

CoM Control for Underactuated 2D Hopping Robots with Series-Elastic Actuation via Higher Order Partial Feedback Linearization

Pat Terry and Katie Byl

Abstract—In this work we introduce a method for enforcing stance phase trajectories on the center of mass (CoM) for a series-elastic actuated 2D hopping robot with realistic actuator dynamics. There is an abundance of work regarding the implementation of highly simplified hopper models, with the hopes of extracting fundamental control ideas for running and hopping robots. However, real-world systems cannot be fully described by such simple models, as real actuators have their own dynamics including additional inertia and non-linear frictional losses. Therefore, an important step towards demonstrating high controllability and robustness to real-world, uneven terrain is in providing accurate higher-order models of real-world hopper dynamics. Due to the series-elastic actuation and compliance in the leg, a classical acceleration-based Partial Feedback Linearization (PFL) construction is impossible due to the spring force instantaneously determining the acceleration of the leg of the robot, therefore we present a solution that constructs PFL control laws about the 4th derivative, the jounce. Building on earlier work, we generate CoM trajectories by abstracting the system to a single point mass moving along a 4th order quadratic trajectory and solve for the effective force vector to indirectly enforce trajectories on the body angle acceleration and ensure stable body motions. By controlling the CoM directly, we provide solutions that account for the real impact dynamics of the system in order to accurately regulate stride length, and provide a framework that allows for future work in the construction of trajectories to achieve desired results such as stride switching, operation on rough terrain, and disturbance rejection.

I. INTRODUCTION

Hopping robots, aka hoppers, are useful systems for studying methods by which legged robots can navigate and select footholds on intermittent, rough terrain, and they also have obvious potential for both fast and energy efficient locomotion. Bridging the gap between highly idealized spring-loaded inverted pendulum (SLIP) models and higher-order, lossy, real-world dynamics of hopping robots remains an open question. This work provides control solutions for hopper models with real features such as underactuation, actuator dynamics, body inertia, and impact dynamics associated with unsprung mass. We address the problem of stride length regulation by controlling CoM coordinates directly using feedback control, however we also must ensure the body angle of the robot is well behaved, thus having only a hip actuator and a leg series elastic actuator results in a challenging underactuated control problem. This work is an extension of our previous work in [1], for which we are introducing several improvements. First, the system we

consider is a more realistic hopping robot model that utilizes series-elastic actuation and has actuator dynamics. Second, by using series elastic actuation we hope to achieve similar results but with more realistic actuator requirements. Third, we improve upon the PFL construction by changing the formulation such that the derivatives of the leg actuator u_{hip} can be analytically calculated. Lastly, we improve our trajectory generation by increasing the trajectory polynomial order to allow for initial accelerations more aligned to the real system's leg spring force.

Using compliant springs as energy storage devices for legged locomotion is a concept that has been extensively studied [2], and has led to the application of the spring-loaded inverted pendulum (SLIP) model, which is a simplistic model that is used to help describe the dynamics of many legged animals and legged robots and, most particularly, of hopping robots. Much work has been done in the area of legged locomotion utilizing the SLIP model [3], [4] due to its modeling simplicity. Many studies using SLIP models focus on use of leg angle at touchdown and/or modification of spring force during stance as the only available control actions [5], [6], [7]. Such works investigate control of step length and jumping height without regard for regulation of body attitude, modeling the body as a point mass with no distributed inertia. By contrast, our work here includes a physical body connected to the hip of the robot that must be stabilized. This increases both the range of possible dynamic motions and the complexity of the control strategy required.

Studying the behavior of a single leg and extending this to multiple legs has been shown to be effective [8] and has led to the development of many hopping and other leg-actuated robots, including well-known work led by Raibert [9] which produced many successful experimental results. In contrast to Raibert's work, our control strategies aim to minimize use of hand-tuned experimental parameters in an effort to increase robustness on unknown terrain environments. Operation of actual hardware ultimately requires leg placement on irregularly spaced footholds on potentially rough terrain [10], and for this to be achieved, accurate state tracking is desirable. Additionally, real world series elastic actuators have dynamics, therefore the assumption that length of the leg can be set either instantaneously or with constant acceleration for desired stance phase trajectories is not valid. Compression to a desired value during stance is clearly subject to important dynamic effects, often the actuator is still undergoing compression when the stance phase has ended. Additionally, simply including only linear (viscous) damping terms is not sufficient; non-linear Coulomb frictional effects

P. Terry and K. Byl are with the Robotics Lab in the ECE Dept, University of California, Santa Barbara, CA 93106 USA pwterry@umail.ucsb.edu, katiebyl@ece.ucsb.edu

must be estimated as well to have an accurate model.

To address control of similar, underactuated systems, much work has been done using the method of Partial Feedback Linearization (PFL), building on seminal work Spong [11]. Building from Spong’s work, we seek to directly set accelerations of a number of independent system degrees of freedom equal to the number of available actuators via feedback linearization, even if they are not directly actuated [12]. Spong’s famous Acrobot work was focused on PFL of a specific actuated or unactuated link physically on the system [13]; however, we are interested in utilizing PFL in order to control two of three auxiliary parameters, using our two available torque inputs. Those three parameters are: x_{cm} and y_{cm} , the coordinates of the center of mass, and the body angle of the robot, ϕ . Applying feedback linearization techniques to this system is a challenging problem not only because the system is underactuated, but because the compliant leg makes traditional acceleration-based PFL methods impossible due to the force being set instantaneously by the spring. It is however possible to extend the PFL technique by linearizing a higher order term, and thus we develop a control law for our hopper via this extension.

The rest of the paper is organized as follows. Section II introduces the specifics of the Hopper model used, Section III then constructs the required PFL control law to enforce CoM trajectories. The generation of the CoM trajectories using an equivalent point-mass model is explored in Section IV, and lastly Section V provides simulation examples for CoM trajectory regulation during stance.

II. HOPPER MODEL

The hopper model studied in this work is shown in Fig. 1 and closely reflects actual hardware being constructed in our lab, exhibiting real actuator dynamics, body and leg inertia, and both sprung and unsprung mass. The states X of the system are given as

$$X = [\theta, \phi, L, L_{act}, \dot{\theta}, \dot{\phi}, \dot{L}, \dot{L}_{act}]^T \quad (1)$$

where θ represents the leg angle, ϕ represents the body angle, L represents the leg length, and L_{act} represents the position of the series-elastic actuator. The system can be described as having a *flight phase* and a *stance phase*. As real world actuators have significant sources of both linear and nonlinear friction, we choose to model our system’s losses using both linear damping and non-linear Coulomb friction. During the flight phase the system simply follows ballistic dynamics, which are omitted here due to space limitations. The transitions between the stance phase and the flight phase, denoted *take-off* and *touch-down*, are discrete events at which additional energy in the system is lost due to the mass in the robot’s leg.

It can be shown using a Lagrangian approach that the

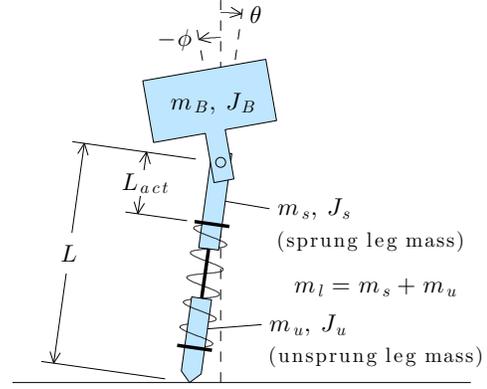


Fig. 1. Our hopper model consists of a compliant leg with both sprung and unsprung mass, connected to body which rotates freely around the hip joint. The system has two actuators, u_{hip} , which acts at the hip joint, and u_{leg} , which actuates the series-elastic actuator. The series elastic actuator position L_{act} is a state with real dynamics and represents the physical active spring compression achieved via u_{leg} , and L represents the leg length of the robot.

dynamics of the stance phase are

$$\begin{bmatrix} \ddot{\theta} \\ \ddot{\phi} \\ \ddot{L} \\ \ddot{L}_a \end{bmatrix} = M^{-1} \left(C + \begin{bmatrix} -NK_t u_{hip} - b_3 \dot{\theta} \\ NK_t u_{hip} \\ -b_k \dot{L} - f_2 \text{sgn}(\dot{L}) \\ -\nu u_{leg} - b_1 \dot{L}_a - f_1 \text{sgn}(\dot{L}_a) \end{bmatrix} \right) \quad (2)$$

where matrices M and C are both functions of X . The actuator variable u_{hip} outputs current to drive the actuators on the robot with motor torque constant K_t and gear ratio N . Similarly, actuator variable u_{leg} outputs current to compress the active portion of the spring via motor force constant ν . It is important to note that u_{leg} only appears instantaneously in the dynamics for L_{act} , only the current active spring position L_{act} has instantaneous effect on any leg or angle accelerations. The coefficients b and f represent damping and Coulomb frictional constants respectively.

Next we consider the variables we wish to regulate. These variables are the center of mass locations of the robot, given as

$$\begin{aligned} x_{cm} &= \frac{1}{m_l + m_B} (m_l x_{leg} + m_B x_{body}) \\ y_{cm} &= \frac{1}{m_l + m_B} (m_l y_{leg} + m_B y_{body}) \end{aligned} \quad (3)$$

where m_l represents the combined leg mass m_s and foot mass m_f , and x_{leg} , y_{leg} represent the position of the CoM of the leg. Similarly, x_{body} , y_{body} represent the position of the CoM of the body.

Since we aspire to generate trajectories for the CoM directly, we must account for the effect of the impact dynamics on the CoM. At *touch-down*, the foot of the robot impacts the ground causing the unsprung portion of the leg’s velocity to be dissipated. The instantaneous effect on the CoM velocity

L_0	0.534 m	Natural leg length
m_e	7.11 kg	Eff. actuator mass
m_l	0.5479 kg	Leg mass
m_f	0.95 kg	Foot mass
m_B	5.25 kg	Body mass
J_B	0.3 kg m ²	Body inertia
J_l	0.015 kg m ²	Total Leg inertia
k	2,389 N/m	Spring constant
c	0.021 m	Spring pre-load
ν	51.1 N/A	Actuator constant
b_3	1.1563 Ns/m	Act. lin. friction
b_k	3.333 Ns/m	Act. lin. friction
f_1	1.52 N	Act. coulomb friction
b_2	1.18 Ns/m	L lin. friction
N	66	Motor Gear ratio
K_t	0.0369 Nm/A	Torque Constant
f_2	1.03 N	L coulomb friction
u_{max}	20 A	Max allowed u

TABLE I
MODEL PARAMETERS USED FOR SIMULATION STUDY

can be written as

$$\begin{aligned}\dot{x}_{cm}^+ &= \dot{x}_{cm}^- - \frac{m_l}{2(m_s + m_l)} \dot{x}_{foot}^- \\ \dot{y}_{cm}^+ &= \dot{y}_{cm}^- - \frac{m_l}{2(m_s + m_l)} \dot{y}_{foot}^-.\end{aligned}\quad (4)$$

At *take-off*, the moving frame of the body impacts the leg assembly causing the system to lift off the ground. Similar to results seen with Raibert Hoppers [9], the radial change in velocity can be found by conservation of linear momentum and results in additional loss on the CoM velocity as

$$\begin{aligned}\dot{r}^+ &= \frac{m_s}{m_l + m_s} \dot{L}^- + \frac{m_l}{2(m_l + m_s)} L_0 \dot{\theta} \\ \dot{x}_{foot}^+ &= \dot{r}^+ \sin(\theta) \\ \dot{x}_{foot}^+ &= \dot{r}^+ \cos(\theta).\end{aligned}\quad (5)$$

In order to achieve simulation accuracy, the model parameters used in this study reflect preliminary system identification results of hardware currently under construction. The model parameters used in this study are shown in Table I.

III. PFL CONSTRUCTION

This section introduces our PFL control framework. Due to the system's series elastic actuation, constructing the acceleration of our control variables only yields one available input term. This can be verified by observing Equation 2 and noting the acceleration of L , θ , and ϕ are not instantaneously affected by u_{leg} . Therefore, when we take derivatives to define \ddot{x}_{cm} , \ddot{y}_{cm} and substitute Equation 2 for the state acceleration variables we obtain

$$\begin{aligned}\ddot{x}_{cm} &= \beta_x(X)u_{hip} + \epsilon_x(X) \\ \ddot{y}_{cm} &= \beta_y(X)u_{hip} + \epsilon_y(X),\end{aligned}\quad (6)$$

where β , and ϵ are functions of the state X . The fourth state variable L_{act} , the active spring compression, appears in the dynamics for the acceleration of the remaining three state variables. Therefore, two derivatives must be taken for u_{leg} to instantaneously affect the CoM, thus we consider constructing our input-output linearization of the y -component of the

center of mass with relative degree 4. To cope mathematically with the Coulombic friction terms in the model, we first approximate these terms via a scaled arctangent function as

$$W = M^{-1} \left(C + \begin{bmatrix} -NK_t u_{hip} - b_3 \dot{\theta} \\ NK_t u_{hip} \\ -b_k \dot{L} - \frac{2f_2}{p_i} \operatorname{atan}\left(\frac{\dot{L}}{\mu}\right) \\ -\nu u_{leg} - b_1 \dot{L}_a - \frac{2f_1}{p_i} \operatorname{atan}\left(\frac{\dot{L}_a}{\mu}\right) \end{bmatrix} \right) \quad (7)$$

which is a reasonable approximation for μ sufficiently small. We proceed by taking two derivatives of the state acceleration equations in 7 to obtain dynamics for the jounce of the system as

$$J(X, u_{leg}, u_{hip}, \dot{u}_{hip}, \ddot{u}_{hip}) \approx \frac{d}{dt} \frac{d}{dt} W. \quad (8)$$

Next, we take additional time derivatives of Equation 3 to define $\ddot{\ddot{y}}_{cm}$, and by substituting in Equations 2 and 8 we form our control variables as

$$\begin{aligned}\ddot{\ddot{y}}_{cm} &= \gamma_y u_{leg} + \beta_y u_{hip} + \eta_y \dot{u}_{hip} + \alpha_y \ddot{u}_{hip} + \epsilon_y \\ \ddot{\ddot{x}}_{cm} &= \beta_x u_{hip} + \epsilon_x\end{aligned}\quad (9)$$

where the equation for x_{cm} is unchanged and γ , η , and α are higher order terms that are also a function of only the state X .

We define the control law for x_{cm} , which is trivial as only one actuator has effect on the acceleration, as

$$u_{hip} = \frac{1}{\beta_x} (-\epsilon_x + v_x) \quad (10)$$

where v_x is calculated via PD feedback, to drive the system to desired references as

$$v_x = K_p(x_{ref} - x_{cm}) + K_d(\dot{x}_{ref} - \dot{x}_{cm}). \quad (11)$$

In order to define the control law for y_{cm} we must determine the two input derivative terms, which can be calculated as follows. We construct the first derivative of u_{hip} as

$$\dot{u}_{hip} = \frac{1}{\beta_x} (-\dot{\epsilon}_x + \dot{v}_x) - \frac{\dot{\beta}_x}{\beta_x^2} (-\epsilon_x + v_x). \quad (12)$$

Next we make use of the fact that

$$\dot{v}_x = K_p(\dot{x}_{ref} - \dot{x}_{cm}) + K_d(\ddot{x}_{ref} - \ddot{v}_x). \quad (13)$$

Since the coefficients $\dot{\epsilon}_x$ and $\dot{\beta}_x$ are functions of X and state accelerations excluding L_{act} , using Equation 2 with our control law in Equation 10 allows these terms to be calculated. Next, we construct the second derivative of u_{hip} in exactly the same manner, and similarly make use of the fact that

$$\ddot{v}_x = K_p(\ddot{x}_{ref} - \ddot{v}_x) + K_d(\dddot{x}_{ref} - \ddot{v}_x). \quad (14)$$

New coefficients $\ddot{\epsilon}_x$ and $\ddot{\beta}_x$ appear and are functions of X , state jerk excluding L_{act} , and state accelerations, which we can calculate using Equations 2 and 9 for the state dynamics with Equations 10 and 12 for the input terms. In this case the term \ddot{L}_{act} appears in $\ddot{\epsilon}_x$, therefore the component of \ddot{u}_{hip} due

to u_{leg} must be separated and used to generate new effective coefficients as

$$\begin{aligned}\ddot{u}_{hip} &= \delta_1 + \frac{\delta_2}{\beta_x} u_{leg} \\ \tilde{\gamma}_y &= \gamma_y + \alpha_y \frac{\delta_2}{\beta_x}.\end{aligned}\quad (15)$$

Finally, using equations 10, 12, and 15 we define our control law for y_{cm} as

$$u_{leg} = \frac{1}{\tilde{\gamma}_y} (-\epsilon_y - \beta_y u_{hip} - \eta_y \dot{u}_{hip} - \alpha_y \delta_1 + v_y) \quad (16)$$

where v_y is calculated via feedback, to drive the system to desired references as

$$\begin{aligned}v_y &= v_1 + v_2 \\ v_1 &= K_1(y_{ref} - y_{cm}) + K_2(\dot{y}_{ref} - \dot{y}_{cm}) \\ v_2 &= K_3(\ddot{y}_{ref} - \ddot{y}_{cm}) + K_4(\ddot{y}_{ref} - \ddot{y}_{cm}).\end{aligned}\quad (17)$$

Since the relative order of the feedback linearization is four, we require four total poles in the closed-loop dynamics of y_{cm} . One option for setting the controller gains is to first select a dominant pole-pair with natural frequency ω_n and damping ratio ζ , and then set a significantly faster decay rate for the two remaining, real-valued poles, z_3 and z_4 . For a chosen set of w_n, ζ, z_3, z_4 , the controller gains are:

$$\begin{aligned}K_1 &= z_3 z_4 w_n^2 \\ K_2 &= (z_3 + z_4) w_n^2 + 2z_3 z_4 \zeta w_n \\ K_3 &= w_n^2 + 2\zeta(z_3 + z_4) w_n + z_3 z_4 \\ K_4 &= z_3 + z_4 + 2\zeta w_n.\end{aligned}\quad (18)$$

Additionally, we require both references and estimates of the acceleration and jerk for y_{cm} , however since u_{hip} and its derivative are known the acceleration and jerk can be analytically calculated. Trajectory generation used to generate the required references is explored in the next section

IV. CoM TRAJECTORY GENERATION

A. Equivalent Point-Mass Model Dynamics

Our trajectory generation builds on similar work in [1], in which we generate trajectories for the CoM and body attitude with the ground reaction force (GRF) pointing towards the CoM, allowing direct control of the position of the robot while keeping the body angle ϕ well behaved. This is accomplished since the system in the prior work has only a single rigid body, and pointing the GRF at its CoM imparts no net torques on this body, resulting in a controlled system with dynamics quite similar to the classic point-mass SLIP model. In order to use such a strategy for the hopper system in the present work, we must augment the point-mass model used in our previous work with an additional parameter, $\delta(t)$, shown in Figure 2, which represents an angular offset of the force vector that provides deliberate torques to the bodies. The dynamics for the reduced system during the stance phase

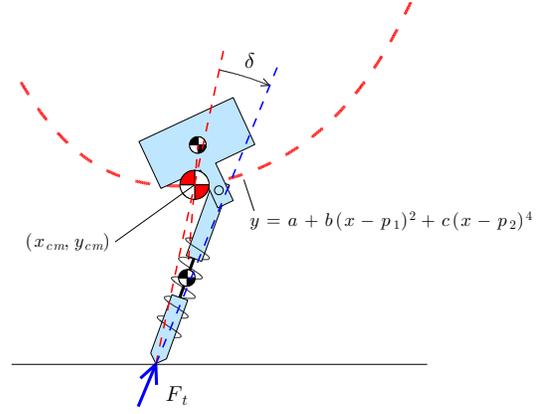


Fig. 2. CoM overlaid with the full model during stance. In our planned trajectories, the ground reaction force is intentionally constrained to point exactly toward a point offset from the center of mass of the system by angle $\delta(t)$. This offset angle can be used to enforce trajectories precisely on the body angle acceleration during the stance phase, while maintaining desired terminating CoM trajectory values.

can be written as

$$\begin{aligned}\ddot{x}_{cm} &= \frac{1}{m} F_t \sin(\tan^{-1}(\frac{x_{cm}}{y_{cm}} + \delta(t))) = \frac{1}{m} F_t s_1 \\ \ddot{y}_{cm} &= \frac{1}{m} F_t \cos(\tan^{-1}(\frac{x_{cm}}{y_{cm}} + \delta(t))) - g = \frac{1}{m} F_t c_1 - g \\ \ddot{\phi}_{st} &= f_{st}(\phi, \dot{\phi}, L, \dot{L}, \theta, \dot{\theta}, L_{act}, u_{hip})\end{aligned}\quad (19)$$

where the function f_{st} is the equation for the non-linear dynamics of the body angle from Equation 2.

Next we define y_{cm} as a function of x_{cm} . We desire trajectories that are symmetric in y_{cm} , and asymmetric in \dot{x}_{cm} and \dot{y}_{cm} , which allows for correction of impact energy losses at take-off and touch-down, and also allows for trajectory switching into different gaits with different desired stride lengths. We also desire the ability to set not only the initial positions and velocities of x_{cm} and y_{cm} , but also the initial acceleration of y_{cm} to align with the expected initial spring force. The curve we choose to enforce is

$$y_{cm}(x_{cm}) = a + b(x_{cm} - p_1)^2 + c(x_{cm} - p_2)^4 \quad (20)$$

We next write the acceleration of y_{cm} as:

$$\ddot{y}_{cm} = y'' \dot{x}_{cm}^2 + y' \ddot{x}_{cm} \quad (21)$$

where the derivatives $y'' = \frac{d^2 y_{cm}}{dx_{cm}^2}$ and $y' = \frac{dy_{cm}}{dx_{cm}}$ can be computed analytically since we have defined $y_{cm}(x_{cm})$ in Eq. 20. Next, we combine Eq. 19 with Eq. 21 to solve for F_t as:

$$F_t = \frac{1}{c_1 - y' s_1} (m y'' - g), \quad (22)$$

and by substituting Eq. 22 into Eq. 19 we generate stance phase trajectories by selecting trajectory coefficients and solving the differential equation in an ODE solver. The coefficients a, b , and c are determined by initial conditions. Two equations are trivially generated by considering Equation 20 evaluated at initial conditions y_0 and x_0 , and taking one

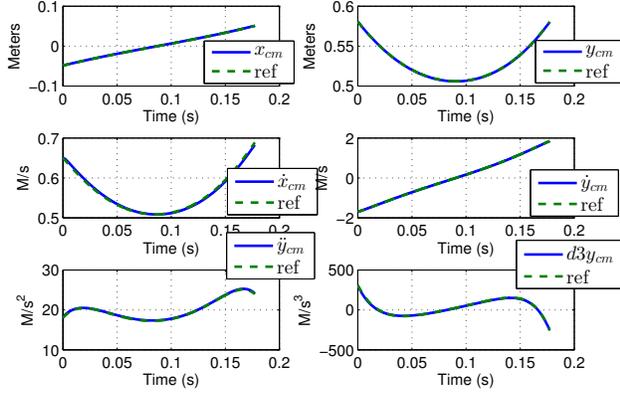


Fig. 3. The Center of Mass variables are controlled directly during the stance phase via PFL. The velocity curves are intentionally asymmetric to precisely account for energy loss from impact dynamics. Simulation results show that good performance can be achieved during the stance phase.

derivative and evaluating with initial conditions \dot{y}_0 and \dot{x}_0 . In order to align the initial acceleration of y_{cm} to a desired value we must construct an additional equation by taking two derivatives of Equation 20 as

$$\ddot{y} = (2b + 12c(p_2 - x)^2)\dot{x}^2 - \ddot{x}(b(2p_1 - 2x) + 4c(p_2 - x)^3). \quad (23)$$

For the above trajectory to be valid it must be equal to the point-mass model's acceleration in Equation 19. Equating Equations 23 and 19 allows us to generate the last needed equation. Solving for the coefficients and substituting in their solution for \ddot{y}_{cm} results in the condition

$$\ddot{x}_{cm} = \frac{\sin(\delta(t) + \tan^{-1}(\frac{x_{cm}}{y_{cm}}))(\dot{y}_{cm} + g)}{\cos(\delta(t) + \tan^{-1}(\frac{x_{cm}}{y_{cm}}))} \quad (24)$$

and since \ddot{x}_{cm} can be set instantaneously via our control, $\ddot{y}_{cm}(0)$ becomes a design parameter. We ensure the initial values of our trajectories for \dot{y}_{cm} and \dot{x}_{cm} align with the real system by setting L_{act} and \dot{L}_{act} to the appropriate touch-down values, which is easily accomplished by controlling them during the previous flight phase.

B. Body Angle Dynamics

For our hopping robot, simply pointing the force vector F_t exactly at the CoM, i.e. $\delta(t) = 0$, will result in a small amount of injected angular momentum at the end of the stance phase. This is contrasted with our work in [1], where such strategy was applicable. Additionally, during the next flight phase re-positioning the leg will inevitably incur additional angular rotation of the body. We therefore consider using $\delta(t)$ to enforce desired conditions on the acceleration of the body.

We begin by re-writing f_s from our equivalent model's stance phase dynamics in Equation 19 in terms of our PFL control variables x_{cm} and y_{cm} . First we make note that we can eliminate $u(t)$ as a variable as

$$u_{hip,stance} = \frac{1}{\beta_x}(-\epsilon_x + \ddot{x}_{cm}) \quad (25)$$

where we have used Equations 10 and 11 and the assumption that the PFL controller is functioning. We use Equation 3 and its first derivatives to generate three equations for the state variables L , \dot{L} , θ , and $\dot{\theta}$, and generate the last required equation by using the second derivatives of Equation 3 together with the system dynamics in Equation 2, which is required as L_{act} only appears in second derivative terms of the position states. Thus, we re-write the body angle acceleration using a change of variables as

$$\ddot{\phi}_{st} = f_{st}(\phi, \dot{\phi}, x_{cm}, y_{cm}, \dot{x}_{cm}, \dot{y}_{cm}, \delta(t)) \quad (26)$$

Here we note that at every time t_i , we can find the required $\delta(t_i)$ to achieve a desired angular acceleration $\ddot{\phi}(t_i)$. For the work presented in this paper the solution was found by using a simple binary search algorithm while solving the differential equation in 19. Thus, if the initial body angle ϕ_0 and velocity $\dot{\phi}_0$ are known such a method can be used to enforce trajectories on $\ddot{\phi}$ that result in desired take-off values ϕ_{TO} and $\dot{\phi}_{TO}$.

We next consider the body dynamics for the subsequent flight phase. The dynamics of ϕ during flight are a function of only ϕ , $\dot{\phi}$, θ , $\dot{\theta}$ and u_{hip} . During the flight phase we use a simple PFL controller to re-position the leg to the next touch-down angle associated with our stance phase initial conditions. Thus, we can determine the future actuator input as

$$u_{hip,flight} = \frac{1}{\beta_\theta}(-\epsilon_\theta + \ddot{\theta}) \quad (27)$$

where the PFL coefficients β_θ , ϵ_θ are constructed from the flight phase dynamics and we have used the assumption that the PFL controller is functioning. The flight phase trajectory of the leg angle, $\theta_{fl}(t)$, is a design choice with the only requirement being that terminating value is the next touch-down angle, given by

$$\theta_{TD} = -\cos^{-1}\left(\frac{y_0(m + m_l) - l_2 m \cos(\phi)}{L_0(m + \frac{m_l}{2})}\right) \quad (28)$$

Using this trajectory, we can then write the flight phase dynamics for ϕ as

$$\ddot{\phi}_{fl} = f_{fl}(\phi, \dot{\phi}, \theta_{fl}, \dot{\theta}_{fl}, \ddot{\theta}_{fl}). \quad (29)$$

For a given designed ϕ_{TO} , $\dot{\phi}_{TO}$, and $\theta_{fl}(t)$, we use Equation 29 to calculate the flight phase trajectory of the body and determine the subsequent touch-down values ϕ_{TD} and $\dot{\phi}_{TD}$

C. Design Task

The first design decision is the selection of a desired stride length S_{step} and CoM initial conditions x_0 , y_0 , \dot{x}_0 , and \dot{y}_0 . The stride length is given as

$$S_{step} = x_{TO} - x_0 + \frac{2\dot{x}_{TO}^+ \dot{y}_{TO}^+}{g}. \quad (30)$$

The initial conditions θ_0 , $\dot{\theta}_0$, and ϕ_0 are uniquely determined, and the initial body angular rate $\dot{\phi}_0$ is a design parameter. For a system with no impact dynamics, the terminating trajectory velocity values $\dot{x}(N)$ and $\dot{y}(N)$ could simply be \dot{x}_0 and $-\dot{y}_0$, resulting in a symmetric trajectory, however this is not the

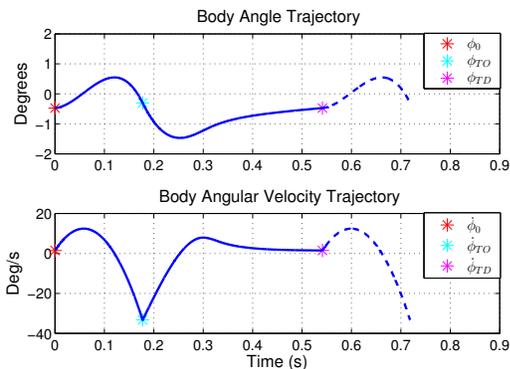


Fig. 4. The body angle is carefully controlled such that the touch-down values are equal to the initial conditions from the previous step.

case for real systems. The stride length is not determined by the terminating velocities of the trajectory $\dot{x}(N)$ and $\dot{y}(N)$, but rather the velocities after the take-off dynamics occur. By using the impact dynamics in Equations 5 and 4 with Equation 3 and its derivatives we can analytically compute the required terminating trajectory values to enforce the condition $\dot{x}_{TD}^+ = \dot{x}_0$ and $\dot{y}_{TD}^+ = \dot{y}_0$ as

$$\begin{aligned} \dot{x}_{TO}^- &= f_1(x_0, y_0, \dot{x}_0, \dot{y}_0, \phi_0, \dot{\phi}_0, \phi_{TO}, \dot{\phi}_{TO}) \\ \dot{y}_{TO}^- &= f_2(x_0, y_0, \dot{x}_0, \dot{y}_0, \phi_0, \dot{\phi}_0, \phi_{TO}, \dot{\phi}_{TO}). \end{aligned} \quad (31)$$

We use the asymmetry coefficients p_1 and p_2 to add asymmetry such that the terminating trajectory values are \dot{x}_{TO}^- and \dot{y}_{TO}^- . Note that we can also easily construct trajectories to switch from one set of CoM trajectories to the other by simply finding the proper p_1 and p_2 . Therefore, the design task simply becomes selecting \dot{x}_0 , and \dot{y}_0 to achieve a desired stride length and finding the appropriate $p_1, p_2, \dot{\phi}_0, \phi_{TO}$, and $\dot{\phi}_{TO}$ such that

$$\begin{aligned} \dot{x}(N) &= \dot{x}_{TO}^- \\ \dot{y}(N) &= \dot{y}_{TO}^- \\ \dot{\phi}_{TD} &= \dot{\phi}_0 \\ \phi_{TD} &= \phi_0. \end{aligned} \quad (32)$$

The trajectories are generated offline by solving the differential equation presented in Equations 19 and 22 for a set of parameters, and enforced in real time using the PFL laws. An example of such a trajectory for the body angle is shown in Fig. 4, where we design the body angle to remain relatively small.

V. SIMULATION RESULTS

A simulation study was conducted to provide simple examples of the control methods presented in this paper. The trajectories presented here are for a stride length of 35 cm. Table II lists the model and design values used in this study. The controller is able to successfully enforce the designed stance phase trajectories, and the body angle initial and final values are correctly aligned, by design. The actuator output for a single step is shown in Figure 5, which illustrates

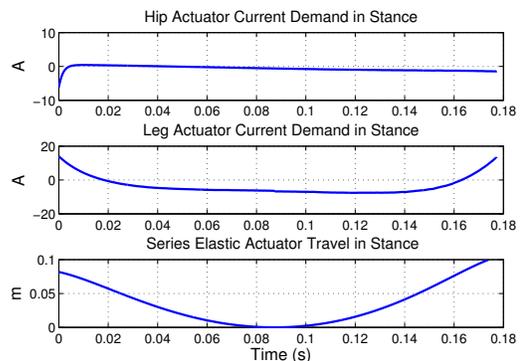


Fig. 5. Both the hip angle actuator and the hip series elastic actuator are active to achieve CoM tracking. The physical position of the series elastic actuator is shown to note it stays within the physical limits of our robot.

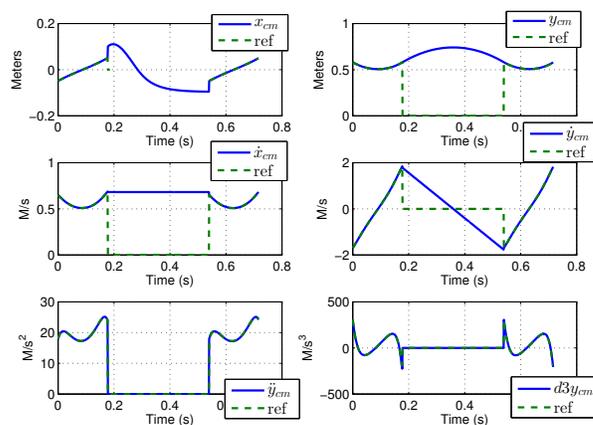


Fig. 6. Simulation results over 2 steps enforcing CoM trajectories. The initial conditions at the second stance phase are aligned with the trajectory, by design.

that the controller's output is both reasonable and within our hardware limits.

VI. CONCLUSION

We have presented a control framework for an under-actuated hopping robot model, with real actuator dynamics that are driven by series elastic actuation, that allows for direct CoM trajectory planning using high order Partial Feedback Linearization techniques. By constructing an equivalent point-mass model we provide a method that allows for direct control of the CoM such that accurate stride regulation and switching is possible, while also maintaining body stability. We provide a method that allows for the construction of trajectories with arbitrary initial conditions while maintaining desired take-off velocities.

There are specifics of the system we do not model in this study, and leave open for future work, including trajectory optimization for energy efficiency. Real world problems such as sensor accuracy are also left for future work. Future work includes constructing a database of trajectories with varying initial conditions to account for problems such as

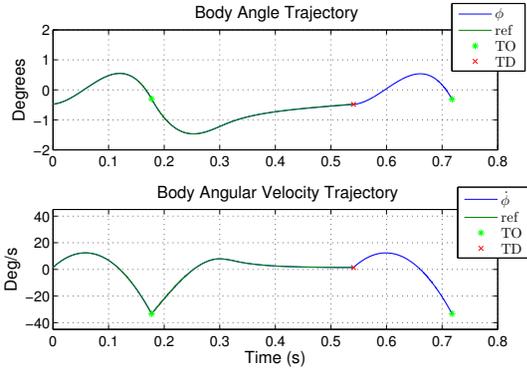


Fig. 7. Simulation results over 2 steps for the body angle. As expected, the body angle is carefully controlled and maintains well behaved during the stance and flight phase, and is correctly aligned with the subsequent stance phase initial condition.

$x(0), \dot{x}(0)$	-0.05 m, 2.2 m/s	x_{cm} IC
$y(0), \dot{y}(0)$	0.5805 m, -1.7 m/s	y_{cm} IC
$\phi(0), \dot{\phi}(0)$	-0.4753 deg, 1.47 deg/s	ϕ IC
$\phi(TO), \dot{\phi}(TO)$	-0.3 deg, -33 deg/s	ϕ design
$\dot{y}(N), \dot{x}(N)$	0.6891 m/s, 1.8516 m/s	velocity design
p_1, p_2	1.4E-3, 2.5E-3	asymmetry constants
$\omega_n, \zeta, z_3, z_4$	20, 0.9, 100, 120	y_{cm} PFL params
K_p, K_d	900, 54	x_{cm} PFL params

TABLE II

CONTROL PARAMETERS USED FOR SIMULATION STUDY

uneven terrain and long term angular drift. Operation on rough terrain results in unknown disturbances to the initial position and velocities of the robot, thus having a method for generating alternate trajectories for varying initial conditions is an extension planned for this work. Similarly, generating a database of such trajectories can also be used to account for other real world problems such as long term angular drift.

ACKNOWLEDGMENTS

This work was supported by the Army Robotics CTA (Grant No. W911NF-08-R-0012). This work was supported by the Institute for Collaborative Biotechnologies through grant W911NF-09-0001 from the U.S. Army Research Office. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

REFERENCES

- [1] P. Terry and K. Byl, "A higher order partial feedback linearization based method for controlling an underactuated hopping robot with a compliant leg," in *Proc. IEEE Conf. on Decision and Control*, 2014.
- [2] R. M. Alexander, "Three uses for springs in legged locomotion," *Int. Journal of Robotics Research*, vol. 9, no. 2, pp. 53–61, 1990.
- [3] J. Schmitt and J. Clark, "Modeling posture-dependent leg actuation in sagittal plane locomotion," *Bioinspiration and Biomimetics*, vol. 4, no. 1, pp. 1–17, 2009.
- [4] Ö. Arslan and U. Saranlı, "Reactive planning and control of planar spring-mass running on rough terrain," *IEEE Trans. on Robotics*, vol. 28, no. 3, pp. 567–579, 2012.
- [5] G. Piovan and K. Byl, "Enforced symmetry of the stance phase for the spring-loaded inverted pendulum," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 1908–1914, 2012.

- [6] R. Blickhan and R. J. Full, "Similarity in multilegged locomotion: Bouncing like a monopode," *Journal of Comparative Physiology A: Neurothology, Sensory, Neural, and Behavioral Physiology*, vol. 173, pp. 509–517, Nov. 1993.
- [7] J. Seipel and P. Holmes, "A simple model for clock-actuated legged locomotion," *Regular and Chaotic Dynamics*, vol. 12, no. 5, pp. 502–520, 2007.
- [8] M. H. Raibert, M. Chepponis, and H. Brown, "Running on four legs as though they were one," *IEEE Journal of Robotics and Automation*, vol. 2, no. 2, pp. 70–82, 1986.
- [9] M. H. Raibert, *Legged Robots that Balance*. Cambridge, Massachusetts: MIT Press Series in Artificial Intelligence, 1986.
- [10] J. Hodgins and M. Raibert, "Adjusting step length for rough terrain locomotion," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 1, pp. 289–298, 1991.
- [11] M. W. Spong, "Underactuated mechanical systems," in *Control Problems in Robotics and Automation*, pp. 135–150, Springer, 1998.
- [12] M. W. Spong, "Partial feedback linearization of underactuated mechanical systems," in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, pp. 314–321, 1994.
- [13] M. W. Spong, "Swing up control of the acrobot using partial feedback linearization," in *Proc. American Control Conference (ACC)*, pp. 2158–2162, 1994.