviations, while equal in magnitude, may be more costly than others in the long term. This subtlety is addressed by the long-term costs encoded in the optimal cost-to-go, while the error norm penalty in (4.5.48) manages no such long-term trade-off. A summary of the whole-body control results for the MIT DRC team, which used this approach in hardware with ATLAS, can be found in Kuindersma et al. (2015, 2014). In particular, the formulation of minimizing $L(t, \mathbf{x}_A, \ddot{\mathbf{p}}_A^{com})$ was cited to provide practical robustness over other methods that simply enforce decreasing the template-based optimal-cost-to-go over time.

Overall, these optimization-based control methods guarantee template tracking error will asymptotically approach zero when sufficient control authority exists. If there are unmodeled aspects of the anchor system that affect $\mathbf{x}_A$ but are not captured in the template, these unmodeled aspects represent a persistent disturbance to the template-based control system. Otherwise, asymptotic tracking across these examples is accomplished in (4.5.48) and (4.5.52), through carefully selecting joint torques that compensate for nonlinear joint-space dynamics and replace them with those commanded for the locomotion features. Tracking performance, as a result, is predicated on the correctness of the dynamic model itself and thus may exhibit sensitivity to modeling errors. Other less model-intensive methods are able to overcome these drawbacks but provide a looser match between template and anchor features.

4.5.4.3 Anchoring the Template Through Less Model-Intensive Methods

Virtual model control is a less model-intensive method to track locomotion features over time. Virtual model control (VMC) uses virtual components to formulate fictitious forces that govern interaction between real and virtual systems (Pratt et al., 2001). These fictitious forces are then realized through static joint torque mappings, often using the Jacobian transpose. Virtual systems may represent springs and dampers, or other general mechanisms that are designed to produce virtual interactions and regulate the state of the real system. Thus, in a sense, virtual model control is a descendant of simple impedance control, yet more general interaction dynamics may be authored through design of virtual models. VMC has been applied widely in quadruped robots to stabilize dynamic locomotion plans (Park and Kim, 2015; Semini et al., 2015; Coros et al., 2011; Winkler et al., 2014) and in dynamic balance for humanoids (Stephens and Atkeson, 2010). When physical template models are used in a template controller, the template itself can often be viewed as a virtual model. When this is the case, VMC provides a direct method to approximately realize the template dynamics. By only relying on kinematic data from the real system to resolve virtual template forces, VMC is essentially model-free at a dynamic level. Thus, without compensation for dynamic effects, VMC relies on heuristic gain tuning to find controller parameters that provide suitable performance at each operating point. Due to the changes in neglected dynamic forces across operating regimes, however, performance claims for these controllers can often only be verified empirically.

4.5.4.4 Template-Inspired Mechanical Design

Throughout this section, rigid-body dynamics (4.5.37) have been assumed with torque sources modeled at the joints. Yet, many modern machines do include other forms of impedance (compliance and damping) at the joints and more generally across the body structures. The natural dynamics of these structures may point toward the selection of a particular template that can significantly reduce the burden of imparting a template/anchor relationship through closed-loop control. For instance, the bipedal-SLIP model was used to develop high-level template-based controllers for ATRIAS (Rezazadeh et al., 2015). Lower-level control mechanisms did not explicitly attempt to impart bipedal-SLIP dynamics to the machine. However, since ATRIAS was designed to exhibit dynamics that embody the bipedal-SLIP (Ramezani et al., 2014), the robot has been shown to exhibit qualitative similarity to its template in terms of its ground reaction forces. A similar success story may be found in Raibert's early hoppers, whose control systems where inspired by SLIP-based laws and implemented on physical SLIP-type robots with air springs (Raibert, 1986). Further intersection of template-based control and template-inspired design presents interesting future prospects to simplify closed-loop control of legged machines.

4.5.5 CONCLUSIONS

This section has described the design process for using template models for control. We have discussed the three main sub-problems that must be integrated to realize a template-based control system. First, a template pertinent to the locomotion task must be identified. There is no right or wrong template for a given task and those from biology need only serve as inspiration. Next, control methodologies for the template itself must be identified and developed. In some cases, the template may not provide sufficient control flexibility to achieve desired controller specifications, requiring modifications to the template under consideration. Following the solution of a template control problem, controlled template dynamics must be retargeted to the full system, addressing the coordination of many actuators to achieve low-dimensional specifications. Many methods exist to realize these target template dynamics, which vary in the degree of replication quality that they impart. The application of template models for control does require the intuition of a human designer. However, its focus on the most important characteristics of locomotion has enabled reactive control architectures in many experimental robots, and may yet play a key role in unlocking the full potentials of legged machines.

REFERENCES

Arslan, O., Saranli, U., Morgul, O., 2009. An approximate stance map of the spring mass hopper with gravity correction for nonsymmetric locomotions. In: IEEE International Conference on Robotics and Automation, ICRA'09, pp. 2388–2393.

Bellman, R., 1957. Dynamic Programming. Princeton University Press, Princeton, NJ.

Bertsekas, D.P., 2005. Dynamic Programming and Optimal Control. Athena Scientific.

Blickhan, R., 1989. The spring–mass model for running and hopping. J. Biomech. 22 (11–12), 1217–1227.

Carver, S., 2003. Control of a Spring Mass Hopper. PhD thesis. Cornell University, Ithaca, NY.

Coros, S., Karpathy, A., Jones, B., Reveret, L., van de Panne, M., 2011. Locomotion skills for simulated quadrupeds. In: ACM SIGGRAPH 2011 Papers, SIGGRAPH '11. ACM, New York, NY, USA, pp. 59:1–59:12.

da Silva, M., Abe, Y., Popović, J., 2008. Interactive simulation of stylized human locomotion. In: ACM SIGGRAPH 2008 Papers. Los Angeles, California, pp. 82:1–10.

Diehl, M., Bock, H., Diedam, H., Wieber, P.-B., 2006. Fast direct multiple shooting algorithms for optimal robot control. In: Diehl, M., Mombaur, K. (Eds.), Fast Motions in Biomechanics and Robotics. In: Lecture Notes in Control and Information Sciences, vol. 340. Springer, Berlin, Heidelberg, pp. 65–93.

Dimitrov, D., Sherikov, A., Wieber, P.-B., 2011. A sparse model predictive control formulation for walking motion generation. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2292–2299.

Englsberger, J., Ott, C., Albu-Schaffer, A., 2015. Three-dimensional bipedal walking control based on divergent component of motion. IEEE Trans. Robot. 31, 355–368.

Ernst, M., Geyer, H., Blickhan, R., 2009. Spring-legged locomotion on uneven ground: a control approach to keep the running speed constant. In: 12th Int. Conf. on Climbing and Walking Robots (CLAWAR), pp. 639–644.

Escande, A., Mansard, N., Wieber, P.-B., 2014. Hierarchical quadratic programming: fast online humanoid-robot motion generation. Int. J. Robot. Res. 33 (7), 1006–1028.

Feng, S., Xinjilefu, X., Huang, W., Atkeson, C.G., 2013. 3D walking based on online optimization. In: Proc. of the IEEE-RAS International Conference on Humanoid Robots.

Full, R.J., Koditschek, D.E., 1999. Templates and anchors: neuromechanical hypotheses of legged locomotion on land. J. Exp. Biol. 202 (23), 3325–3332.

Garofalo, G., Ott, C., Albu-Schaffer, A., 2012. Walking control of fully actuated robots based on the bipedal SLIP model. In: IEEE Int. Conf. on Robotics and Automation, pp. 1456–1463.

Geyer, H., Seyfarth, A., Blickhan, R., 2005. Spring–mass running: simple approximate solution and application to gait stability. J. Theor. Biol. 232 (3), 315–328.

Geyer, H., Seyfarth, A., Blickhan, R., 2006. Compliant leg behaviour explains basic dynamics of walking and running. Proc. R. Soc. B, Biol. Sci. 273 (1603), 2861–2867.

Ghigliazza, R.M., Altendorfer, R., Holmes, P., Koditschek, D.E., 2003. Passively stable conservative locomotion. SIAM J. Appl. Dyn. Syst.

Herdt, A., Diedam, H., Wieber, P.-B., Dimitrov, D., Mombaur, K., Diehl, M., 2010. Online walking motion generation with automatic foot step placement. Adv. Robot. 24 (5–6), 719–737.

Kajita, S., Kanehiro, F., Kaneko, K., Yokoi, K., Hirukawa, H., 2001. The 3D linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, vol. 1, pp. 239–246.

Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., Hirukawa, H., 2003. 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, vol. 2, pp. 1620–1626.

Koolen, T., de Boer, T., Rebula, J., Goswami, A., Pratt, J., 2012. Capturability-based analysis and control of legged locomotion, Part 1: theory and application to three simple gait models. Int. J. Robot. Res. 31 (9), 1094–1113.

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

Kuindersma, S., Deits, R., Fallon, M., Valenzuela, A., Dai, H., Permenter, F., Koolen, T., Marion, P., Tedrake, R., 2015. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. In: Autonomous Robots, pp. 1–27.

Liu, Y., Wensing, P.M., Orin, D.E., Zheng, Y.F., 2015. Dynamic walking in a humanoid robot based on a 3D actuated dual-slip model. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 5710–5717.

Liu, Y., Wensing, P.M., Schmiedeler, J.P., Orin, D.E., 2016. Terrain-blind humanoid walking based on a 3-d actuated dual-slip model. IEEE Robot. Autom. Lett. 1, 1073–1080.

Mordatch, I., de Lasa, M., Hertzmann, A., 2010. Robust physics-based locomotion using low-dimensional planning. In: ACM SIGGRAPH 2010 papers. Los Angeles, California, pp. 71:1–8.

Orin, D.E., Goswami, A., Lee, S.-H., 2013. Centroidal dynamics of a humanoid robot. Auton. Robots 35 (2), 161–176.

Park, H.-W., Kim, S., 2015. Quadrupedal galloping control for a wide range of speed via vertical impulse scaling. Bioinspir. Biomim. 10 (2), 025003.

Park, H.-W., Wensing, P., Kim, S., 2015. Online planning for autonomous running jumps over obstacles in high-speed quadrupeds. In: Proceedings of Robotics: Science and Systems. Rome, Italy.

Piovan, G., Byl, K., 2016. Approximation and control of the slip model dynamics via partial feedback linearization and two-element leg actuation strategy. IEEE Trans. Robot. 32, 399–412.

Poulakakis, I., Grizzle, J., 2009. The spring loaded inverted pendulum as the hybrid zero dynamics of an asymmetric hopper. IEEE Trans. Autom. Control 54, 1779–1793.

Pratt, J., Chew, C.-M., Torres, A., Dilworth, P., Pratt, G., 2001. Virtual model control: an intuitive approach for bipedal locomotion. Int. J. Robot. Res. 20 (2), 129–143.

Pratt, J., Koolen, T., de Boer, T., Rebula, J., Cotton, S., Carff, J., Johnson, M., Neuhaus, P., 2012. Capturability-based analysis and control of legged locomotion, part 2: application to m2v2, a lower-body humanoid. Int. J. Robot. Res. 31 (10), 1117–1133.

Pratt, J., Carff, J., Drakunov, S., Goswami, A., 2006. Capture point: a step toward humanoid push recovery. In: IEEE-RAS Int. Conf. on Humanoid Robots. Genova, Italy, pp. 200–207.

Raibert, M.H., 1986. Legged Robots that Balance. MIT Press, Cambridge, MA, USA.

Ramezani, A., Hurst, J.W., Hamed, K.A., Grizzle, J., 2014. Performance analysis and feedback control of ATRIAS, a three-dimensional bipedal robot. J. Dyn. Syst. Meas. Control 136 (2), 021012.

Rezazadeh, S., Hubicki, C., Jones, M., Peekema, A., Van Why, J., Abate, A., Hurst, J., 2015. Spring–mass walking with ATRIAS in 3D: robust gait control spanning zero to 4.3 kph on a heavily underactuated bipedal robot 10. V001T04A003.

Seipel, J.E., Holmes, P., 2005. Running in three dimensions: analysis of a point-mass sprung-leg model. Int. J. Robot. Res. 24 (8), 657–674.

Semini, C., Barasuol, V., Boaventura, T., Frigerio, M., Focchi, M., Caldwell, D.G., Buchli, J., 2015. Towards versatile legged robots through active impedance control. Int. J. Robot. Res. 34 (7), 1003–1020.

Seyfarth, A., Geyer, H., Günther, M., Blickhan, R., 2002. A movement criterion for running. J. Biomech. 35 (5), 649–655.

Sherikov, A., Dimitrov, D., Wieber, P.B., 2015. Balancing a humanoid robot with a prioritized contact force distribution. In: 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), pp. 223–228.

Stephens, B.J., Atkeson, C.G., 2010. Dynamic balance force control for compliant humanoid robots. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems. Taipei, Taiwan, pp. 1248–1255.

Takenaka, T., Matsumoto, T., Yoshiike, T., 2009. Real time motion generation and control for biped robot-1st report: walking gait pattern generation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1084–1091.

Tedrake, R., Manchester, I.R., Tobenkin, M., Roberts, J.W., 2010. LQR-trees: Feedback motion planning via sums-of-squares verification. Int. J. Robot. Res. 29 (8), 1038–1052.

Tedrake, R., Kuindersma, S., Deits, R., Miura, K., 2015. A closed-form solution for real-time ZMP gait generation and feedback stabilization. In: 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), pp. 936–940.

Vejdani, H.R., Wu, A., Geyer, H., Hurst, J.W., 2015. Touch-down angle control for spring-mass walking. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 5101–5106.

Wensing, P.M., Orin, D.E., 2013a. Generation of dynamic humanoid behaviors through task-space control with conic optimization. In: IEEE Int. Conf. on Rob. and Automation. Karlsruhe, Germany, pp. 3103–3109.

Wensing, P.M., Orin, D.E., 2013b. High-speed humanoid running through control with a 3D-SLIP model. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Rob. and Sys. Tokyo, Japan, pp. 5134–5140.

Westervelt, E., Grizzle, J., Koditschek, D., 2003. Hybrid zero dynamics of planar biped walkers. IEEE Trans. Autom. Control 48, 42–56.

Wieber, P.-B., 2008. Viability and predictive control for safe locomotion. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2008, pp. 1103–1108.

Winkler, A., Havoutis, I., Bazeille, S., Ortiz, J., Focchi, M., Dillmann, R., Caldwell, D., Semini, C., 2014. Path planning with force-based foothold adaptation and virtual model control for torque controlled quadruped robots. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 6476–6482.

Wu, A., Geyer, H., 2013. The 3-D spring–mass model reveals a time-based deadbeat control for highly robust running and steering in uncertain environments. IEEE Trans. Robot. 29 (5), 1114–1124.

Ye, Y., Liu, C.K., 2010. Optimal feedback control for character animation using an abstract model. In: ACM SIGGRAPH 2010. New York, NY, USA, pp. 74:1–9.