Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on MABEL

Koushil Sreenath¹, Hae-Won Park², Ioannis Poulakakis³ and JW Grizzle¹

Abstract

A mathematical formalism for designing running gaits in bipedal robots with compliance is introduced and subsequently validated experimentally on MABEL, a planar biped that contains springs in its drivetrain. The methods of virtual constraints and hybrid zero dynamics are used to design a time-invariant feedback controller that respects the natural compliance of the open-loop system. In addition, it also enables active force control within the compliant hybrid zero dynamics allowing within-stride adjustments of the effective stance leg stiffness. The proposed control strategy was implemented on MABEL and resulted in a kneed-biped running record of 3.06 m/s (10.9 kph or 6.8 mph).

Keywords

bipedal robots, running, hybrid systems, zero dynamics, compliance, force control.

1. Introduction

High-performance robot running requires the tight integration of the robot’s mechanical and control systems. Successful running machines involve compliant elements (such as springs) which, combined with the hybrid underactuated nature of their dynamics and the small time intervals available for control, present a challenge to state-of-the-art feedback design approaches. In this article, we provide a method that combines the analytical tractability afforded by the hybrid zero dynamics (HZD) framework, with physically intuitive compliance control, to induce reliable, fast running gaits on the bipedal robot MABEL, obtaining speeds up to 3.06 m/s in physical laboratory experiments; see Figure 1.

Empirical controllers assisted from intuition gained through the analysis of simplified spring–mass models have been successful in stabilizing running on legged machines with particular morphology. Raibert and his collaborators in the 1980s introduced a set of simple, intuitive principles to make various one-foot gaits possible on monopodal, bipedal, and, through the concept of virtual legs, on quadrupedal robots (Raibert, 1986). The proposed controllers regulate forward velocity by suitably positioning the legs during flight, and regulate hopping height and torso pitch by making use of motor torques during stance. These controllers have achieved record speeds of up to 5.9 m/s on a monopodal hopper (Koechling, 1989).

The success of Raibert’s control procedures prompted a series of robots (Sayyad et al., 2007), and mathematical models (Holmes et al., 2006), to investigate a variety of design and control aspects of robot running, including self-stability (Ghigliazza et al., 2003), energy minimization (Ahmadi and Buehler, 1997, 2006), active force control (Koepl et al., 2010), and energy removal strategies (Andrews et al., 2011). The majority of these systems are monopodal and feature light, prismatic, springy legs that are typically connected to the robot’s torso so that the hip joint coincides with the torso’s center of mass (COM). It is not clear, however, how control methods developed in the context of such systems can be transferred to robots whose morphology departs significantly from these assumptions. In particular, bipedal robots (such as MABEL, Figure 1) whose legs comprise revolute knee joints and have significant weight, and are coupled nontrivially to the torso dynamics, represent a challenge to control approaches derived on the basis of Raibert-style hoppers.

Contrary to walking gaits (for which a variety of controllers with analytically tractable properties are available; see Ames et al., 2006; Chevallereau et al., 2003; Gregg and Spong, 2010; Spong, 1999, for instance) only a few control methods are available for running bipeds. In many
cases, running was implemented on robots that were not specifically designed for such motions. Examples include humanoids like Sony’s QRIO (Nagasaka et al., 2004), Honda’s ASIMO (Hirose and Ogawa, 2007), the HRP family (Kajita et al., 2005, 2007), and HUBO (Cho et al., 2009). Recently, Toyota’s humanoid achieved running at speeds up to 1.94 m/s (Tajima et al., 2009). In all these cases, the underlying controllers are based on the Zero Moment Point (ZMP) criterion for stability, and the resulting running gaits exhibit short flight durations and low ground clearance during flight.

A quite different paradigm for control law design has been employed to induce running on RABBIT, a planar biped with revolute knees and rigid links (Morris et al., 2006). According to this framework, running gaits are ‘embedded’ in the dynamics of the robot through a set of holonomic output functions which are driven to zero by its actuators; see Westervelt et al. (2007) for a detailed overview of the method. Although running with significant flight duration and good ground clearance was successfully realized, it could not be sustained for more than six steps. Failure to maintain running in RABBIT was a consequence of its lack of compliance combined with the limitations of its actuators.

Elastic energy storage in compliant elements is of central importance in explaining the mechanics of running (Alexander, 1990; McMahon and Cheng, 1990), and is indispensable for the realization of running in legged robots (Raibert, 1986; Hurst and Rizzi, 2008). In particular, springs can store (in the first part of stance, as the leg contracts) and then release (in the second part of stance, when the leg extends) part of the energy needed to redirect the COM of the robot upwards prior to the flight phase. In the absence of springs, the actuators would have to perform negative work on impact and then supply the energy required for flight. These considerations motivated the design of MABEL, a planar bipedal robot, which incorporates compliant elements for both energy efficiency and shock absorption.

The presence of compliance, however, poses strict requirements on the control system, which must work in concert with the springs of the open-loop system to achieve closed-loop stability. To design feedback control laws that take advantage of compliant elements, the notion of compliant hybrid zero dynamics was introduced in Poulakakis (2008). The proposed method organizes the robot around a lower-dimensional physically compliant mechanical system, the spring loaded inverted pendulum (SLIP), which governs the closed-loop dynamics of the higher-dimensional system (Poulakakis and Grizzle, 2009a). The method was extended in Poulakakis and Grizzle (2009b) to induce hopping motions on the monopedal robot Thumper (a single-legged version of MABEL) and was further refined in Sreenath et al. (2011b) to produce dynamically stable walking motions experimentally on MABEL, where the designed controller preserved the natural compliant dynamics in the closed-loop ensuring the compliance performs the negative work at impact, thereby resulting in energy-efficient walking gaits. The nonlinear compliant HZD controller implemented on MABEL was instrumental in obtaining fast walking at a top sustained speed of 1.5 m/s (3.4 mph).

The notion of compliant HZD is central to controlling running on MABEL. However, contrary to walking motions, running is typically characterized by the presence of flight phases (McMahon et al., 1987), during which only limited control authority can be exercised over the system. In fact, MABEL spends approximately 40% of its running cycle in flight, leaving about 200 ms per stride for the stance phase, during which control over the system’s total energy and torso motion can be exerted. The duration of the stance phase can be effectively regulated through adjusting the leg stiffness. For example, reducing the stiffness of the leg springs can extend the stance phase duration, thereby offering enhanced control capability in continuous time through the robot’s actuators. However, as was observed in Rummel and Seyfarth (2008), in running with segmented legs that employ compliant revolute knee joints, reducing the leg stiffness can cause the robot to collapse at moderate leg compressions. Particularly in MABEL, which weighs 65 kg, extending the stance duration by reducing leg stiffness results in the leg collapsing, raising the need for effective leg compliance adjustment policies to achieve reliable highly dynamic running motions.

Leg stiffness adaptation strategies have been studied extensively in the context of biomechanics. For instance, it is known that human runners adjust their leg stiffness to maintain similar peak ground reaction forces and contact times on ground surfaces with different properties (Ferris and Farley, 1997; Ferris et al., 1998). Further, through experiments on running guinea fowl encountering unexpected terrain drops, Daley et al. (2006) and Daley and Biewener (2006) demonstrate that large perturbations of up to 40% of their hip height can be handled by changing leg stiffness. Motivated by these experiments, an active

Fig. 1. A composite illustrating the dynamic and agile running gait obtained on MABEL.
force control strategy has been suggested in Koepl et al. (2010) and an active energy removal controller has been proposed in Andrews et al. (2011) to enhance the robustness of single-leg hoppers to perturbations in ground height and ground stiffness.

In this article, we combine stiffness adaptation through active force control with dimensional reduction through motion control to introduce a family of model-based feedback controllers that induce reliable fast running gaits on compliant bipedal robots with revolute knee joints. The proposed control laws act in both continuous and discrete time to impose a set of suitably parameterized virtual non-holonomic constraints that reduce the higher-dimensional robot dynamics to a lower-dimensional hybrid dynamical system, the HZD, which not only respects the open-loop leg compliance, but also effectively tunes it throughout the gait to enhance the robustness of the controller to perturbations in the knee angle at impact. Local stability analysis via Poincaré’s method reveals that the resulting closed-loop system is exponentially stable. This controller is implemented on MABEL, both with passive feet (no ankle actuation) and with point feet, to realize stable running motions. With the passive feet, running was realized at an average speed of 1.07 m/s, while with point feet, running was realized at an average speed of 1.95 m/s and a peak speed of 3.06 m/s. About 40% of the gait was spent in flight, with an estimated peak ground clearance of 7–10 cm. Figure 1 illustrates a composite image of the running gait for MABEL.

The remainder of the paper is organized as follows. Section 2 presents a hybrid model for running that will be used for controller design. Section 3 gives an overview of the control design with Section 4 providing implementation details for achieving exponentially stable and robust running gaits. Section 5 describes the experiments performed to demonstrate the validity of the designed controller. Finally, Section 6 provides concluding remarks.

2. MABEL model

2.1. Description of MABEL

MABEL is a planar bipedal robot that is used as a testbed for the experimental validation of walking and running controller designs. It’s comprised of five links assembled to form a torso and two legs with knees; see Figure 1. The robot weighs 65 kg, has 1 m-long legs, and is mounted on a boom of radius 2.25 m. The legs are terminated in point feet. All actuators are located in the torso so that the legs are kept as light as possible; this is to facilitate rapid leg swinging for running. Unlike most bipedal robots, the actuated degrees of freedom (DOF) of each leg do not correspond to the knee and hip angles. Instead, for each leg, a collection of cable-differentials is used to connect two motors to the hip and knee joints in such a way that one motor controls the angle of the virtual leg (henceforth called the leg angle) consisting of the line connecting the hip to the toe, and the second motor is connected in series with a spring in order to control the length or shape of the virtual leg (henceforth called the leg shape); see Figure 2. Table 3 provides a glossary of symbols used in the paper. More details on
the design of MABEL can be found in Park et al. (2011), Grizzle et al. (2009), and Hurst (2008).

Springs in MABEL appear in series with an actuator. They serve to isolate the reflected rotor inertia of the leg-shape motors from the impact forces at leg touchdown and to store energy in the compression phase of a running gait, when the support leg must decelerate the downward motion of the robot’s COM; the energy stored in the spring can then be used to redirect the COM upwards for the subsequent flight phase. These properties (shock isolation and energy storage) enhance the energy efficiency of running and reduce the overall actuator power requirements. MABEL has a unilateral spring which compresses, but does not extend beyond its rest length. This ensures that springs are present when they are useful for shock attenuation and energy storage, and absent when they would be a hindrance for lifting the legs from the ground.

The following sections will develop the hybrid model appropriate for a running gait comprised of continuous phases representing stance and flight phases of running, and discrete transitions between the two.

2.2. MABEL’s unconstrained dynamics

The configuration space \( Q_e \) of the unconstrained (or extended) dynamics of MABEL is nine-dimensional: five DOF are associated with the links in the robot’s body, two DOF are associated with the springs in series with the two leg-shape motors, and two DOF are associated with the horizontal and vertical position of the robot in the sagittal plane. A set of coordinates suitable for parametrization of the robot’s linkage and transmission is

\[
q_e \ := \ (q_{L_A}; q_{mLS_a}; q_{Bsp_a}; q_{L_Asw}; q_{mLS_{sw}}; q_{Bsp_{sw}}; q_{\text{Tor}}; p_{\text{hip}}^h; p_{\text{hip}}^v)
\]

From Table 3 and the angles illustrated in Figure 2(b), \( q_{\text{Tor}} \) is the torso angle, and \( q_{L_A}, q_{mLS_a} \), and \( q_{Bsp_a} \) are the leg angle, leg-shape motor position and \( B_{\text{spring}} \) position respectively for the stance leg. The swing leg variables \( q_{L_A_{sw}}, q_{mLS_{sw}} \), and \( q_{Bsp_{sw}} \) are defined similarly. For each leg, \( q_{L_S} \) is uniquely determined by a linear combination of \( q_{mLS} \) and \( q_{Bsp} \), reflecting the fact that the cable differentials place the spring in series with the motor, with the pulleys introducing a gear ratio. The coordinates \( p_{\text{hip}}^h \) and \( p_{\text{hip}}^v \) are the horizontal and vertical positions of the hip in the sagittal plane.

The method of Lagrange is employed to obtain the equations of motion. In computing the Lagrangian, the total kinetic energy is taken to be the sum of the kinetic energies of the transmission, the rigid linkage, and the boom. The potential energy is computed in a similar manner, with the difference being that the transmission contributes to the potential energy of the system only through its gravitational potential energy. This distinction is made since it is more convenient to model the unilateral spring as an external input to the system. The resulting model of the robot’s unconstrained dynamics is determined as

\[
D_{e} (q_e) \ddot{q}_e + C_{e} (q_e, \dot{q}_e) \dot{q}_e + G_{e} (q_e) = \Gamma_e
\]

where \( D_{e} \) is the inertia matrix, \( C_{e} \) contains Coriolis and centrifugal terms, \( G_{e} \) is the gravity vector, and \( \Gamma_{e} \) is the vector of generalized forces acting on the robot, expressed as

\[
\Gamma_{e} = B_e u + E_{\text{ext}} (q_e) F_{\text{ext}} + B_{\text{fric}} \tau_{\text{fric}} (q_e, \dot{q}_e) + B_{\text{sp}} \tau_{\text{sp}} (q_e, \dot{q}_e)
\]

where the matrices \( B_e, E_{\text{ext}}, B_{\text{fric}}, \) and \( B_{\text{sp}} \) are derived from the principle of virtual work and define how the actuator torques, \( u \), the external forces, \( F_{\text{ext}} \), at the leg, the joint friction forces, \( \tau_{\text{fric}} \), and the spring torques, \( \tau_{\text{sp}} \), enter the model, respectively. The dimension of \( u \) is four, corresponding to the two brushless DC motors on each leg for actuating leg shape and leg angle.

2.3. MABEL’s constrained dynamics

The model (1) can be particularized to describe the stance and flight dynamics by incorporating proper holonomic constraints.

2.3.1. Dynamics of stance For modeling the stance phase, the stance toe is assumed to act as a passive pivot joint (no actuation, no slip, and no rebound). Thus, the coordinates of the stance leg and torso define the Cartesian position of the hip. \( (p_{\text{hip}}^h, p_{\text{hip}}^v) \). The springs in the transmission are appropriately chosen so that they are stiff enough to support the entire weight of the robot. Consequently, it is assumed that the spring on the swing leg does not deflect, that is, \( q_{Bsp_{sw}} \equiv 0 \). The stance configuration space, \( Q_s \), is therefore a co-dimension three submanifold of \( Q_e \). With these assumptions, the generalized configuration variables in stance are taken as \( q_s := (q_{L_A}; q_{mLS}; q_{Bsp}; q_{L_Asw}; q_{mLS_{sw}}; q_{Tor}) \). Defining the state vector \( x_s := (q_s; \dot{q}_s) \in TQ_s \), where \( TQ_s \) is the tangent bundle of \( Q_s \). the stance dynamics can be expressed in standard form as

\[
\dot{x}_s = f_s(x_s) + g_s(x_s) u
\]

2.3.2. Dynamics of flight In the flight phase, both feet are off the ground, and the robot follows a ballistic motion under the influence of gravity. Thus the flight dynamics can be modeled by the unconstrained dynamics developed earlier. However, in order to eliminate the stiffness in integrating the differential equations representing the flight model, an additional assumption can be made. Since the springs are stiff enough to support the entire weight of the robot, during flight when the feet are off the ground, it can be assumed that the springs are at their rest position and do
not deflect.\(^1\) Therefore, \(q_{\text{bспg}} \equiv 0\) and \(q_{\text{bспpg}} \equiv 0\). Thus, the configuration space of the flight dynamics is a co-dimension two submanifold of \(Q_0\), in other words, \(Q_0 := \{q_\epsilon \in Q_0 \mid q_{\text{bспg}} \equiv 0, q_{\text{bспpg}} \equiv 0\}\). It follows that the generalized configuration variables in the flight phase can be taken as \(q_t := (q_{\text{A}1}; q_{\text{M}1} s_{\text{u}}; q_{\text{A}2}; q_{\text{M}2} s_{\text{u}}; q_{\text{T}1} c; p_{\text{h}}; p_{\text{b}})\). Defining the state vector \(x_t := (q_t; \dot{q}_t) \in TQ_t\), where \(TQ_t\) is the tangent bundle of \(Q_t\), the flight dynamics can be expressed in standard form as

\[
\dot{x}_t = f_t(x_t) + g_t(x_t)u
\]

(4)

\[2.4. \text{MABEL's transitions} \]

\[2.4.1. \text{Stance-to-flight transition map} \]

Physically, the robot takes off when the normal component of the ground reaction force acting on the stance toe, \(F_{\text{left}}^N\), becomes zero. The ground reaction force at the stance toe can be computed as a function of the acceleration of the COM and thus depends on the inputs \(u \in U\) of the system described by (3). Mathematically, the transition occurs when the solution of (3) intersects the co-dimension one switching manifold

\[S_{s \to f} := \{x_s \in TQ_s \times U \mid f_{\text{left}}^N = 0\}\]

(5)

On transition from the stance to the flight phase, the stance leg comes off the ground and takes off. During the stance phase, the spring on the stance leg is compressed. When the stance leg comes off the ground, the spring rapidly decompresses and impacts the hard stop. The stance-to-flight transition map, \(\Delta_{s \to f} : S_{s \to f} \to TQ_t\), accounts for this. Further details are omitted for the sake of brevity and interested readers are referred to Sreenath (2011, Chapter 3).

\[2.4.2. \text{Flight-to-stance transition map} \]

The robot physically transitions from flight phase to stance phase when the swing toe contacts the ground surface. The impact is modeled here as an inelastic contact between two rigid bodies. It is assumed that there is no rebound or slip at impact. Mathematically, the transition occurs when the solution of (4) intersects the co-dimension one switching manifold

\[S_{f \to s} := \{x_f \in TQ_f \mid p_{\text{bспpg}}^w = 0\}\]

(6)

In addition to modeling the impact of the leg with the ground, and the associated discontinuity in the generalized velocities of the robot (Hürmüzlü and Chang, 1992), the transition map accounts for the assumption that the spring on the new swing leg remains at its rest length, and for the relabeling of the robot’s coordinates so that only one stance model is necessary. In particular, the transition map \(\Delta_{f \to s} : S_{f \to s} \to TQ_s\) consists of three subphases executed in the following order: (a) standard rigid impact model (Hürmüzlü and Chang, 1992); (b) adjustment of spring velocity in the new swing leg; and (c) coordinate relabeling.

\[2.5. \text{Hybrid control design model for running} \]

The hybrid model of running is based on the dynamics developed in Section 2.3 and the transition maps presented in Section 2.4, and is given by

\[
\Sigma_s : \begin{cases} 
\dot{x}_s = f_s(x_s) + g_s(x_s)u, 
& (x_s^-, u^-) \notin S_{s \to f} \\
\dot{x}_s^+ = \Delta_{s \to f}(x_s^-, u^-), 
& (x_s^-, u^-) \in S_{s \to f}
\end{cases}
\]

\[
\Sigma_f : \begin{cases} 
\dot{x}_f = f_f(x_f) + g_f(x_f)u, 
& (x_f^-, u^-) \notin S_{f \to s} \\
\dot{x}_f^+ = \Delta_{f \to s}(x_f^-), 
& x_f^- \in S_{f \to s}
\end{cases}
\]

(7)

\[2.6. \text{Validation model} \]

The model developed in the previous sections will be used for control design. However, we note that the developed model does not capture the following aspects of the experimental testbed: (a) a compliant ground and the possibility of slipping; (b) stretchy cables in the transmission of the robot; and (c) dynamics of the boom. A more detailed model was developed in Park et al. (2011) to capture these effects, however, it is not computationally tractable for use in control design for running. Instead, we will design the controller based on the model developed here and then use the detailed model for validation of the controller prior to experimental deployment.

\[3. \text{Control design for running} \]

This section presents a controller for inducing stable running motions on MABEL. To do this, the controller creates an actuated compliant HZD that enables actively adjusting the effective leg stiffness during the stance phase. Details about the implementation of this controller are relegated to Section 4.

\[3.1. \text{Overview of the control method} \]

The control objective is to design exponentially stable running periodic gaits that are robust to perturbations, so as to accommodate inevitable differences between the model and the robot. To achieve this objective, the feedback introduces control on four levels; see Figure 3. On the first level, continuous-time feedback controllers \(\Gamma_p^w\) with \(p \in \mathcal{P} := \{s, f\}\) are employed in the stance and flight phases to impose suitably parametrized virtual holonomic constraints that restrict the motion of the system on lower-dimensional invariant and attractive surfaces \(Z_{\text{inv}}\) embedded in the state space. On the second level, discrete-time feedback controllers \(\Gamma_p^w\) are employed at transitions between the stance and flight phases to render the surfaces \(Z_{\text{inv}}\) hybrid-invariant (Westervelt et al., 2007). The system in closed loop with the controllers \(\Gamma_p^w\) and \(\Gamma_p^w\) admits a well-defined HZD that governs the stability properties of the higher-dimensional robot plant.
3.2. Continuous-time control

3.2.1. Motion control. The motion controller asymptotically imposes a set of virtual holonomic constraints through feedback. Its purpose is to synchronize the links of the robot to achieve common objectives in running, such as supporting the torso, advancing the swing leg in relation to the stance leg, and specifying foot clearance. Virtual constraints can be expressed in the form of an output, that when zeroed by a feedback controller, enforces the constraint. For each phase \( p \in \mathcal{P} \) in running, the virtual constraints can then be expressed in the form

\[
y_p = h_p(q_p, \alpha_p) = H^0_{p} q_p - H^0_{d}(\theta_p, \alpha_p) \tag{8}\]

Here \( H^0_{p} \) represents a selection matrix, and \( H^0_{q} \) represents the controlled variables, corresponding to a linear combination of the configuration variables; \( H^0_{d} \) is the desired evolution that is described through Bézier polynomials parametrized by a strictly monotonic function of the joint configuration variables, \( \theta_p \), whose physical meaning will be specified in Section 4; and \( \alpha_p \) are coefficients of the Bézier polynomials. In implementing the controller, one can choose the controlled variables by selecting \( H^0_{p} \), and their corresponding desired evolution by selecting \( \alpha_p \) in (8).

To enforce the constraints, the objective of the actuators is to zero the output defined by (8). Following Isidori (1995), we differentiate the output twice with respect to time, obtaining

\[
\frac{d^2 y_p}{dt^2} = L^0_p h_p(x_p, \alpha_p) + L_{g_p} L_{f_p} h_p(q_p, \alpha_p) u_p \tag{9}
\]

where \( L_{g_p} L_{f_p} h_p(q_p, \alpha_p) \) is the decoupling matrix. Under the conditions of Westervelt et al. (2007, Lemma 5.1), the decoupling matrix has full rank, and

\[
u^*_p(x_p, \alpha_p):= - (L_{g_p} L_{f_p} h_p(q_p, \alpha_p))^{-1} L^2_p h_p(x_p, \alpha_p) \tag{10}\]

is the unique control input that renders the surface

\[
Z_{u_p} = \{ x_p \in TQ_p \mid h_p(q_p, \alpha_p) = 0, \quad L^1_p h_p(x_p, \alpha_p) = 0 \} \tag{11}
\]

invariant under the continuous dynamics for \( p \in \mathcal{P} \), in other words, for every \( z_p \in Z_{u_p} \),

\[
J^p_{u_p}(z_p) := f^p (z_p) + g^p (z_p) u^*_p(z_p) \in T_{z_p} Z_{u_p}.
\]

Zeroing the outputs effectively reduces the dimension of the system by restricting its dynamics on the surface \( Z_{u_p} \), which is called the zero dynamics manifold. The dynamics of the system restricted on \( Z_{u_p} \),

\[
\dot{z}_p = f^p_{u_p} z_{u_p}(z) \tag{12}
\]

is called the zero dynamics. To achieve attractivity of \( Z_{u_p} \), the controller (10) is modified as

\[
u_p = u^*_p(x_p, \alpha_p) - (L_{g_p} L_{f_p} h_p(q_p, \alpha_p))^{-1} \left( \frac{K_{p,p}^p}{\epsilon^2} y_p^p + \frac{K_{p,D}^p}{\epsilon} y_c^p \right) \tag{12}
\]

where \( \epsilon > 0 \) is sufficiently small, and \( K_{p,p} \) and \( K_{p,D} \) are such that \( \lambda^2 + K_{p,D} \lambda + K_{p,p} = 0 \) is Hurwitz.
**Remark 1** A modification of this control scheme that will be useful in accommodating compliance tuning during stance is to reserve one of the actuators for active force control within the zero dynamics. In more detail, during the stance phase, where four actuators are available, we will engage only three to impose virtual holonomic constraints and reserve the stance leg shape motor, \(u_{\text{mls}}\), as an input available for control within the zero dynamics. In this case, the continuous stance dynamics can be rewritten as

\[
\dot{x}_s = f_s(x_s) + g_s(x_s) \tilde{u}_s + g_{\text{mls}}(x_s) u_{\text{mls}}
\]

where \(\tilde{u}_s\) are the actuators used to enforce virtual constraints. Then, the zero dynamics becomes

\[
\dot{z}_p = f^p_s |z_{up}(z_s) + g_{\text{mls}}(z_s) u_{\text{mls}}
\]

### 3.3.1. Hybrid invariance

The controller (10) renders the zero dynamics manifold forward-invariant and attractive. However, at discrete transitions, there is no guarantee that the post-transition state may belong on the zero dynamics manifold of the subsequent phase. In particular, \(x^- = \Delta_{s \rightarrow s'}(x^-) \in Z_{\alpha_s}\) and \(x^+ = \Delta_{s \rightarrow s'}(x^+) \in Z_{\alpha_t}\) do not guarantee that \(x^+ \in Z_{\alpha_t}\), where \(x^- \in Z_{\alpha_s}\) and \(x^+ \in Z_{\alpha_t}\) are horizontal manifolds of specific states.

### 3.3.2. Active force control

The explicit appearance of \(u_{\text{mls}}\) in the zero dynamics (13) allows us to use feedback to create a virtual compliant element. In particular, by defining the feedback

\[
u_{\text{mls}}(x_s) = -k_{\text{vc}}(q_{\text{mls}} - q_{\text{mls,cc}})
\]

where \(q_{\text{mls,cc}}\), is implemented using the motor leg shape actuator. An additional damping element could be added if desired. The transmission of MABEL places this virtual compliant element in series with the physical compliance. Since both these compliances are in series, this method provides a means of dynamically varying the effective compliance of the system. For future use, note that the existence of the virtual compliant element introduces a parameter vector \(\alpha_{\text{vc}} = (k_{\text{vc}}, q_{\text{mls,cc}})\).

To provide some intuition, virtual compliance facilitates energy injection to enable takeoff and effectively accounts for the softening of the leg spring as the knee bends, as observed in Rummel and Seyfarth (2008), thereby preventing the stance knee from excessively bending. Beyond the control of running, this method of creating a virtual compliant element was instrumental in maintaining good ground-contact forces for large step-down walking experiments (see Park et al. (2011) for 5 inches step-down, and Park et al. (2012) for up to 8 inches step-down). Furthermore, as will be seen in Section 4.6, virtual compliance can easily account for cable stretch and for asymmetry of the robot due to the boom, which are not included in the model for control.

### 3.3.3. Discrete-time control

**3.3.1. Hybrid invariance**

The controller (10) renders the zero dynamics manifold forward-invariant and attractive. However, at discrete transitions, there is no guarantee that the post-transition state may belong on the zero dynamics manifold of the subsequent phase. In particular, \(x^- = \Delta_{s \rightarrow s'}(x^-) \in Z_{\alpha_s}\) and \(x^+ = \Delta_{s \rightarrow s'}(x^+) \in Z_{\alpha_t}\) do not guarantee that \(x^+ \in Z_{\alpha_t}\), where \(x^- \in Z_{\alpha_s}\) and \(x^+ \in Z_{\alpha_t}\) are horizontal manifolds of specific states.

**3.3.2. Active force control**

The explicit appearance of \(u_{\text{mls}}\) in the zero dynamics (13) allows us to use feedback to create a virtual compliant element. In particular, by defining the feedback

\[
u_{\text{mls}}(x_s) = -k_{\text{vc}}(q_{\text{mls}} - q_{\text{mls,cc}})
\]

where \(q_{\text{mls,cc}}\), is implemented using the motor leg shape actuator. An additional damping element could be added if desired. The transmission of MABEL places this virtual compliant element in series with the physical compliance. Since both these compliances are in series, this method provides a means of dynamically varying the effective compliance of the system. For future use, note that the existence of the virtual compliant element introduces a parameter vector \(\alpha_{\text{vc}} = (k_{\text{vc}}, q_{\text{mls,cc}})\).

To provide some intuition, virtual compliance facilitates energy injection to enable takeoff and effectively accounts for the softening of the leg spring as the knee bends, as observed in Rummel and Seyfarth (2008), thereby preventing the stance knee from excessively bending. Beyond the control of running, this method of creating a virtual compliant element was instrumental in maintaining good ground-contact forces for large step-down walking experiments (see Park et al. (2011) for 5 inches step-down, and Park et al. (2012) for up to 8 inches step-down). Furthermore, as will be seen in Section 4.6, virtual compliance can easily account for cable stretch and for asymmetry of the robot due to the boom, which are not included in the model for control.

### 3.3.4. Robustness to disturbances

The control construction so far render the desired periodic running gait locally exponentially stable. To enhance the robustness of our control scheme, an additional event-based controller is introduced to update a set of parameters \(\gamma \in \mathcal{G}\), which includes
parameters to modify the virtual compliance stiffness, and swing height. The nonlinear controller that is used to modify the $\gamma$ parameters is detailed in Section 4.5. We only mention that the control design is motivated by insight obtained in the context of controlling simpler hopper models, such as the SLIP. Special attention is paid to ensure the exponential stability property is preserved under the action of the controller by studying the properties of the Poincaré map, $P_\gamma : S_\gamma \times B \times G \to S_\gamma$, that includes all four layers of control.

4. Controller implementation details

4.1. Virtual constraint design for stance

During stance, the objective of the controller is threefold. First, it ensures that the torso enters the flight phase with suitable initial conditions so that excessive torso pitching is avoided. Second, it guarantees sufficient ground clearance of the swing leg to allow its proper positioning in anticipation of touchdown. And third, it creates a virtual compliance element that effectively ‘tunes’ the physical leg stiffness to offer enhanced control authority during the stance phase.

The first two objectives of the stance control action can be achieved by devoting three out of the four available actuators to impose virtual holonomic constraints on the torso motion captured by its pitch angle $q_{\text{Tor}}$ and on the motion of the swing leg described by the angles $q_{\text{LA}_\text{sw}}$ and $q_{\text{mL}_\text{sw}}$. Hence, in defining the output (8) for stance, we choose the controlled variables as

$$H^s_0 q_s = (q_{\text{LA}_\text{sw}}, q_{\text{mL}_\text{sw}}, q_{\text{Tor}})^T$$

(19)

The virtual constraints imposed on the control variables (19) are parametrized by fifth-order Bézier polynomials through the monotonically increasing angle $\theta_s$, formed by the virtual leg connecting the toe with the hip relative to the ground,

$$\theta_s(q_s) = \pi - q_{\text{LA}_\text{sw}} - q_{\text{Tor}}$$

(20)

(see Figure 2(a)). The detailed design of the constraints is documented in Sreenath (2011, Section 6.3). We only mention here that substantial torso control can be developed only during stance, due to the fact that the angular momentum about the COM is conserved in the flight phase. To avoid excessive pitching motions during the ensuing flight phase, the corresponding virtual holonomic constraint imposed on $q_{\text{Tor}}$ is designed to drive the torso so that at the end of the stance phase it leans forward with a backward angular velocity. This is important because a forward torso velocity at the beginning of flight would result in an excessive forward pitch at the end of flight due to the conservation of angular momentum, requiring correction of a large torso error during the relatively short (compared to the walking motions in Sreenath et al., 2011b) stance phase.

To realize an actively tuned virtual compliant component as described in Section 3.2.2, we make use of the fourth actuator, $u_{\text{mL}_\text{sw}}$, which is available for control.

4.2. Virtual constraint design for flight

During flight, the controller serves two purposes. First, it rapidly lifts the stance leg to avoid toe stubbing at the early stages of its swing phase. Second, it positions the swing leg, whose touchdown is anticipated, at a proper absolute angle. To achieve these objectives, all four actuators will be recruited to enforce suitable virtual holonomic constraints by zeroing the output functions (8), in which the controlled variables are chosen as

$$H^f_0 q_f = (q_{\text{mL}_\text{sw}}, q_{\text{LA}_\text{sw}} + q_{\text{Tor}}, q_{\text{mL}_\text{sw}}, q_{\text{LA}_\text{sw}})^T$$

(21)

where $(q_{\text{mL}_\text{sw}}, q_{\text{LA}_\text{sw}})$ refer to the coordinates of the stance leg (the leg that was in stance and switched to swing for the flight phase). Similarly to the stance phase, fifth-order Bézier polynomials are employed to design the virtual holonomic constraints. The polynomials are parametrized based on the monotonically increasing quantity $\theta_f$, which corresponds to the horizontal position of the hip,

$$\theta_f(q_f) = \rho^h_{\text{hip}}$$

(22)

4.3. Event transitions

Transitions between continuous-time phases offer the possibility of updating certain parameters that are introduced through the virtual constraints (e.g., Bézier polynomial coefficients) to achieve the control objectives, such as the hybrid invariance condition described in Section 3.3.1. Up to this point, we have considered two transitions, which are imposed by the physics of the robot running; namely, the stance-to-flight and the flight-to-stance transitions occurring at the switching surfaces $S_{s\to f}$ and $S_{f\to s}$, and governed by the transition maps $\Delta_{s\to f}$ and $\Delta_{f\to s}$, respectively; see Section 2 for definitions.

In addition to the transitions separating the stance and flight phases, we will further divide stance into two subphases: stance-compression (sc) and stance-decompression (sd). The transition from sc to sd occurs at the switching surface

$$S_{sc\to sd} = \{s_0 \in TQ_s \mid H_{sc\to sd}(s_0) = 0\}$$

(23)

where the threshold function is $H_{sc\to sd} := \theta_s - \theta_{sd}$, with $\theta_s$ defined by (20) and $\theta_{sd}$ a constant. The corresponding transition map is the identity map (i.e., $\Delta_{sc\to sd} := \text{id}$), reflecting the fact that the state does not change as the robot passes from stance-compression to -decompression.
In contrast to the state that remains unchanged through $S_{sc\rightarrow sd}$, certain parameters characterizing the stance subphases can be updated as the controller switches from compression to decompression. In particular, the stiffness and rest position, $\alpha_{vc} = (k_{vc}, q_{mLS_{vc}})$, of the virtual compliant element (14) introduced through the stance-phase continuous control action of Section 3.2.2 can have different values during the two stance subphases. Hence, in the $sc$ and $sd$ phases, $\alpha_{vc}$ will be chosen as $\alpha_{vc}^{sc}$ and $\alpha_{vc}^{sd}$ respectively, with $\alpha_{vc}^{sc} \neq \alpha_{vc}^{sd}$. The update ensures that the parameters are only changed at transitions, in other words $\alpha_{vc} = 0$ for the continuous dynamics. Intuitively, this parameter update facilitates energy injection during $sd$ to enable lift-off.

### 4.4. Gait design through optimization

A periodic running gait is designed through an optimization procedure that selects the parameters introduced by the virtual constraints and the virtual compliance element to minimize energy consumption per step, subject to constraints to meet periodicity as well as workspace and actuator limitations. In more detail, the cost function employed is

$$J_{nom}(\alpha_{sc}, \alpha_{sf}, \alpha_{sc}^{sd}, \alpha_{sf}^{sd}) = \frac{1}{p_{stb_{sc}}(q_i)} \int_0^{T_f} ||u(t)||^2 dt \tag{24}$$

where $T_f$ is the step duration (stance plus flight time) and $p_{stb_{sc}}$ is the step length. Minimizing this cost function tends to reduce peak torque demands and minimize the electrical energy consumed per step. The nonlinear constrained optimization routine `fmincon` of MATLAB’s Optimization Toolbox is used to perform the numerical search for desired gait, optimizing 31 different parameters; further details can be found in Sreenath (2011).

Following this procedure, a nominal periodic running gait at 1.34 m/s is obtained. Figure 4 depicts the virtual constraints for the stance and flight phases, along with other configuration variables, during one step of running. The squares on the plots indicate the transition from stance to flight configuration variables, during one step of running. The swing leg

Figure 5 illustrates the actuator torques used to realize the gait, and all motor torques are well within the capacity of the actuators, namely 30 Nm. The stance leg shape torque is relatively large, initially to support the weight of the robot as the stance knee bends, and subsequently to inject energy in the $sd$ phase to achieve lift-off. Note that the stance motor leg shape torque is discontinuous at the $sc$ to $sd$ transition due to an instantaneous change in the parameters $\alpha_{vc}$ of the virtual compliance. All torques are discontinuous at the stance-to-flight transition due to the impact of the spring with the hard-stop; see Figure 2(b).

Figure 6 illustrates the evolution of the swing leg height and the vertical position of the COM of the robot. The swing foot is over 15 cm above the ground at its peak to offer good ground clearance for hard impacts. During the stance phase, the COM undergoes an asymmetric motion with the lowest point of potential energy being around 52% into the stance phase. During the flight phase, the COM has a ballistic trajectory. As noted in McMahon and Cheng (1990) and Holmes et al. (2006), both these aspects of COM motion are dominant characteristics of running. Finally, Figure 7 illustrates the vertical component of the ground reaction force. Immediately upon impact, during the $sc$ phase, there is a peak in the ground reaction force due to the spring compressing rapidly on impact. During most of the $sc$ phase, the force is fairly constant. On transition to the $sd$ phase, the energy injection causes the force to rapidly increase at first and then go to zero, at which point stance-to-flight transition occurs.

### 4.5. Parameter update strategies

#### 4.5.1. Exponential stability

To analyze the stability of the running gait obtained in Section 4.4 in closed loop with the continuous-time controllers (12), we employ the method of Poincaré. Let $S_{sc\rightarrow sd}$ be the Poincaré section. Then the stability properties of the periodic running orbit can be captured by the stability properties of the corresponding fixed point of the restricted Poincaré map $\rho : S_{sc\rightarrow sd} \cap Z_{\alpha_{sc}^{sd}} \rightarrow S_{sc\rightarrow sd} \cap Z_{\alpha_{sc}^{sd}}$; see Morris and Grizzle (2005, 2009). Numerical computations of the eigenvalues of the linearization of the restricted Poincaré map about the fixed-point of interest reveals that the corresponding running gait is unstable with a dominant eigenvalue of magnitude 1.19. In fact, all the running gaits we have been able to compute were unstable.

To locally exponentially stabilize the desired running gait, we introduce the additional outer-loop discrete-time controller $\Gamma^\beta$. These parameters are a subset of those introduced through the continuous-time control action (namely, $\alpha_{sc}, \alpha_{sf}, \alpha_{vc}^{sc}, \alpha_{vc}^{sd}$) and are denoted by $\beta$ to emphasize the fact that they are updated via the loop $\Gamma^\beta$ of Figure 3 to ensure stability. The parameters $\beta$ include the stiffness and rest position of the virtual compliant element (14), which
This controller updates $\beta$ each time the surface $S_{sc \rightarrow sd}$ is crossed, ensuring that the eigenvalues of linearization of the closed-loop Poincaré map are all within the unit circle; for the particular design implemented here, the dominant eigenvalue has magnitude 0.8383, concluding that the fixed point is locally exponentially stable.

Figure 8 shows three steps of the running gait under the controller that includes $\Gamma^\nu$, $\Gamma^\rho$, and $\Gamma^\beta$. The obtained motion can continue indefinitely in simulation.

4.5.2. Robustness to perturbations

The control design proposed so far combines continuous- and discrete-time control to exponentially stabilize the system, accommodating perturbations in the torso pitch angle of up to $6^\circ$ in both the forward and backward directions. While this performance in stabilizing the torso is adequate for experimental implementation, the controller in its current form cannot reject errors in the stance leg shape that exceed $5^\circ$ at impact; see Figure 9(a). This observation motivates the additional

enables modification of the energy stored during compression and injected during decompression; $\beta_{k_v}^{p}$, $\beta_{mLS_v}^{p}$ for $p \in \{sc, sd\}$. In addition, they include the touchdown angle $\beta_{TD}$ of the swing leg to regulate the forward running speed, the torso angle at lift-off $\beta_{Tor}$ that influences the initial conditions of the ensuing flight phase, and an offset $\beta_{Bp_{st}}$ that is added to $\theta_{f}^{-}$ to change the position of lift-off. In summary,

$$\beta = (\beta_{k_v}^{sc}, \beta_{k_v}^{sd}, \beta_{TD}, \beta_{mLS_v}^{sc}, \beta_{mLS_v}^{sd}, \beta_{Bp_{st}}, \beta_{Tor}) \in B$$

includes the parameters that are updated in an event-based fashion by the component $\Gamma^\beta$ of the control law.

For the design and experimental implementation of $\Gamma^\beta$, the full-order Poincaré map is considered. The switching surface in the definition of the Poincaré map $P_\beta$ in Section 3.3.2 is chosen as $S_{\beta} = S_{sc \rightarrow sd}$, resulting in the discrete-time nonlinear control system (16) with $\beta$ appearing as its input. The controller $\Gamma^\beta$ corresponds to the discrete LQR (18) designed based on the linearization (17) of (16) about the fixed point $(x^{-}, \beta^{-})$, as was discussed in Section 3.3.2.

**Fig. 4.** Evolution of the virtual constraints and configuration variables for a nominal fixed point (periodic running gait) at a speed of 1.34 m/s and step length of 0.7055 m. The squares indicate the location of the transition from stance to flight phase. The circle on the $q_{Bp_{st}}$ plot illustrates the location of the stance-compression-to-stance-decompression event transition.
Fig. 5. Actuator torques corresponding to the nominal fixed point. The squares illustrate the location of the transition from stance to flight phase. The circle on the $u_{mLS}$ plot illustrates the location of the stance-compression-to-stance-decompression event transition. Note that the torques are discontinuous at stance-to-flight transitions. Also note the additional discontinuity for $u_{mLS}$ at the stance-compression-to-stance-decompression event transition due to the instantaneous change in the offset for the virtual compliance at this transition.

Fig. 6. Evolution of swing leg height and vertical center of mass (COM) of the robot for the nominal fixed point. The COM trajectory clearly illustrates the lowest point of potential energy during the stance phase and the ballistic trajectory in the flight phase, both of which are dominating characteristics of running. The squares illustrate the location of the transition from stance to flight phase.

control layer $\Gamma^\nu$ of Figure 3, which, as was mentioned in Section 3.3.3, is added to improve the robustness of our control design to perturbations in the knee angle at impact.

To implement this controller, a number of parameters detailed below will be updated by $\Gamma^\nu$ on entry to the stance phase; that is, at the switching surface $S_\gamma = \Delta_{s\rightarrow f}(S_{f\rightarrow s}) \subset TQ_s$, where $S_{f\rightarrow s}$ is the flight-to-stance switching surface defined by (6) and $\Delta_{s\rightarrow f}$ is the corresponding transition map. The motivation for considering the touchdown event is that it provides an immediate response to errors arising in the preceding flight phase, such as landing with an excessively bent knee, or velocity mismatch caused by imperfections in the ground-contact model.

We continue our discussion on this additional control component $\Gamma^\nu$ by providing some intuition. First note that, to produce the same leg force, the compression required in a segmented revolute-knee leg with joint compliance is larger than that required in a prismatic leg. This phenomenon was observed in Rummel and Seyfarth (2008), and, in the context of MABEL, implies that the stiffness of the virtual compliant element should be modified (i.e. increased) to prevent the leg from excessively bending to develop sufficient force for supporting the weight of the robot. Furthermore, the swing leg may have to contract additionally to ensure sufficient ground clearance in the presence of shorter stance leg lengths. To accommodate these requirements, the
Fig. 7. Vertical component of the ground reaction force for the nominal running fixed point. At the stance-compression-to-stance-decompression event transition (indicated by the circle), the change in the offset for the virtual compliance causes the spring to compress further which increases the ground reaction force considerably. Takeoff occurs when the ground reaction force goes to zero (indicated by the square).

Fig. 8. Stick figure plot of three steps of running. The stance leg is illustrated in red, while the swing leg is illustrated in blue. Stick figures with darker shades are in flight phase, while those with lighter shades are in stance phase. From the stick figure it can be easily deduced that the flight phase lasts around 30% of the gait.

virtual compliance stiffness, $\gamma_{sc}^c$, as well as the knee angle of the swing leg, $\gamma_{LSsw}$, will be updated according to

$$\begin{align*}
\gamma_{sc}^c &= \begin{cases}
K_{ksc} (q_{LSst}^{++} - q_{LSst}^{+}) , & q_{LSst}^{+} - q_{LSst}^{++} > 0 \\
0, & \text{otherwise}
\end{cases} \\
\gamma_{LSsw} &= \begin{cases}
K_{LSsw} (q_{LSst}^{++} - q_{LSst}^{+}) , & q_{LSst}^{+} - q_{LSst}^{++} > 0 \\
0, & \text{otherwise}
\end{cases}
\end{align*}$$

(25)

where $q_{LSst}^{++}$ denotes the stance leg shape angle right after touchdown, and $q_{LSst}^{+}$ its nominal value. The gains $K_{ksc}$ and $K_{LSsw}$ are tuned through simulations, and their values are provided in Table 1.

An additional corrective action embedded in $\Gamma^\gamma$ regards the regulation of the forward running speed. To do this, $\Gamma^\gamma$ updates a parameter $\gamma_{tor}$, which shapes the virtual holonomic constraint imposed on the torso motion at the beginning of stance, based on the difference between the current forward speed and its nominal value. This allows leaning of

<table>
<thead>
<tr>
<th>Gain parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_{ksc}$</td>
<td>0.46</td>
</tr>
<tr>
<td>$K_{LSsw}$</td>
<td>1.5</td>
</tr>
<tr>
<td>$K_{tor}$</td>
<td>0.16</td>
</tr>
<tr>
<td>$K_{tor}$</td>
<td>0.31</td>
</tr>
<tr>
<td>$K_{\delta_{sc\rightarrow sd}}$</td>
<td>-0.37</td>
</tr>
</tbody>
</table>
the torso forward to increase speed, or backward to decrease speed, and is implemented through the prescription
\[
\gamma_{\text{tor}} = \begin{cases} 
K_{\text{tor}}^+ (p_{\text{hip}}^{h,x+} - p_{\text{hip}}^{h,x++}), & (p_{\text{hip}}^{h,x+} - p_{\text{hip}}^{h,x++}) > 0 \\
K_{\text{tor}}^- (p_{\text{hip}}^{h,x+} - p_{\text{hip}}^{h,x++}), & \text{otherwise}
\end{cases}
\] (26)

where \(p_{\text{hip}}^{h,x+}\) is the horizontal speed of the hip at impact with the ground, \(p_{\text{hip}}^{h,x+}\) is its nominal value, and gains \(K_{\text{tor}}^+\) and \(K_{\text{tor}}^-\) are provided in Table 1. As speed increases, the energy injected during the sd phase decreases because the time spent in this phase decreases with increasing speed. To account for this, the controller \(\Gamma^\gamma\) will update one more parameter, namely \(\gamma_{\text{sc}}\rightarrow\text{sd}\), that modifies the location of transition from xc to sd to increase or decrease the period over which energy is injected in the sd phase. This is achieved through
\[
\gamma_{\text{sc}}\rightarrow\text{sd} = \begin{cases} 
K_{\text{sc}}(p_{\text{hip}}^{h,x+} - p_{\text{hip}}^{h,x++}), & (p_{\text{hip}}^{h,x+} - p_{\text{hip}}^{h,x++}) > 0 \\
0, & \text{otherwise}
\end{cases}
\] (27)

where \(p_{\text{hip}}^{h,x+}\) and \(p_{\text{hip}}^{h,x++}\) have the same meaning as in (26) and \(K_{\text{sc}}\rightarrow\text{sd}\) is provided in Table 1. In summary,
\[
\gamma = (\gamma_{\text{sc}}\rightarrow\text{xc}, \gamma_{\text{tor}}, \gamma_{\text{LSsw}}, \gamma_{\text{sc}}\rightarrow\text{sd}) \in G
\]
includes the parameters that are updated in an event-based fashion by the component \(\Gamma^\gamma\) of the control law.

Under the influence of \(\Gamma^\gamma\), the robustness to perturbations is increased and, as shown in Figure 9(b), perturbations up to 5\(^\circ\) in the impact leg-shape angle (the knee being bent an additional 10\(^\circ\)) can be rejected. The stability of the closed-loop system can be analyzed through the eigenvalues of the linearization of the Poincaré map \(P_\gamma\) introduced in Section 3.3.3; more details can be found in Appendix B where it is shown that the linearized Poincaré map has a dominant eigenvalue of magnitude 0.6072 indicating that the closed-loop system with the additional component \(\Gamma^\gamma\) is locally exponentially stable.

Remark 2 Note that the controllers \(\Gamma^a, \Gamma^\infty,\) and \(\Gamma^\beta\) have been designed through rigorous control synthesis approaches, whereas the design of the outermost control loop, \(\Gamma^\gamma\), has been based on heuristics. It is noted that the controllers \(\Gamma^a, \Gamma^\infty,\) and \(\Gamma^\beta\) achieve stable running in simulations on the design model. The controller \(\Gamma^\gamma\) aids in the experimental validation of running by increasing the closed-loop system’s robustness to perturbations in the knee angle at impact and to imperfections in the ground-contact model. Section 5.4 provides additional comments in this regard.

4.6. Preparing for experimental deployment

One aspect that needs to be incorporated in the control design prior to experimental deployment on MABEL is cable stretch. In the leg-shape coordinates, cable stretch reaches a peak value of almost 15\(^\circ\) just prior to lift-off, which, given that the nominal peak leg-shape is around 25\(^\circ\) (see Figure 4), amounts for over 60\% of motion in the knee, thus further amplifying knee bending. To account for this
issue, the nominal controller design will be modified. In more detail, the compliance due to cable stretch will be modeled as a spring–damper system placed in series with the physical spring (Bspring) and the motor leg shape actuator in the transmission mechanism. Then, active force control on the stance leg can be used to modify the virtual compliance, kvc, so that the compliance due to the cable stretch, kcable, together with kvc, has the value of the effective compliance, k*, obtained through the optimization procedure detailed in Section 4.4; in other words,

$$\frac{1}{k_{\text{cable}}} + \frac{1}{k_{\text{vc}}} = \frac{1}{k^*}$$  (28)

With this modification, the effective compliance of the leg is now the same as for that without cable stretch; in other words, cable stretch has been accounted for by the control design. Table 2 provides the values for the various compliances discussed in this section.

This modified running controller is next validated on the detailed model as mentioned in Section 2.6, and is ready for experimental deployment.

### 5. Running experiments

This section documents experimental implementations of the running controller developed in Sections 3 and 4. To illustrate the power and limitations of the proposed method, three experiments are presented. The first experiment details the execution of a transition controller that transitions from walking to running, the second experiment details a running experiment, and finally the third experiment details the transition from running to walking. Videos of the running experiments are available on YouTube (Sreenath et al., 2011a,c).

#### 5.1. Experiment 1: Two-step transition from walking to running

Running on MABEL can be implemented by transitioning from walking. As in Westervelt et al. (2003), to transition from walking to running the controller modifies the virtual constraints corresponding to a walking gait so that, by the end of a walking step, they are closer to the virtual constraints of the targeted running gait. Instead of a one-step transition from walking to running as was done in Morris et al. (2006), a two-step transition is implemented to enable

<table>
<thead>
<tr>
<th>Source of compliance</th>
<th>Stiffness value</th>
</tr>
</thead>
<tbody>
<tr>
<td>kBsp</td>
<td>3.17</td>
</tr>
<tr>
<td>kcable</td>
<td>2.46</td>
</tr>
<tr>
<td>k*vc</td>
<td>1.66</td>
</tr>
<tr>
<td>kvc</td>
<td>5.08</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source of compliance</th>
<th>Stiffness value</th>
</tr>
</thead>
<tbody>
<tr>
<td>kBsp</td>
<td>3.17</td>
</tr>
<tr>
<td>kcable</td>
<td>2.46</td>
</tr>
<tr>
<td>k*vc</td>
<td>1.66</td>
</tr>
<tr>
<td>kvc</td>
<td>5.08</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>αβ</td>
<td>Coefficients for Bézier polynomial for p ∈ Pβ</td>
</tr>
<tr>
<td>αγ</td>
<td>Coefficients for correction polynomial</td>
</tr>
<tr>
<td>kvc</td>
<td>Parameter vector for the virtual compliant</td>
</tr>
<tr>
<td>β</td>
<td>Parameter vector for Γβ</td>
</tr>
<tr>
<td>γ</td>
<td>Parameter vector for Γγ</td>
</tr>
<tr>
<td>Δs→f</td>
<td>Stance-to-flight impact map</td>
</tr>
<tr>
<td>Δf→s</td>
<td>Flight-to-stance impact map</td>
</tr>
<tr>
<td>Γ′β</td>
<td>Continuous-time controller for p ∈ Pβ</td>
</tr>
<tr>
<td>Γ′γ</td>
<td>Discrete-time controller for hybrid invariance</td>
</tr>
<tr>
<td>Bc</td>
<td>Input matrix for unconstrained dynamics</td>
</tr>
<tr>
<td>Cc</td>
<td>Coriolis matrix for unconstrained dynamics</td>
</tr>
<tr>
<td>Dc</td>
<td>Inertia matrix for unconstrained dynamics</td>
</tr>
<tr>
<td>F1,0</td>
<td>Stance toe normal ground reaction force</td>
</tr>
<tr>
<td>fs, ft</td>
<td>Drift vector field for stance, flight dynamics</td>
</tr>
<tr>
<td>Ge</td>
<td>Gravity matrix for unconstrained dynamics</td>
</tr>
<tr>
<td>gs, gt</td>
<td>Input vector field for stance, flight dynamics</td>
</tr>
<tr>
<td>gmLSst</td>
<td>Input vector field for stance motor leg shape</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>αβ</td>
<td>Coefficients for Bézier polynomial for p ∈ Pβ</td>
</tr>
<tr>
<td>αγ</td>
<td>Coefficients for correction polynomial</td>
</tr>
<tr>
<td>kvc</td>
<td>Parameter vector for the virtual compliant</td>
</tr>
<tr>
<td>β</td>
<td>Parameter vector for Γβ</td>
</tr>
<tr>
<td>γ</td>
<td>Parameter vector for Γγ</td>
</tr>
<tr>
<td>Δs→f</td>
<td>Stance-to-flight impact map</td>
</tr>
<tr>
<td>Δf→s</td>
<td>Flight-to-stance impact map</td>
</tr>
<tr>
<td>Γ′β</td>
<td>Continuous-time controller for p ∈ Pβ</td>
</tr>
<tr>
<td>Γ′γ</td>
<td>Discrete-time controller for hybrid invariance</td>
</tr>
<tr>
<td>Bc</td>
<td>Input matrix for unconstrained dynamics</td>
</tr>
<tr>
<td>Cc</td>
<td>Coriolis matrix for unconstrained dynamics</td>
</tr>
<tr>
<td>Dc</td>
<td>Inertia matrix for unconstrained dynamics</td>
</tr>
<tr>
<td>F1,0</td>
<td>Stance toe normal ground reaction force</td>
</tr>
<tr>
<td>fs, ft</td>
<td>Drift vector field for stance, flight dynamics</td>
</tr>
<tr>
<td>Ge</td>
<td>Gravity matrix for unconstrained dynamics</td>
</tr>
<tr>
<td>gs, gt</td>
<td>Input vector field for stance, flight dynamics</td>
</tr>
<tr>
<td>gmLSst</td>
<td>Input vector field for stance motor leg shape</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>αβ</td>
<td>Coefficients for Bézier polynomial for p ∈ Pβ</td>
</tr>
<tr>
<td>αγ</td>
<td>Coefficients for correction polynomial</td>
</tr>
<tr>
<td>kvc</td>
<td>Parameter vector for the virtual compliant</td>
</tr>
<tr>
<td>β</td>
<td>Parameter vector for Γβ</td>
</tr>
<tr>
<td>γ</td>
<td>Parameter vector for Γγ</td>
</tr>
<tr>
<td>Δs→f</td>
<td>Stance-to-flight impact map</td>
</tr>
<tr>
<td>Δf→s</td>
<td>Flight-to-stance impact map</td>
</tr>
<tr>
<td>Γ′β</td>
<td>Continuous-time controller for p ∈ Pβ</td>
</tr>
<tr>
<td>Γ′γ</td>
<td>Discrete-time controller for hybrid invariance</td>
</tr>
<tr>
<td>Bc</td>
<td>Input matrix for unconstrained dynamics</td>
</tr>
<tr>
<td>Cc</td>
<td>Coriolis matrix for unconstrained dynamics</td>
</tr>
<tr>
<td>Dc</td>
<td>Inertia matrix for unconstrained dynamics</td>
</tr>
<tr>
<td>F1,0</td>
<td>Stance toe normal ground reaction force</td>
</tr>
<tr>
<td>fs, ft</td>
<td>Drift vector field for stance, flight dynamics</td>
</tr>
<tr>
<td>Ge</td>
<td>Gravity matrix for unconstrained dynamics</td>
</tr>
<tr>
<td>gs, gt</td>
<td>Input vector field for stance, flight dynamics</td>
</tr>
<tr>
<td>gmLSst</td>
<td>Input vector field for stance motor leg shape</td>
</tr>
</tbody>
</table>

(Continued)
Fig. 10. Experimental plots of the internal phase variable, joint angles, and motor torques for (a) transition from walking to running (Experiment 1) and (b) transition from running to walking (Experiment 3). The internal phase variable of the controller indicates the walking and running parts of the gait, with the thicker plots indicating the transition steps. Note that for transition to running there are two transition steps: one during walking and the other during running, while for transition to walking there is one transition step during walking. Also note that the peak spring compression for running is around 2.5 times that for walking.

Table 3. (Continued)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{s}^{-}$ - $x_{f}^{-}$</td>
<td>Pre-transition stance, flight state</td>
</tr>
<tr>
<td>$x_{s}^{+}$ - $x_{f}^{+}$</td>
<td>Post-transition stance, flight state</td>
</tr>
<tr>
<td>$y_{p}$</td>
<td>Virtual constraints for $p \in \mathcal{P}$</td>
</tr>
<tr>
<td>$y_{p}^{c}$</td>
<td>Virtual constraints with corrections for $p \in \mathcal{P}$</td>
</tr>
<tr>
<td>$z_{\alpha_{p}}$</td>
<td>Zero dynamics surface for $p \in \mathcal{P}$</td>
</tr>
<tr>
<td>$z_{p}$</td>
<td>State on zero dynamics surface for $p \in \mathcal{P}$</td>
</tr>
</tbody>
</table>

a smoother transition by preventing rapid torso motions on MABEL. This is especially important for gaits where the final and initial values of the torso virtual constraint differ significantly between the walking and running fixed points, respectively. A walking-to-running transition then consists of the following: (a) a transition from the nominal walking gait to a transition-walk-step, followed by (b) a transition from the transition-walk-step to a transition-run-step, and finally (c) a transition from the transition-run-step to the nominal running gait.

Figure 10(a) illustrates plots of various variables for the transition from walking to running. The walking and running sections are clearly marked along with the two transition steps.

5.2. Experiment 2: Running with point feet

Initial experiments on MABEL failed to achieve steady-state running due to foot slippage and the controller’s poor performance in regulating forward speed. This is a consequence of imperfections in the ground-contact model used in the controller design. To address these issues, the point feet were replaced with passive feet with shoes to provide a larger surface area for traction, thereby preventing slipping. With this configuration, successful running was achieved (see Appendix A for more details on these experiments), suggesting certain modifications to the running controller.
of Sections 3 and 4 in order to achieve running on point feet.

In more detail, to regulate the forward speed, the \( \gamma \)-parameter corresponding to the virtual compliance is modified as in (31) and saturation in the \( \beta \)-parameter corresponding to the touchdown angle is introduced as in (32); see Appendix A. Finally, the \( \gamma \)-parameter that modifies the location of the sc to sd phase will also be saturated as a function of the speed as

\[
gamma_{\text{sc} \rightarrow \text{sd}}^{\text{sat}} = \begin{cases} 
0.2, & 0 \leq \dot{\theta}_{h,\text{avg}} \leq 2 \\
0.25, & 2 \leq \dot{\theta}_{h,\text{avg}} \leq 2.5 \\
0.35, & 2.5 \leq \dot{\theta}_{h,\text{avg}} 
\end{cases}
\]  

(29)

At high speeds, the time spent in the sd phase decreases, which results in less energy being injected and smaller push-offs. With the above modification, a well-defined flight phase is maintained even during fast running motions. Next, to prevent the sd phase from causing a lift-off with a high velocity, the sd-to-flight-phase switching surface is modified as follows

\[
S_{\text{sd} \rightarrow \text{f}}^{\text{exp}} := S_{\text{sd} \rightarrow \text{f}} \cap \{ x_s \in TQ_s \mid \dot{\theta}_{h} > \dot{\theta}_{h}^{\text{sd} \rightarrow \text{f}} \}
\]  

(30)

In addition, during the sd phase, the torso is pushed back in a similar manner to that in the running-with-feet experiment. Finally, during the flight phase, the adaptive correction polynomials, as used for the running-with-feet experiment, are deployed. Both these changes counteract the effect of unmodeled cable stretch in the leg angle direction.

With these changes to the controller developed in Sections 3 and 4, the running experiment is carried out as follows. First, walking is initiated on MABEL using the walking controller developed in Sreenath et al. (2011b). Next, the walking-to-running transition controller, presented in Section 5.1, is executed. Finally, on transition to running, the running controller is executed. The running controller induced stable running at an average speed of 1.95 m/s, and a peak speed of 3.06 m/s; 113 running steps were obtained and the experiment terminated when the power to the robot was cut off. At 2 m/s, the average stance and flight times of 233 ms and 126 ms respectively were obtained, corresponding to a flight phase that is 35% of the gait. At 3 m/s, the average stance and flight times of 195 ms and 123 ms respectively were obtained, corresponding to a flight phase that is 39% of the gait. An estimated ground clearance of 3–4 inches (7.5–10 cm) is obtained. The specific cost of mechanical transport \( c_{\text{mt}} \), defined in Collins and Ruina (2005), was computed to be 1.07.

Figure 11(a) depicts snapshots at 100 ms intervals of a typical running step. Figure 12(a) depicts the mean joint angles, and motor torques temporally normalized over time, for 50 consecutive steps of running.

The outer-loop event-based controller parameters are depicted in Figures 13(a) and 14(a). Considerable variation in the speed is observed. In particular, when the speed exceeds 2.5 m/s, large changes in the touchdown angle, \( \theta_{TD} \), and the \( \gamma \)-parameter that affects the transition from sc to sd, \( \gamma_{\text{sc} \rightarrow \text{sd}} \), causes the speed to dramatically drop to under
Fig. 12. Ensemble plots of joint angles and motor torques of the stance and swing legs for 50 consecutive steps of (a) running with point feet, and (b) running with passive feet. The solid lines represent the mean recorded joint angle waveforms, while the dotted lines indicate the upper and lower quartiles over the running steps. The curves were temporally normalized from initial touchdown (0%) to subsequent touchdown (100%).

Fig. 13. Parameter plots for 50 consecutive steps for the outer-loop event-based controller, $\Gamma^P$, for (a) running with point feet and (b) running with passive feet; sc and sd refer to the values of the corresponding $\beta$-parameters in the stance-compression and stance-decompression subphases respectively.

1 m/s. All this is autonomously handled by the controller with no manual intervention. The ability of the controller to recover from slow speeds below 1 m/s, and high speeds above 2.5 m/s illustrates a good robustness to imperfections in the ground-contact model. The controller is also able to account for significant cable stretch (shown in Figure 15).

5.3. Experiment 3: One-step transition from running to walking

This section briefly describes the controller used to transition from running to walking. To realize such transitions, the running controller is switched to a walking controller...
Fig. 14. Parameter plots for 50 consecutive steps for the nonlinear outer-loop controller for increasing robustness to perturbations, $\Gamma^0$, for (a) running with point feet and (b) running with passive feet; $s_c$ and $s_d$ refer to the values of the corresponding $\gamma$-parameters in the stance-compression and stance-decompression subphases respectively.

Fig. 15. Absolute value of leg shape cable stretch and spring compression for the stance leg when running with point feet (Experiment 2). Both variables are scaled to be in the leg shape coordinates. As is seen, cable stretch contributes as much as the spring to the compliance present in the system. This was hinted at in Table 2.

that creates virtual compliance through active force control on the stance leg shape motor. This walking controller essentially treats a running-to-walking transition as a large step-down, similarly to what was done in Park et al. (2012) for walking gaits. Figure 10(b) illustrates plots of various variables for the transition from running to walking. The running and walking sections are clearly marked along with the transition step. Note that transition from running to walking is achieved in a single step.

5.4. Discussion of the experiments

The experimental implementation of running motions on MABEL revealed a number of interesting observations regarding the robot and the proposed controller. First, it was observed that the robot runs faster in experiments than what simulations predict based on the developed models. This behavior is similar to what was observed in walking experiments with MABEL (Sreenath et al., 2011b), and is attributed to the inevitable inaccuracies associated with the ground-contact model. While in Sreenath et al. (2011b, Section 7-B) we suggest various ways of modeling the ground impact, demonstrating that impact scaling can account for speed differences in walking, it is not clear how the parameters of the compliant ground model can be selected to improve the accuracy of the simulations in the case of running.

Another source of inaccuracy is the assumption of planar motion that underlies the model based on which the controller is derived. Clearly, the support boom in the experimental setup constrains the robot’s hip to move on the surface of a sphere and not in the sagittal plane. Furthermore, the boom affects the weight distribution so that the robot weighs 10% (approximately 7 kg) more when supported on the inner leg than when supported on the outer leg. In running, this asymmetry results in harder impacts on the inside leg causing its knee to bend more during the corresponding stance phase. As a consequence, the outer-loop component of the controller tends to overcompensate in the following step; notice the pronounced step-to-step oscillations in the virtual compliance in Figure 14(a). To account for this phenomenon, the controller can be modified so that the virtual compliance is 10% stiffer on the inside leg. Moreover, for smoother running motions, the outer-loop controllers can perform separate step-to-step updates over two steps.
As a final remark, note that the proposed controller combines formal control synthesis procedures with heuristics to experimentally realize running on MABEL. The inner-loop control components (namely, $\Gamma^{\alpha}$, $\Gamma^{\omega c}$, and $\Gamma^{\beta}$) are designed through systematic control methods to meet certain specifications, such as hybrid invariance and local exponential stability. In contrast, the outer-loop event-based controller $\Gamma^{r}$ is based on certain intuitive observations aiming to enhance the robustness of the controller to perturbations in the knee angle at impact and to imperfections in the ground-contact model. To minimize the reliance of the controller on heuristics, the softening effect of the spring for large knee angles can be incorporated in the continuoustime control component by suitably modifying the virtual compliance (14) to include the nonlinear relation between the knee bending angle and the developed leg force, as observed in Rummel and Seyfarth (2008). Similarly, the effect of cable stretch can also be included in (14). With these modifications, the outer-loop components $\Gamma^{r}$ could be removed from the design and $\Gamma^{\beta}$ would be sufficient to ensure both exponential stability and robustness.

6. Conclusion

MABEL contains springs in its drivetrain for the purposes of enhancing the agility and robustness of dynamic locomotion. This paper has presented a model-based control design method to realize the potential of the springs. Experiments have been performed to illustrate and confirm important aspects of the feedback design.

The controller is based on the HZD introduced in Poulaakakis and Grizzle (2009a) and further developed and deployed experimentally in Sreenath et al. (2011b). An important modification was the deliberate inclusion of actuation in the zero dynamics during the stance phase of running, which enabled active force control of the stance knee. Specifically, a virtual compliant element was created to dynamically vary the effective leg compliance during stance. An outer-loop event-based controller was designed to exponentially stabilize the periodic running gait. An additional outer-loop event-based controller was designed to improve the robustness of the periodic running gait to perturbations in the knee angle at impact and to imperfections in the ground-contact model.

The running controller has been experimentally deployed and stable running has been successfully demonstrated on MABEL, both with passive feet and with point feet. The achieved running is dynamic and life-like, exhibiting flight phases of significant duration and high ground clearance. For running with point feet, the developed controller resulted in a kneed-biped running record of 3.06 m/s (10.9 kph or 6.8 mph).

Notes

1. The pre-tension in the cables between the spring and the pulley $B_{\text{spring}}$ seen in Figure 2(b) has been set as close to zero as possible to ensure the spring is not pre-loaded.

2. During flight both feet are off the ground, however, we continue to use stance leg to mean the leg that was on the ground during the stance phase, and similarly for the swing leg.

3. Contrast this with that of humans, where the legs travel roughly 90% of the range of travel during the stance phase.

Funding

This work is supported in part by NSF grants ECS-909300 and CMMI-1130372, and in part by DARPA contract W91CRB-11-1-0002.

Acknowledgments

The authors are grateful to A. Ramezani for his expert help with the experiments. K. Sreenath acknowledges L. McCauley for suggesting the experiments with nontrivial feet, and B. Morris is thanked for his advice, ‘Go big, or go home’, which helped us to focus on multiple steps of running, and for his contributions to the theoretical underpinnings of our work, as cited in the text. J. Konscol is thanked for his weekly visits to the lab and for sharing his engineering experience, which contributed invaluably to the robustness of the electronic setup of the testbed. G. Buche is thanked for his many contributions to the design of the electronics, power supply, and safety interlock systems. Last but not least, we are deeply indebted to J. Hurst for designing MABEL. We hope that this paper has once again confirmed many of his expectations for the robot.

References


Westervelt ER, Grizzle JW and Canudas de Wit C (2003) Switching and PI control of walking motions of planar biped...
Appendix A: Running with passive feet

To avoid slipping, MABEL’s original shins terminating at point feet were replaced with shins terminating in passive feet enclosed in regular running shoes that provide a larger surface area for better traction. With the addition of the passive feet, the model developed in Section 2 is no longer valid due to (a) the different inertia properties between the two designs, and (b) the different toe–ground interactions which, in the case of shoes, result in softer impacts due to the rolling contact with the ground. To accommodate these differences, a number of modifications will be made in the controller.

Modifications to regulate speed: At faster speeds, the time spent in the sd phase decreases. As a result, the effective energy injection is reduced, resulting in a lower peak apex height in flight. To maintain a well-defined flight phase at fast speeds, an additional parameter, apex height in flight. To maintain a well-defined flight phase and prevent slipping towards the end of stance when the stance forces are small, (b) hyper-extension of the heavy shin, and (c) large vertical velocities at lift-off, the sd-to-flight switching surface is modified as below:

\[ S_{sc→sd}^{exp} := S_{sc→sd} \cap \{ x_s \in TQ_v | \beta_s > \beta_{s50\%}, q_{Bsp,at} < 20^\circ \} \]  

where \( \beta_{s50\%} \) is the value of \( \beta_s \) at 50\% into stance.

To prevent (a) slipping towards the end of stance when the stance forces are small, (b) hyper-extension of the heavy shin, and (c) large vertical velocities at lift-off, the sd-to-flight switching surface is modified as below:

\[ S_{sd→f}^{exp} := S_{sd→f} \cap \{ x_s \in TQ_v | q_{Bsp,at} < 15^\circ, q_{LS,at} < 2^\circ \} \cap \{ x_s \in TQ_v | \dot{p}_{hip}^v > \dot{p}_{hip}^{v→s} \} \]  

where \( \dot{p}_{hip}^v \) is the vertical hip velocity, and \( \dot{p}_{hip}^{v→s} \) is the nominal lift-off vertical hip velocity.

Finally, to prevent the shoes from scuffing the ground during leg swing, the swing leg shape virtual constraint is modified to fold the leg by an additional constant amount. Modifications to account for unmodeled cable stretch in leg angle transmission: The running controller accounts for unmodeled cable stretch in the leg shape coordinates, but not the leg angle coordinates. During the sd phase, the nominal virtual constraint specifies that the torso pitch backward. In experiments, the torso is sometimes driven forward to correct tracking errors, which results in forward pitching during flight causing a significant torso error on impact. To prevent this, the torso velocity drops below a threshold, the controller for the torso pushes the torso backward instead of enforcing the virtual constraint.

On initiation of the flight phase, the event-based controller, \( \Gamma^f \), ensures hybrid invariance through the correction polynomials, \( h^f \), as in (15), such that the modified virtual constraints smoothly join the nominal ones half-way into the flight phase. During experiments, large errors at lift-off may cause the modified virtual constraint to initially reverse the direction of motion, resulting in significant leg-angle cable stretch and subsequent large touchdown errors. To handle this, the correction polynomials are modified such that the modified virtual constraint smoothly joins the nominal one at an adaptively chosen location that is either
50%, 75%, or 95% into the flight phase, depending on the sign and magnitude of the error on transition to flight. With these modifications, the running experiments with passive feet can be performed. The walking controller of Sreenath et al. (2011b) is employed, along with a torso offset to lean the torso forward to induce stable walking with the passive feet at 1.26 m/s. The walking-to-running transition controller developed in section 5.1 is used to excite running. On transition, the modified controller described above is executed to sustain running at an average speed of 1.07 m/s obtaining 100 running steps. Figure 11(b) illustrates snapshots of a typical running step. The average stance and flight times are 360 ms and 151 ms, respectively (i.e. flight amounts for 30% of the gait). The ground clearance is approximately 2 inches (5 cm) and the specific cost of mechanical transport ($c_{mt}$) is 0.75. Figure 12(b) depicts the mean joint angles and motor torques, normalized over time, for 50 consecutive steps of running. Figures 13(b) and 14(b) illustrate the $\beta$- and $\gamma$-parameters.

Appendix B: Analyzing the stability of the $\Gamma^\gamma$ controller

To analyze stability, the Poincaré map is numerically computed. To ease computation, the section $\tilde{S}_\gamma = \{ x_s \in TQ_s \mid \theta_s = \theta_{77\%} \}$ will be considered instead of $S_\gamma$; $\tilde{S}_\gamma$ represents a switching surface 77% into the stance phase. We can then study the eigenvalues of the Poincaré map, $\tilde{P}_\gamma : \tilde{S}_\gamma \times B \times G \rightarrow \tilde{S}_\gamma$. Note that, while the Poincaré section $\tilde{S}_\gamma$ is used instead of $S_\gamma$, the $\beta$- and $\gamma$-parameters still continue to be updated on their respective switching surfaces, $S_\beta$ and $S_\gamma$. To define the Poincaré map, $\tilde{P}_\gamma$, we define three maps: $\tilde{P}_1^\gamma : \tilde{S}_\gamma \times B \times G \rightarrow S_\gamma$ which maps a state on $\tilde{S}_\gamma$ along with $\beta$ and $\gamma$ to the post-impact surface, which is also the switching surface $S_\gamma$ for the event-based controller $\Gamma^\gamma$; $\tilde{P}_2^\gamma : S_\gamma \times B \times G \rightarrow S_\beta$ which maps a state on $S_\gamma$ to the sc-to-sd transition surface, which is also the switching surface $S_\beta$ for the event-based controller $\Gamma^\beta$; and finally, $\tilde{P}_3^\gamma : S_\beta \times B \times G \rightarrow \tilde{S}_\gamma$, which maps a state on $S_\beta$ back to $\tilde{S}_\gamma$, the Poincaré section under consideration. To further clarify this, we define

$$x_p^\gamma[k] = \tilde{P}_1^\gamma(x_p[k], \beta[k], \gamma[k])$$

$$x_p^\beta[k] = \tilde{P}_2^\gamma(x_p^\gamma[k], \beta[k], \Gamma^\gamma(x_p^\gamma[k]))$$

Then,

$$x_p[k + 1] = \tilde{P}_\gamma(x_p[k], \beta[k], \gamma[k])$$

$$= \tilde{P}_\gamma(x_p^\gamma[k], \Gamma^\beta(x_p^\gamma[k]), \Gamma^\gamma(x_p^\gamma[k]))$$

Thus the $\gamma$-parameters continue to be updated on the switching surface $S_\gamma$, while the $\beta$-parameters are updated on the switching surface $S_\beta$. The switching section $\tilde{S}_\gamma$ serves only to define the Poincaré map $\tilde{P}_\gamma$. With this, the eigenvalues of the linearized Poincaré map were computed and a dominant eigenvalue of magnitude 0.6072 was obtained, indicating that the closed-loop system still remains exponentially stable.