A Higher Order Partial Feedback Linearization Based Method for Controlling an Underactuated Hopping Robot with a Compliant Leg

Pat Terry and Katie Byl

***Abstract*— This paper considers control techniques for a planar model of an underactuated hopping robot. Our focus is on tracking trajectories for center of body and body angle, given a single point of support on the ground, with the goals of having stable and robust gaits and of switching among reference trajectories to achieve variable stride lengths on potentially rough terrain. Having a control framework capable of generating and tracking feasible body movements for an underactuated compliant hopper in the presence of ground collisions and uneven terrain is a challenging problem. In this work we provide a robust solution by extending partial feedback linearization (PFL) techniques. We investigate the problem of having a passive spring element in the leg and attempting partial feedback linearization based control, and show that while this cannot be achieved for instantaneous control of the CoM accelerations, it is possible to linearize the instantaneous CoM jerk and develop suitable control laws.**

I. Introduction

The potential benefits of legged locomotion are many-fold and have been an area of active study for many years. In particular, legged robots can operate where only intermittent footholds are available, allowing them to traverse rough terrain, to operate in areas where wheeled vehicles can not. Studying the behavior of a single leg and extending this to multiple legs has been shown to be effective [1], [2] and has led to the development and study of dynamic, spring-legged robots. In this work, we employ a similar approach in an effort to better understand legged locomotion and develop robust, stable, and accurate control.

Analyses of legged locomotion often employ planar, multi-linked models with some combination of actuated joints, unactuated joints, and compliant components such as springs, toward capturing only the most essential dynamics necessary to develop fundamental principles of control. In this work we focus on control strategies for two multi-link hopping robot models, with the goal of developing robust control strategies that may be extended to more complex multi-legged robots. Thus, it is desirable for our models to be general enough that our control strategies can be applied to many legged robotic systems. In particular, we are interested in systems with both a stance phase and ballistic phase, in which total body inertia is configuration-dependent, where regulation of both center of body location and body attitude are essential, and where stance contact with the ground occurs at a point (i.e., no ankle torque can be generated). In addition, we wish to consider the challenge of including a passive spring element, which can potentially improve energy efficience but also complicates control design, as discussed later.

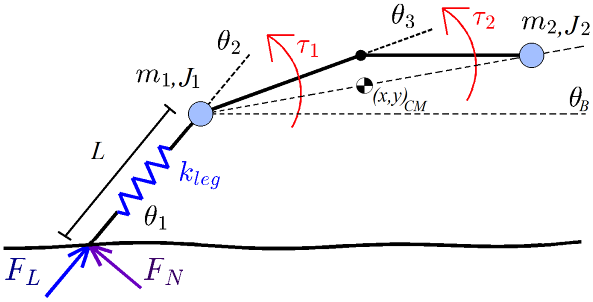

Fig. 1. The stance phase dynamics are modeled as a planar kinematic chain, with an optional spring (or spring-damper) element in the first link. The system is underactuated, with no torque actuation between the first link and the ground. Our objective is to use τ_1 and τ_2 to regulate both components of the ground contact force (F_L and F_N) independently, to in turn control the center of mass (x_{cm} and y_{cm}) and body angle (θ_B) via PFL. However, because F_L is instantaneously set by the spring force, we must either exclude the spring or extend the PFL method to a higher order to implement this approach.

Our essential model consists of three serial links. The first link is modeled as massless and may (optionally) contain a passive spring-damper element, as depicted in Fig. 1, while the other two links have mass and inertia. Two actuators apply torques at the hip (θ_2) and spine (θ_3) joints. For our model, the three control variables of interest are the center of mass positions (x_{cm} and y_{cm}) and body angle (θ_B) of the robot, defined as:

$$\begin{aligned} x_{cm} &= \frac{1}{m_1+m_2}(m_1x_1+m_2x_2) \\ y_{cm} &= \frac{1}{m_1+m_2}(m_1y_1+m_2y_2) \end{aligned} \tag{1}$$

$$\theta_B = tan^{-1}(\frac{y_2-y_1}{x_2-x_1}). \tag{2}$$

The ability to directly track trajectories for the CoM is very desirable, as this simplifies high-level trajectory planning to achieve different stride lengths, and to compensate for uneven ground, and maintaining attitude control of θ_B is essential for stability. Many studies using spring-loaded inverted pendulum (SLIP) models, or “hoppers’ focus on use of leg angle at touchdown and/or modification of spring force during stance as the only available control actions [3], [4], [5], [6]. 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.

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

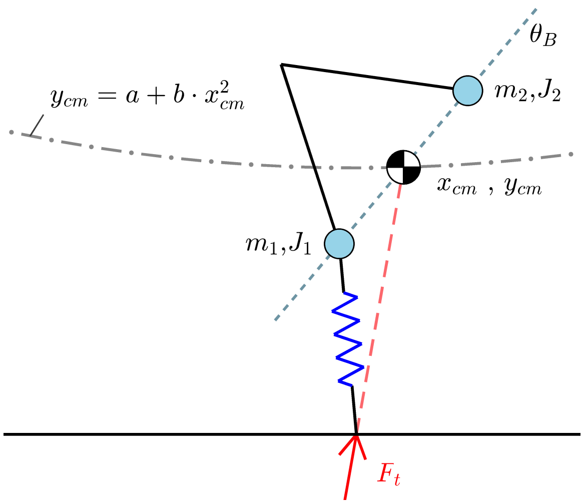

Fig. 2. CoM overlayed with the full model during stance. In our planned trajectories, the ground reaction force is intentionally constrained to point exactly toward the center of mass of the system. If the system begins with zero initial angular velocity, as desired, then this GRF constraint automatically maintains zero angular velocity throughout stance, by design.

By contrast, our work here includes an actuated hip and spine that affect both total system inertia and overall body angle. This increases both the range of possible dynamic motions and the complexity of the control strategy required.

Although for this study we are particularly interested in hopping gaits, one can add a fourth, massless link to the serial chain to produce a bounding model with a flexible or articulated spine, which is an active area of study [7], [8], [9], [10]. The control framework we present, while directly applicable to underactuated hopping robots, could correspondingly also be extended to a two-part bounding gait, in which one-legged support phases alternate between rear and front pairs of legs.

The reference trajectories for CoM positions and velocities presented in this paper are designed such that the corresponding ground reaction force (GRF) vector will point directly through the center of mass, as illustrated in Fig. 2, thus inducing no net torque on the body inertia. We focus on use of active control to switch between such reference trajectories and to reject impulsive perturbations that would be typical in rough terrain locomotion. Specifically, we focus on a problem statement capturing the following general characteristics and challenges: (1) a single, underactuated point contact sets the GRF (2) (at least) two actuators are available to affect the two (x and y) planar components of GRF magnitude, (3) body inertia changes with configuration to further complicate system dynamics, and (4) the system must recover rapidly from any perturbations at every step.

To address control of similar, underactuated systems, much work has been done using the method of Partial Feedback Linearization (PFL), building on seminal work by Spong [11]. Our compliant, multi-link system shares similarities with both hopper and biped models. PFL of multi-link robotic systems has been extended to study biped locomotion problems such as the Compass Gait walker [12]. 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 [13]. Spong's famous Acrobot work was focused on PFL of a specific actuated or unactuated link physically on the system [14]; 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, θ_B.

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 compliant system via this extension.

We also consider an alternate model in which we remove the compliant leg altogether so we can directly apply acceleration-based PFL, and show that results for the two models are nearly identical, neglecting energy cost differences. Excluding a spring from the leg of the robot increases the impulsive disturbance effect of and energy loss due to ground collisions. For real-world control, aspects of both compliant and non-compliant models are relevant, as hardware implementations of real robots have potentially significant losses from the ground impact, one common example being the unsprung portion of the leg of a hopper [15]. Human legs exhibit similar behavior in that they are not perfectly lossless springs, which has been observed in actual energy recovery experiments [16], with some studies suggesting gait selection and transition speed has a significant impact on energy recovery [17]. Our control provides a robust way of maintaining stability despite large linear and angular velocity instantaneous perturbations caused by ground collisions, and is general enough to work for both the compliant model presented in Fig. 1 and its non-compliant counterpart.

The remainder of this paper is organized as follows. Section II describes our extended PFL control law derivation, and provides control laws for both compliant and non-compliant versions of the model. Section III discusses our specific control implementation, developing CoM reference trajectories and dividing the stance phase into two separate PFL control phases in order to improve stability on rough terrain, and in the presence of possible ground collision losses. Lastly, Section IV provides simulation results for implementation of our control strategies on rough terrain, for both compliant and non-compliant versions of the model.

II. Modeling and PFL Construction

Using the Lagrangian method, we derive the nonlinear equations of motion in terms of the state vector, X:

$$X = [\theta_1, \theta_2, \theta_3, L, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{L}]^T \tag{3}$$

The resulting equations for link accelerations can be written as:

$$W(X,\tau_1,\tau_2) = \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{L} \end{bmatrix} = M^{-1}\left(C + \begin{bmatrix} 0 \\ \tau_1 \\ \tau_2 \\ -b_k\dot{L} \end{bmatrix} \right) \tag{4}$$

where matrices M and C are both functions of X. The CoM (x_{cm},y_{cm}) coordinates and body angle (θ_B) of the system are the three control variables we are interested in, Using PFL, we can control at most two of these three, because we have only two actuators. We start by taking derivatives to define $\ddot{x}_{cm}$, $\ddot{y}_{cm}$ and $\ddot{\theta}_B$, and by substituting Eq. 4 for the state acceleration variables, we rewrite the acceleration of our control variables as follows:

$$\begin{aligned} \ddot{x}_{cm} &= \beta_x(X)\tau_2 + \gamma_x(X)\tau_1 + \epsilon_x(X) \\ \ddot{y}_{cm} &= \beta_y(X)\tau_2 + \gamma_y(X)\tau_1 + \epsilon_y(X) \\ \ddot{\theta}_B &= \beta_\theta(X)\tau_2 + \gamma_\theta(X)\tau_1 + \epsilon_\theta(X), \end{aligned} \tag{5}$$

where β, γ, and ϵ are functions of the state X. At this point, it is important to note that use of a standard, acceleration-based PFL the construction will fail if the system has a compliant leg as shown in Fig. 1. This is because along the direction of the leg the radial force is always set instantaneously by the spring as $F_L = -k_{leg}\Delta L - b_k\dot{L}$, so that we can only instantaneously set one acceleration degree of freedom. This manifests in β evaluating to zero for all X. One potential way around this problem is to simply study the case when the spring is completely removed, and therefore L remains constant and is not required as a state. For that case, we can now define our feedback control law to track references for two desired acceleration variables. For example, if we wish to track x_{cm} and y_{cm}, then:

$$\tau_2 = \frac{1}{\beta_y - \beta_x\frac{\gamma_y}{\gamma_x}}\left(\epsilon_x\frac{\gamma_y}{\gamma_x} - \epsilon_y - v_1\frac{\gamma_y}{\gamma_x} + v_2\right) \tag{6}$$

$$\tau_1 = \frac{1}{\gamma_x}(-\epsilon_x - \beta_x\tau_2 + v_1), \tag{7}$$

where v_1 and v_2 are calculated via PD feedback, to drive the system to desired references:

$$\begin{aligned} v_1 &= K_p(x_{ref} - x_{cm}) + K_d(\dot{x}_{ref} - \dot{x}_{cm}) \\ v_2 &= K_p(y_{ref} - y_{cm}) + K_d(\dot{y}_{ref} - \dot{y}_{cm}). \end{aligned} \tag{8}$$

Another solution is to extend PFL to a higher order derivative. Then instead of linearizing the equations for acceleration of the control variables we linearize the jerk and develop a similar control law. We start by defining the jerk EoM by taking the time derivative of Eq. 4, noting that the additional time dependent variables τ_1 and τ_2 must be included.

$$J(X,\tau_1,\tau_2,\dot{\tau}_1,\dot{\tau}_2) = \frac{d}{dt}W(X,\tau_1,\tau_2) \tag{9}$$

Next, we take additional time derivatives of Eq. 1 to define $\dddot{x}_{cm}$, $\dddot{y}_{cm}$ and $\dddot{\theta}_B$, and by substituting in Eq. 4 and Eq. 9 we rewrite the jerk control variable equations in a form similar to that of the acceleration equations seen in Eq. 5:

$$\begin{aligned} \dddot{x}_{cm} &= \beta'_x(X)\tau_2 + \gamma'_x(X)\tau_1 + \eta_x(X)\dot{\tau}_1 + \epsilon'_x(X) \\ \dddot{y}_{cm} &= \beta'_y(X)\tau_2 + \gamma'_y(X)\tau_1 + \eta_y(X)\dot{\tau}_1 + \epsilon'_y(X) \\ \dddot{\theta}_B &= \beta'_\theta(X)\tau_2 + \gamma'_\theta(X)\tau_1 + \eta_\theta(X)\dot{\tau}_1 + \epsilon'_\theta(X), \end{aligned} \tag{10}$$

Since the spring inclusion results in $\beta = 0$ in Eq. 5 for all X, there is no coefficient associated with $\dot{\tau}_2$ in the jerk derivation, however in the general case of this derivation, for example if utilized with the model when a spring is not included, an additional coefficient will appear.

III. Control Implementation

To implement Eqn 10, we approximate the derivative of the torque input coefficients by using a first order difference approximation, denoting a new variable $\tau_1[t{-}T]$ as the stored previous controller output value, at some controller sampling time T. We can then group the coefficients together to create new effective coefficients as:

$$\begin{aligned} \dot{\tau}_1 &\approx \frac{\tau_1 - \tau_1[t-T]}{T} \\ \tilde{\epsilon} &= \epsilon' - \frac{\eta\tau_1[t-T]}{T} \\ \tilde{\gamma} &= \gamma' + \frac{\eta}{T} \end{aligned} \tag{11}$$

If we wish to control x_{cm} and y_{cm}, we can now use a similar feedback law as

$$\tau_2 = \frac{1}{\beta'_y - \beta'_x\frac{\tilde{\gamma}_y}{\tilde{\gamma}_x}}\left(\tilde{\epsilon}_x\frac{\tilde{\gamma}_y}{\tilde{\gamma}_x} - \tilde{\epsilon}_y - \hat{v}_1\frac{\tilde{\gamma}_y}{\tilde{\gamma}_x} + \hat{v}_2\right) \tag{12}$$

$$\tau_1 = \frac{1}{\tilde{\gamma}_x}(-\tilde{\epsilon}_x - \beta'_x\tau_2 + \hat{v}_1), \tag{13}$$

where $\hat{v}_1$ and $\hat{v}_2$ are calculated as before to drive the system to desired references:

$$\begin{aligned} \hat{v}_1 &= v_1 + K_{dd}(\ddot{x}_{ref} - \ddot{x}_{cm}) \\ \hat{v}_2 &= v_2 + K_{dd}(\ddot{y}_{ref} - \ddot{y}_{cm}). \end{aligned} \tag{14}$$

Since we increased the order of the derivative the PFL is acting on, we now require an additional pole in the controller dynamics, which means we must estimate the acceleration of our control variables. This can be accomplished by simply using the most-recently stored torque commands, $\tau_1[t-T]$ and $\tau_2[t-T]$, in the place of τ_1 and τ_2 in Eq. 5 to give a reasonable numerical estimate. 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 remaining, real-valued pole, p_3. For a chosen set of ω_n, ζ, p_3, the gains are:

$$\begin{aligned} K_p &= p_3\omega_n^2 \\ K_d &= \omega_n^2 + 2p_3\zeta\omega_n \\ K_{dd} &= p_3 + 2\zeta\omega_n \end{aligned} \tag{15}$$

Lastly, by utilizing a PFL-based technique we have the flexibility to choose *any two* of our three variables of interest to control, and therefore if desired we could for example instead control y_{cm} and θ_B.

Since we can directly control only two out of our three variables, we construct hopping trajectories such that the uncontrolled variable is well behaved, and provide a method of dividing the stance phase to correct for perturbations such as energy loss at ground impact or rough terrain levels.

A. Center of Mass Trajectory Generation

Trajectories of the CoM and body attitude over time ultimately depend upon the ground reaction force (GRF), and in our approach, we correspondingly generate trajectories by planning allowable motion for a point-foot contact and for no rotation of the body; i.e., with the GRF planneded to point directly toward the CoM. Because the trajectories we present in this work are generated such that the total net torque w.r.t. the CoM is zero, under ideal conditions with no perturbations there would be no change in the angular momentum of the system, and the body angle would remain constant. Such trajectories can be constructed by abstracting our system to a system with a single point mass representing the CoM of the system, being acted upon by an external force F_t, which acts at the point-foot contact and is directly exactly at the CoM, as shown in Fig. 2.

We begin our construction of a CoM trajectory by characterizing y_{cm} as a function of x_{cm}. For this study we choose a quadratic function as:

$$y_{cm}(x_{cm}) = a + b(x_{cm} + c)^2 \tag{16}$$

The reasoning behind this choice is that it results in smooth motions and allows us to easily construct both symmetric and asymmetric trajectories. For most trajectories wanting to regulate a certain stride length or apex height we take $c = 0$ to enforce symmetry, however asymmetric trajectories with $c \neq 0$ can also be used to transition between different gaits. The acceleration of CoM trajectories are written as:

$$\begin{aligned} \ddot{x}_{cm} &= \frac{1}{m}F_t sin(tan^{-1}(\frac{x_{cm}}{y_{cm}})) = \frac{1}{m}F_t s_1 \\ \ddot{y}_{cm} &= \frac{1}{m}F_t cos(tan^{-1}(\frac{x_{cm}}{y_{cm}})) - g = \frac{1}{m}F_t c_1 - g \end{aligned} \tag{17}$$

And note that we can also write the acceleration of y_{cm} as:

$$\ddot{y}_{cm} = y''\dot{x}_{cm}^2 + y'\ddot{x}_{cm} \tag{18}$$

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. 16. Next, we combine Eq. 17 with Eq. 18 to solve for F_t as:

$$F_t = \frac{1}{c_1 - y's_1}(my'' - g) \tag{19}$$

And by substituting Eq. 19 into Eq. 17 we have a differential equation in terms of all known params, and by selecting desired initial conditions $x(0)_{cm}$, $\dot{x}(0)_{cm}$ and utilizing an ODE solver, we generate trajectories for CoM positions, velocities, and accelerations as functions of time for the stance phase. Since our controller needs only CoM stance phase trajectories to operate, we can control the gait step length by choosing trajectories with different terminating velocities. Given two trajectories with different $\dot{x}_{cm}$ terminating values, we construct asymmetric trajectories that switch to or from the desired trajectory set. To simply illustrate the ability to switch between different gaits, we designed three CoM trajectory sets for "Small", "Medium", and "Large" strides, such that the difference in step length between each set is roughly ten percent. Figure 3 shows example trajectories for y_{cm} and $\dot{x}_{cm}$ used to achieve gait switching.

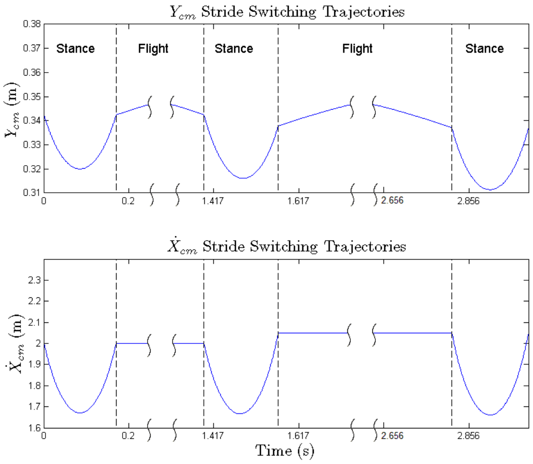

Fig. 3. We use symmetric CoM trajectories to regulate a specified stride length based on the value of $\dot{x}_{cm}$ at take-off, and use asymmetric trajectories to switch between them. Above shows example trajectories used to switch from a "medium" stride length to a "large" stride length, in which the value of $\dot{x}_{cm}$ at take-off is increased. Also note that the flight phases in this figure are truncated to show the stance phase in greater detail.

B. Division of the Stance Phase

Thus far we have assumed that our system is able to stay on our designed CoM trajectories perfectly for all time. A real implementation of this control framework however must remain robust to non-zero perturbations of the states, which in turn will inject angular velocity into the system that must be removed.

For the non-compliant version of the model, a significant source of perturbation to the CoM and body angle velocities is the ground impact, since all radial velocity along the leg is lost during an inelastic ground collision. An example simulation of this is shown in Fig. 4, which illustrates the discontinuities in slope (i.e., in velocity) that occur at impact and also shows results for a naive implementation of PFL in which only x_{cm} and y_{cm} are controlled, while θ_B is ignored.

For the compliant model, no energy is lost at impact and so there are also no instantaneous velocity changes. For a given stance-phase trajectory, it is possible the spring will not be finished decompressing when the trajectory terminates. One solution to this problem is to extend the stance phase trajectory by forward solving the ballistic equation for the CoM. In other words, $\dot{x}_{cm}$ remains constant (at the planned take-off velocity) and $y_{cm}(x_{cm})$ keeps the total vertical energy $(mgy + \frac{1}{2}m\dot{y}^2)$ also constant. Similar to the issue with the non-compliant mode, this will result in injection of some amount of undesired angular velocity.

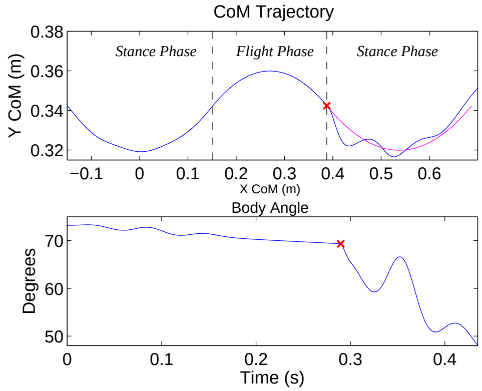

Fig. 4. For the non-compliant model, collision with the ground causes the CoM and body angle velocities to instantaneously change. When using PFL control only on x_{cm} and y_{cm} trajectories, these misalignments in initial velocity cause significant errors, as shown.

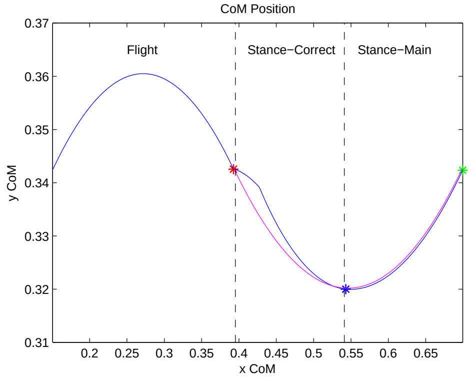

Fig. 5. To control x_{cm}, y_{cm} and θ_B, the stance phase is divided into two phases. The Correction phase removes any present angular velocity and brings the system to the desired CoM (x, y) trajectory in space. Actual simulation data is shown in blue, and compared to Fig. 4 the system is considerably more stable.

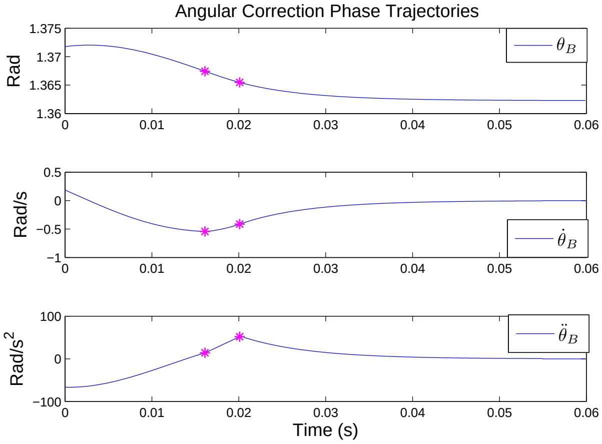

Fig. 6. Example smooth trajectory generated for the body angle in the Correction Phase. The trajectory is generated by combining two polynomials and a decaying exponential function, and solving for the required coefficients such that the touchdown initial conditions are aligned with the trajectory initial values. The trajectories return the body angle to its nominal value and remove all angular velocity.

To address these problems, we divide the stance phase into two parts, *Correction* and *Main*, as shown in Fig. 5. The Main phase is the PFL implementation presented in Eq. 8, while the Correction phase instead controls y_{cm} and θ_B using the same PFL set-up. This is done to remove any angular velocity introduced into the system, and return the body angle to a nominal value. Additional disturbances to the system due to ground height offsets (i.e., rough terrain) may be also treated as misalignments of the trajectory initial conditions at impact, therefore this correction strategy is also well suited to improve robustness to disturbances, more generally.

At the instant of ground impact, the body angle of the system may be above or below the desired system value. The Correction Phase trajectories for the body angle are planned on the fly to match the initial body angle, velocity, and acceleration conditions and then to converge to zero angular momentum by the end of the correction phase. We construct smooth trajectories from a combination of polynomial and exponential functions. Figure 6 for example illustrates a trajectory generated for negative body angle drift with positive angular velocity and negative body angular acceleration at touch-down.

After the body angle trajectory is chosen, we construct a valid trajectory for y_{cm}. The trajectories for y_{cm} are generated by casting the y_{cm} trajectory as a function of our uncontrolled variable, x_{cm}, as we previously defined in Eq. 16. We therefore set the y_{cm} reference trajectories in the Correction phase as a function of the current x_{cm}, i.e.,

$$\begin{aligned} y_{ref} &= y_{traj}(x_{cm}) \\ \dot{y}_{ref} &= \dot{y}_{traj}(x_{cm}) \\ \ddot{y}_{ref} &= \ddot{y}_{traj}(x_{cm}) \end{aligned} \qquad (20)$$

During the flight phase, the mass stance link can be repositioned arbitrarily without affecting the rest of the dynamics, so we only have one available actuator τ_2, which we use to implement simple collocated PFL on the spine angle θ_3, in order to return the angle to the initial stance value. The trajectory is generated in a similar manner as the body angle trajectories, and an example is shown in Fig. 7.

IV. Results on Rough Terrain

A simulation study was conducted using Matlab's ODE solver. Our proposed control algorithm was successfully implemented in simulation for the model shown in Fig. 1, for both the case of removing the compliant leg and using the PFL law shown in Eq. 8, and for the case of including the compliant leg and using the extended PFL control law seen in Eq. 14. Both implementations use the same trajectory generation, as discussed in Section III, which was designed to be general enough to work for both systems. The non-compliant system uses the traditional acceleration-based PFL construction (Eqs. 6-8), while the compliant system applies PFL to set the jerk (Eqs. 12-14). The parameters used in this study are shown in Tables I and II.

Simulations were performed for both even and uneven ground levels, and the controller successfully regulates

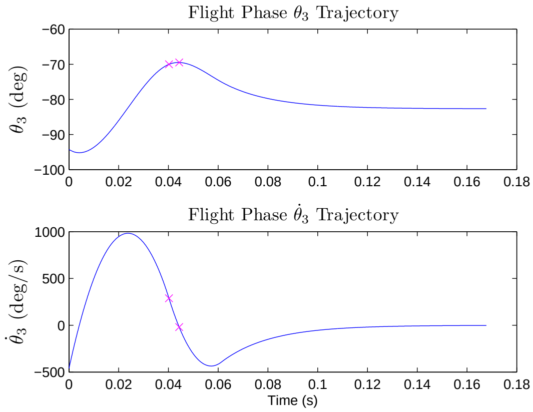

Fig. 7. Example smooth trajectory generated for spine angle, θ_3, during the flight phase. The trajectory is generated in an identical way as the body angle trajectories previously seen in Fig. 6, by using a combination of polynomials and decaying exponential functions to ensure initial condition alignment and to provide convergence with zero velocity to initial spine angle.

m_1, m_2	2.0 kg	Hip masses
J_1, J_2	0.0052 $kg\ m^2$	Joint inertias
L_0	0.2540 m	Link natural lengths
$\theta_B(0)$	1.28 rad	Initial body angle
$x_{small}(0)$, $\dot{x}_{small}(0)$	-0.16 m, 1.9 m/s	Small stride IC
$x_{med}(0)$, $\dot{x}_{med}(0)$	-0.15 m, 2.0 m/s	Med stride IC
$x_{large}(0)$, $\dot{x}_{large}(0)$	-0.14 m, 2.1 m/s	Large stride IC
a_{small}, b_{small}	0.33, 1	Small stride coeffs
a_{med}, b_{med}	0.32, 1	Med stride coeffs
a_{large}, b_{large}	0.31, 1	Large stride coeffs
ω_n, ζ	17.6, 0.99	PFL gain parameters

TABLE I
SIMULATION PARAMETERS FOR NON-COMPLIANT MODEL

switching between different stride lengths. An example terrain map is shown in Fig. 8, which is randomly generated prior to runtime. Figures 9 and 10 show example reference tracking data taken from simulations during the stance phase for the compliant and non-compliant implementations, respectively. During the Correction phase, the controller is able to drive angular velocity toward zero, return the body angle to the nominal value, and converge back to CoM trajectories reasonably well. Stride length switching results on flat and rough terrain, for the compliant and non-compliant implementations are shown in Figures 11 through 13.

The effect of the terrain being uneven is essentially the same as that of adding an undesired misalignment to y_{cm}. The effect of rough terrain can be observed by viewing x_{cm}, y_{cm}, and θ_B at touchdown (Fig. 14) and at the switching point from the Correction phase to the Main phase (Fig. 15). When rough terrain is added, the values touchdown states have noticeable variability, as clearly seen in Fig. 14. Since the Correction phase is built to correct for misalignments, the variance is significantly reduced by the time the controller switches to Stance-Main, illustrated in Fig. 15. In particular, the body angle is very consistent at the time of the switch, which can be noted by observing that the scale of the body

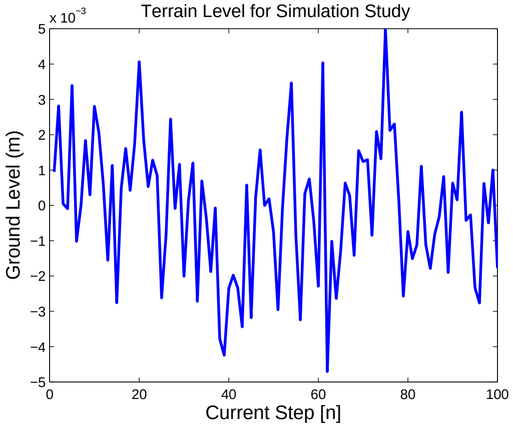

Fig. 8. Randomly generated terrain level used for simulation study, with a maximum ground disturbance of 0.5cm.

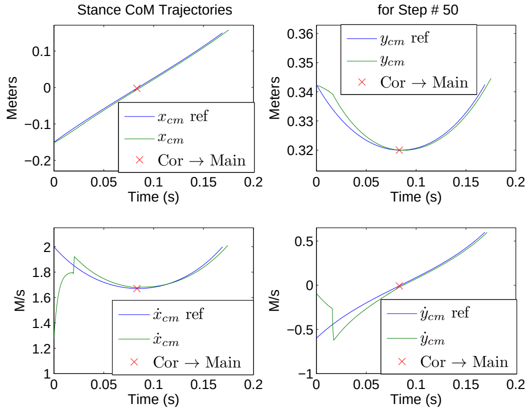

Fig. 9. Above shows example CoM stance phase data for simulating the non-compliant model on rough terrain using the proposed control framework. The velocities are mis-aligned at time zero from the ground impact, and the system is able to converge to the desired trajectories with reasonable error.

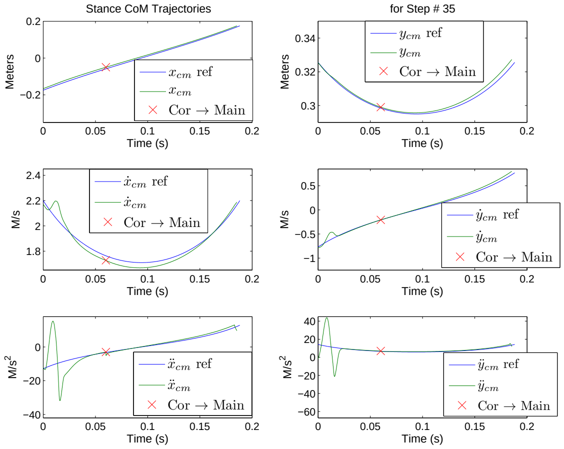

Fig. 10. Above shows example CoM stance phase data for simulating the compliant model on rough terrain using the proposed control framework. The system is able to maintain reasonable error to the desired CoM trajectories in the presence of terrain noise.

m_1, m_2	2.0 kg	Hip masses
J_1, J_2	0.0052 $kg\ m^2$	Joint inertias
L_0	0.2540 m	Link natural lengths
k_{leg}	2,625 N/m	Spring constant
b_k	25.36 Ns/m	Spring damping
$\theta(0)_B$	1.37 rad	Initial body angle
$x_{small}(0)$, $\dot{x}_{small}(0)$	-0.103 m, 1.82 m/s	Small stride IC
$x_{med}(0)$, $\dot{x}_{med}(0)$	-0.141 m, 2.0 m/s	Med stride IC
$x_{large}(0)$, $\dot{x}_{large}(0)$	-0.175 m, 2.2 m/s	Large stride IC
a_{small}, b_{small}	0.341, 1.49	Small stride coeffs
a_{med}, b_{med}	0.327, 1.35	Med stride coeffs
a_{large}, b_{large}	0.295, 1	Large stride coeffs
ω_n, ζ, p_3	25, 0.9, 700	PFL gain parameters

TABLE II

SIMULATION PARAMETERS FOR COMPLIANT MODEL

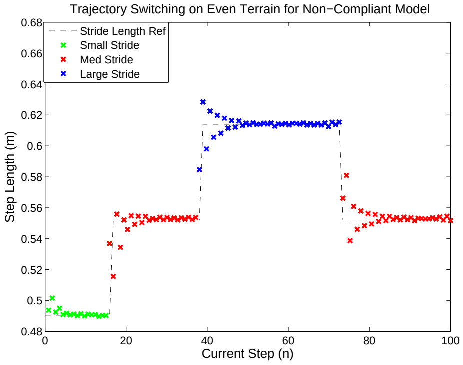

Fig. 11. Trajectory switching simulation on even terrain for non-compliant implementation.

angle axis in Fig. 15 is significantly more confined than at touchdown. Results are qualitatively similar for both the compliant and non-compliant models, and full results for each are not included simply due to space constraints.

V. CONCLUSIONS AND FUTURE WORK

We have presented control methods for an underactuated hopping robot, both with and without a compliant leg, that allows for direct CoM trajectory planning via an extension of Partial Feeback Linearization techniques. We provide a robust control method for correcting perturbations to both models at each step. We focus on tracking a sequence of trajectories for a single leg in support with the ground, robust to variations in ground height and other variability, which enables switching among gaits to vary foothold selection and speed.

There are specifics of the system we do not model in this study, and leave open for future work. We are not focused on finding optimal trajectories for the system to follow, for example. Energy efficiency and limitations of real-world actuators are also not considered in this work, and sensor accuracy is assumed such that full state information is available.

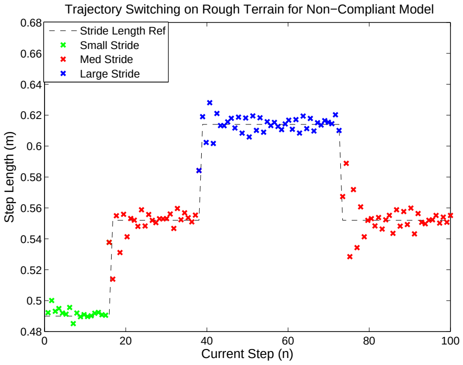

Fig. 12. Trajectory switching simulation on rough terrain for non-compliant implementation.

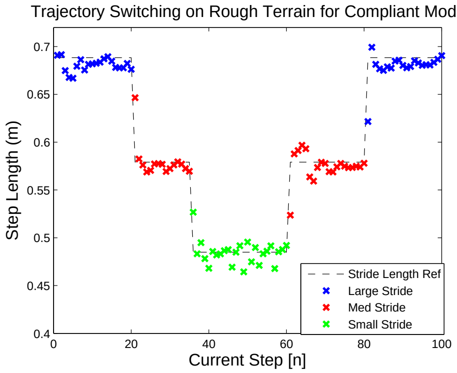

Fig. 13. Trajectory switching simulation on rough terrain for compliant model.

Future work includes optimization of trajectories both for energy efficiency and for robustness and potentially considering gaits other than zero angular momentum hopping. A driving goal is to develop high-level motion plans that switch optimally between feasible reference trajectories, based on information about upcoming terrain. We also plan to investigate the idea of using similar higher order PFL techniques to develop similar control ideas for hopping robots with series-elastic actuators acting to modify the effective impedance of the leg over time.

ACKNOWLEDGMENTS

This work is supported in part by DARPA's M3 program (Grant No. W911NF-11-1-0077) and by the Army Robotics CTA (Grant No. W911NF-08-R-0012).

REFERENCES

[1] 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.

[2] R. Player, M. Beuhler, and M. Raibert, "BigDog," in *Proc. SPIE*, vol. 6230, Jun 2006.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

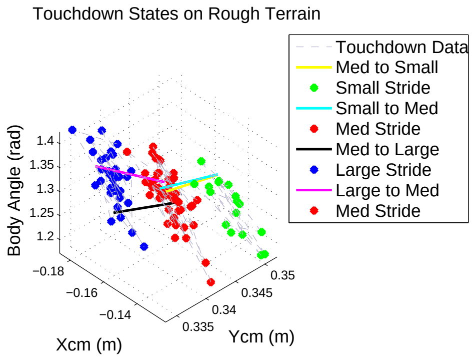

Fig. 14. Touchdown CoM states for a simulation on rough terrain, for the non-compliant implementation. The states are significantly scattered due to uncertain ground level at impact.

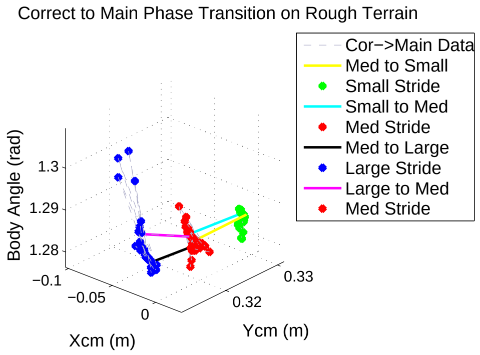

Fig. 15. Correction to Main CoM transition states for rough terrain, for the non-compliant implementation. The controller is able to consistently correct for the large errors at touchdown, and bring the states to more localized points.

[7] M. S. Fischer, N. Schilling, M. Schmidt, D. Haarhaus, and H. Witte, "Basic limb kinematics of small therian mammals," *Journal of Experimental Biology*, vol. 205, no. 9, pp. 1315–1338, 2002.

[8] K. F. Leeser, "Locomotion experiments on a planar quadruped robot with articulated spine," Master's thesis, MIT, 1996.

[9] B. M. Haueisen, *Investigation of an Articulated Spine in a Quadruped Robotic System*. PhD thesis, University of Michigan, 2011.

[10] M. Kani, M. Derafeshian, H. Bidgoly, and M.Ahmadabadi, "Effect of flexible spine on stability of a passive quadruped robot: Experimental results," in *IEEE Int. Conf. on Robotics and Biomimetics (ROBIO)*, 2011.

[11] M. W. Spong, "Underactuated mechanical systems," in *Control Problems in Robotics and Automation*, pp. 135–150, Springer, 1998.

[12] S. Kochuvila, S. Tripathi, and T. S. B. Sudarshan, "Control of a compass gait biped robot based on partial feedback linearization," in *Advances in Autonomous Robotics*, pp. 117–127, Springer, 2012.

[13] M. W. Spong, "Partial feedback linearization of underactuated mechanical systems," in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 314–321, 1994.

[14] M. W. Spong, "Swing up control of the acrobot using partial feedback linearization," in *Proc. American Control Conference (ACC)*, pp. 2158–2162, 1994.

[15] K. Byl, M. Byl, M. Rutschmann, B. Satzinger, L. van Blarigan, G. Piovan, and J. Cortell, "Series-elastic actuation prototype for rough terrain hopping," in *Proc. IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, pp. 103–110, 2012.

[16] G. A. Cavagna, H. Thys, and A. Zamboni, "The sources of external work in level walking and running," *Physiology*, vol. 262, pp. 639–657, 1976.

[17] K. Sasaki and R. R. Neptune, "Muscle mechanical work and elastic energy utilization during walking and running near the preferred gait transition speed," *Gait and Posture*, vol. 23, pp. 383–390, 2006.