Toward agile control of a flexible-spine model for quadruped bounding

Katie Byl[a], Brian Satzinger[a], Tom Strizic[a], Pat Terry[a] and Jason Pusey[b]

[a]Robotics Lab, University of California at Santa Barbara, USA;
[b]Army Research Lab (ARL), Aberdeen, MD, USA

ABSTRACT

Legged systems should exploit non-steady gaits both for improved recovery from unexpected perturbations and also to enlarge the set of reachable states toward negotiating a range of known upcoming terrain obstacles. We present a 4-link planar, bounding, quadruped model with compliance in its legs and spine and describe design of an intuitive and effective low-level gait controller. We extend our previous work on meshing hybrid dynamic systems and demonstrate that our control strategy results in stable gaits with meshable, low-dimension step-to-step variability. This meshability is a first step toward enabling switching control, to increase stability after perturbations compared with any single gait control, and we describe how this framework can also be used to find the set of n-step reachable states. Finally, we propose new guidelines for quantifying "agility" for legged robots, providing a preliminary framework for quantifying and improving performance of legged systems.

1. INTRODUCTION

One goal in developing legged robot systems is to provide a high degree of agility. Intuitively, being agile means that future states (i.e., position and velocity variables defining snapshots of the dynamic robot as it moves) are not deterministically pre-ordained and can instead be controlled to a high degree. For example, a robot that repeats a constant, low-level gait sequence arguably lacks agility, no matter how robust it is to terrain variability. In this work, we introduce two related concepts of agility: many-to-one and one-to-many control authority. By many-to-one, we mean that a system should be capable of arriving at the same final state (e.g., landing at the far side of a ditch-like obstacle) from a large set of initial conditions; similarly, the system should be capable of recovering from many different perturbations to return back to one particular, nominal gait. As a complementary notion, one-to-many agility implies that a robot starting at one particular initial condition (e.g., sitting at rest) should be able to reach many different states in a characteristic time, and that it should do so accurately.

In this work, we present a 4-link planar, bounding, quadruped model with compliance in its legs and spine and design five low-level gait controllers. In this planar model, a single link represents the pair of back legs, two links represent a flexible spine, and the final link captures both front legs. The system is highly underactuated, with only one active hip torque available for control when only one leg link is in contact with the ground. Our low-level gait control uses the stance leg hip torque to drive the overall body dynamics to follow a desired, time-based reference trajectory using a simple proportional plus derivative (PD) controller. We discuss intuitive design of the low-level control briefly, toward a long-term goal of developing a family of such controllers that facilitate switching control and on methods to develop intelligent high-level switching policies to improve agility.

This work extends our previous methods on meshing hybrid dynamic systems and builds on recent work to quantify agility of a dynamic legged system.[1] A similar optimization framework can arguable generate switching policies to increase stability after perturbations compared with any single gait control and finds the set of n-step reachable states. Solutions to these problems inspire new metrics quantifying agility for legged robots by quantifying the volumes of initial condition states for the many-to-one (ditch-jumping) problem and of final achievable states in the one-to-many problem formulation.

The addition of a flexible spine to quadrupedal robots is motivated by a desire to improve energy efficiency, robustness, and range of motion over comparable rigid spine quadrupedal designs. In 1996, Leeser[2] built a planar quadruped robot with hydraulic hip, spine, and leg actuators, as well as passive air springs in the legs. It was demonstrated on a planarizing boom and treadmill and used Raibert style control. In 2012, UPenn's Canid robot was developed.[3] Canid has electrically actuated hip joints, as well as an electrically actuated spine with an elastic element. The design of controllers for Canid is an open research problem, and serves as a motivation for the work in this paper.

Other simulation work in this area includes Haueisen,[4] who considered the effect of an actuated spine on a planar quadrupedal simulation model consisting of 6 point masses connected by rigid links and actuated rotational knee, hip, and spine joints. Culha[5] considered a similar model, but with passive linear spring-damper legs instead of Haueisen's actuated knee. In both cases, the spine is actuated and does not include a passive elastic element.

Figure 1 shows a simplified model for a Canid-like robot system. The model allows for only planar motion, and the flexible spine of the robot is approximated as a single degree of freedom rotational joint. During stance, we assume that there is a point-foot contact with the ground, with no slip and with no actuation about the contact pivot.

Toward simplifying modeling while capturing the most essential elements of dynamics and control for the system, this paper will consider a simplified model, shown in Figure 2, consisting of two point masses located at the hip joints, a passive torsional spring-damper spine, and passive linear spring-damper legs. (Only the springs are shown in the image, for simplicity.) Conceptually, there are actuators located at both hips. However, due to the massless legs, the hip actuators are only effective one at a time during the stance phase of the connected leg.

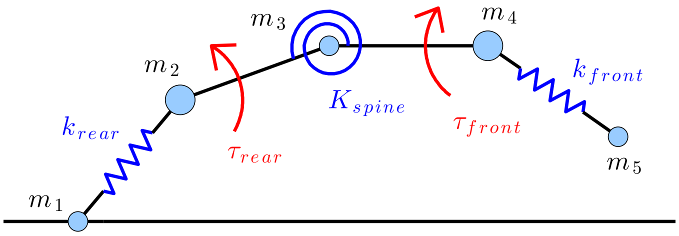

Figure 1. Planar 5-Mass Model Schematic, with a passive spine, torques at front and rear hips, and a small unsprung mass at each of the feet.

2. STATE OF THE ART

In 1996, Leeser built a quadruped robot with hydraulic hip, spine, leg actuators, and passive air springs in the legs.[2] Leeser demonstrated his control strategies on a treadmill with the robot mounted on a planarizing boom. Canid,[3] also mentioned earlier, has also demonstrated bounding behaviors for several steps in both actual hardware and in a high-fidelity simulation using LMS software, from Siemens*.

Simulation work with actuated-spine models also includes that of Haueisen, who considered the effect of an actuated spine on a planar quadrupedal model consisting of 6 point masses connected by rigid links and actuated rotational knee, hip, and spine joints.[4] Culha considered a similar model, but with passive linear spring-damper legs.[5] In both cases, the spine is actuated and does not include a passive elastic element.

Work by Poulakakis et. al. has studied the stability of passive dynamics of a bounding saggital plane quadrupedal model with a rigid spine.[6] More recently, Poulakakis and Cao have studied the passive dynamics of a similar model with an unactuated flexible spine, and discovered families of unstable fixed points.[7]

3. APPROACH

As previously mentioned, we study the planar model shown in Figure 2. There are actuators located at each hip. However, due to the massless legs, the hip actuators are only effective one at a time during the stance phase of the connected leg (our model excludes double-stance configurations). The control inputs, therefore, comprise the leg angles (set at the moment the leg leaves the ground) as well as a torque signal τ_{hip} defined during stance. Controllers to compute these signals must be designed in order to stabilize the system.

*See $http://www.plm.automation.siemens.com/en_us/products/lms/$

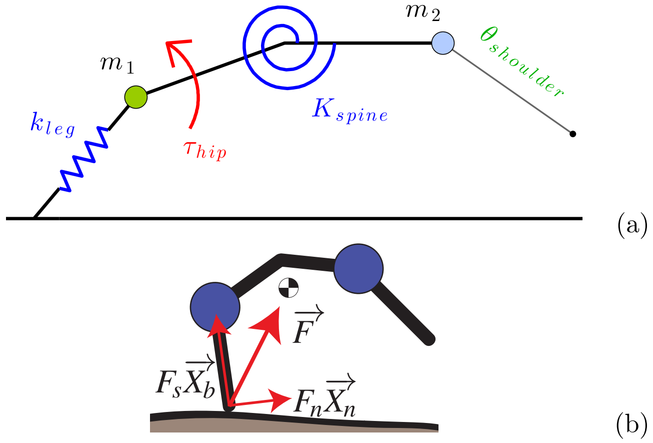

Figure 2. (a) Schematic model of the planar quadruped model showing the point masses, springs, and hip actuator. Dampers in parallel with the springs are not illustrated, for compactness and clarity. Only the first three links affect the dynamics, since the last link (past $\theta_{shoulder}$) is modeled as massless. (b) Simplified view showing ground reaction force vector, orthogonal force components, and center of mass. Note that the magnitude of F_s is determined by the compression of the leg spring, while F_n is directly affected by τ_{hip}.

In order to study the body pitch dynamics analytically and derive a control law to stabilize the body pitch during stance, a simplified model (Figure 3) is derived as follows. The legs are removed from the higher-order model, the masses are assumed to be equal, and only two degrees of freedom (r and θ) are considered. The torsional spring (with spring rate K_t and neutral angle θ_0) at the center of the spine is unchanged. The distance from each point mass to the spine joint is L. A force at the stance foot both accelerates the center of mass and produces a torque, τ, that can be used to control body attitude during stance..

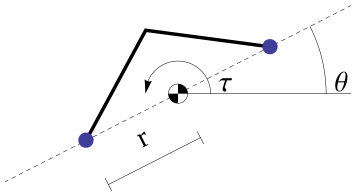

Figure 3. Simplified system used to derive feedback law

4. MODEL DERIVATION

4.1 Continuous Time Dynamics

When a single foot is in contact with the ground, the four links of the Canid-like model form a simple chain, as depicted in Figure 4. Beginning by choosing the generalized coordinates, θ_1 is the absolute angle between the most recent (or current) leg contact with the ground and the first hip mass, while θ_2 and θ_3 are relative angles of the next joints, in order, and r_1 is the length of the last (or current) stance leg. Because we model the last link as massless, θ_4 and r_4 (the last leg angle and length, respectively) do not affect the dynamics of the system. The continuous equations of motion are then derived as follows.

First, assuming the foot touches the terrain at the origin, compute the locations of the four points $[x_b, y_b]'$,$[x_h, y_h]'$,$[x_f, y_f]'$, and $[x_4, y_4]'$. The equations below are included primarily to intuitively clarify the kinematic definitions of the angles and lengths within the system.

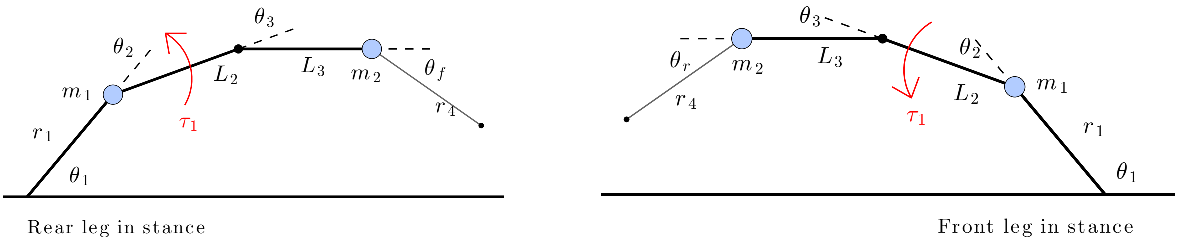

Figure 4. Definitions of angles and lengths for the two-mass planar model. Spring-damper assembly at the stance leg and at the flexible spine are not shown, for clarity. Lengths L_2, L_3 and r_4 are constant lengths, while r_1 varies with leg spring length. The model is symmetry, front to back, so that relabelling occurs at each impact, as shown. Impact dynamics of the front leg result in an instantaneous lift-off of the rear foot, so that there is at most one leg in contact with the ground during steady bounding of the model.

$$\begin{aligned} x_b &= r_1 * cos(\theta_1) \\ y_b &= r_1 * sin(\theta_1) \end{aligned} \tag{1}$$

$$\begin{aligned} x_h &= x_b + L_2 * cos(\theta_1 + \theta_2) \\ y_h &= y_b + L_2 * sin(\theta_1 + \theta_2) \end{aligned} \tag{2}$$

$$\begin{aligned} x_f &= x_h + L_3 * cos(\theta_1 + \theta_2 + \theta_3) \\ y_f &= y_h + L_3 * sin(\theta_1 + \theta_2 + \theta_3) \end{aligned} \tag{3}$$

$$\begin{aligned} x_4 &= x_f + r_4 * cos(\theta_1 + \theta_2 + \theta_3 + \theta_4) \\ y_4 &= y_f + r_4 * sin(\theta_1 + \theta_2 + \theta_3 + \theta_4) \end{aligned} \tag{4}$$

The kinetic co-energy depends on the velocities of the two point masses $[\dot{x}_b, \dot{y}_b]'$ and $[\dot{x}_f, \dot{y}_f]'$, which can be found easily from the equations above, using the chain rule. The kinetic co-energy T^* is:

$$T* = 0.5m_b(\dot{x}_b^2 + \dot{y}_b^2) + 0.5m_f(\dot{x}_f^2 + \dot{y}_f^2) \tag{5}$$

and the potential energy V depends on both the gravitational potential energy of the two masses and the potential energy in the spine spring. (For simplicity, the force due to the series spring in the leg is modeled later as an external force so it can be set to zero when the foot leaves the ground, so the linear spring is not included in V.)

$$V = m_b g y_b + m_f g y_f + 0.5K_{spine}(\theta_3 - \theta_{3,null})^2 \tag{6}$$

Finally, using the Lagrangian method, a non-linear ODE model can be defined in terms of eight states, X (Equation 7):

$$X = [\theta_1, \theta_2, \theta_3, r_1, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{r}_1]) \tag{7}$$

θ_4 and r_4 are excluded here because the fourth segment is massless and does not affect the dynamics, although θ_4 will re-appear when the model is extended to a hybrid system, to handle detection of transitions between front and rear footholds.

4.2 Hybrid Model Extension

The continuous time model defined in the abbreviated derivation above does not handle transitions between front-stance and rear-stance configurations. These transitions can easily be handled by extending the continuous dynamics into a hybrid system with discrete transitions.

The state vector is extended to include four auxiliary states, X_{aux}. θ_4 is included in the hybrid model despite its exclusion from the continuous model because its value is necessary to detect terrain collisions with the non-stance leg. $x_{foothold}$ is the horizontal location of the stance foot on the terrain (the vertical location is defined by the terrain function). t_{stance} is a timer used to count the elapsed time since the last touchdown event. Finally, $stancefoot$ is zero when the rear foot is the active stance foot and one when the front leg is. It is important to note that $stancefoot$ only changes values during a touchdown event, and not during a takeoff event. This is because flight is modeled as a special case of stance where one leg is attached to the ground, but gives two passive degrees of freedom and exerts no forces on the robot. Therefore, $stancefoot$ reflects either the current stance foot, or the most recent stance foot. Takeoff events can be detected by checking if $r_1 > r_{1,a}$; i.e., the spring in the stance leg recovers from compression until is reaches its neutral length, at which point the leg loses contact with the ground.

$$X_{aux} = [\theta_4, x_{foothold}, t_{stance}, stancefoot]' \tag{8}$$

During continuous dynamics, the derivatives of the auxiliary states are zero, except for $\dot{t}_{stance}$, which is one.

$$\dot{X}_{aux} = [0, 0, 1, 0]' \tag{9}$$

When the non-stance foot makes contact with the terrain, a discrete event (jump) occurs. The flag variable $stancefoot$ is changed to $!stancefoot$, t_{stance} is reset to zero, and $x_{foothold}$ is set to the horizontal location of the touchdown (x_4). θ_4 is reset to an initialization value, either $\theta_{4,back}$ or $\theta_{4,front}$ depending on the current stance leg.

$$X^+_{aux} = [\theta_4^+, x_4, 0, !stancefoot]' \tag{10}$$

$$\theta_4^+ = \begin{cases} \theta_{4,back} \text{ if } stancefoot = 0 \\ \theta_{4,front} \text{ if } stancefoot = 1 \end{cases} \tag{11}$$

During a stance transition, the continuous states $X = [\theta_1, \theta_2, \theta_3, r_1, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{r}_1]'$ are updated to reflect the changed coordinate system and preserve the velocities in the system.

The complete hybrid state Z consists of the eight continuous states and the four auxiliary states, forming a 12-dimensional vector.

$$Z = [X, X_{aux}]' \tag{12}$$

In the Simulink model, there are two functions, f and g that manage changes in the state vector Z. The flow function f computes the continuous dynamics (including Equation 9), and the jump function g computes the discrete jumps during stance-leg switching events. The jump function g is used when the collision detection state machine reaches the JUMP state (Figure 6 in Section 4.3).

$$\dot{Z} = f(Z) \tag{13}$$

$$Z^+ = g(Z) \tag{14}$$

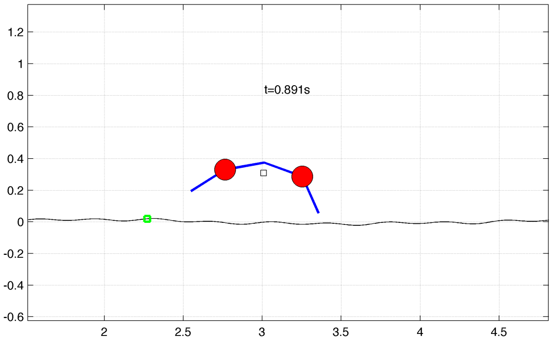

Figure 5. Frame from animation showing sum-of-sinusoids terrain

4.3 Terrain, Collision Detection Algorithm

The terrain height is defined as a function of horizontal position $y = terrain(x)$. Currently, a sum-of-sinusoids approach is used to give a smooth hilly terrain, although a future extension would be to generate the terrain stochastically. Figure 5 shows an example of this terrain, although the amplitude of the sine functions can be configured to be larger or smaller, or be modulated by a function of the horizontal position (for example, to create a gentle ramp from the origin).

The location of the free foot can be computed using Equation 4. The difference in height, δ_y, between y_4 and $terrain(x_4)$ then determines whether the last limb has collided with the ground; i.e., δ_y must be greater than or equal to zero. A finite state machine (Figure 6) is used to simplify collision detection. A discrete state transition cannot occur unless the foot has already risen above the terrain and then falls below it.

$$\delta_y = y_4 - terrain(x_4) \tag{15}$$

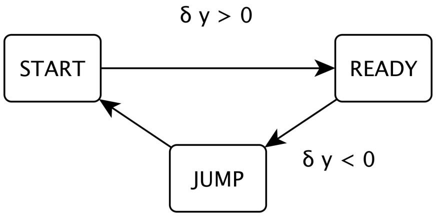

Figure 6. State machine used in terrain collision detection

4.4 Model Parameters

Parameters of the two-mass planar system are given in Table 1. Note the symmetry of the model. As a result, the equations of motion when either the front or rear leg is in stance are identical, so long as a relabeling of elements is performed, as illustrated in Figure 4.

Table 1. Model Parameters

Description	Symbol	Value	Units
leg stiffness	k_{leg}	1550	[N/m]
leg damping	b_{leg}	79	[N/(m/s)]
spine stiffness (torsional)	K_{spine}	60	[Nm/rad]
spine damping (torsional)	B_{spine}	2	[Nm/(rad/s)]
neutral length of spring-leg	$r_{1,null}$	0.254	[m]
section leg segment	L_2	0.254	[m]
section leg segment	L_3	0.254	[m]
massless end segment	r_4	0.254	[m]
rear point mass	m_1	2	[kg]
front point mass	m_2	2	[kg]

5. BODY PITCH CONTROLLER

The body pitch θ can be controlled during stance by torques exerted about the center of mass by the stance leg. The external force F can be broken into two orthogonal components (Figure 7). F_s is the force along the leg, and is passively generated by the linear spring-damper component in the leg. The other component, F_n, is determined by the hip torque τ and the leg length. Therefore, F_n can be chosen in order to direct the overall ground reaction force F in a particular direction, subject to the constraint that F_s cannot be changed. If F is chosen to point precisely through the center of mass, then no torque is exerted on the body. Otherwise, a moment arm exists and the force will induce rotational accelerations.

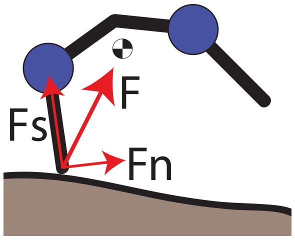

Figure 7. Ground reaction force vector F, and components F_s and F_n

6. GAIT CONTROL OVERVIEW AND RESULTS

Figures 8 and 9 show an animation and corresponding trajectory data for one cycle of a continuous bounding gait for a MATLAB model of the simplified planar quadruped system. The goal of the first phase of the gait is to direct motion of the center of mass of the system (midway between the two point masses) upward and to the right by pushing on the rear ground contact appropriately. More specifically, the rear stance phase of the gait aims initially to keep the ground reaction force (GRF) directed slightly in front of the center of mass, in order to pitch the robot up. Before losing contact with the ground, the GRF shifts to impact a moment in the reverse direction, so the body begins to rotate clockwise before the start of the flight phase and continues with nearly constant angular rotation during ballistic flight. Although designed with desired ground forces in mind, this motion is achieved by commanding simple and intuitive reference trajectories for a subset of the joint angles. Tuning these trajectories sets the angle of take-off and the timing of body rotation.

Our focus in this paper is on the practicality of such an approach on producing a stable gait, rather than details of specific reference trajectories. The most critical aspect is the take-off phase, which is controlled by driving the body angle defined by the line connecting the rear and front point masses in the model. Reference and actual trajectories for the body angle during this phase are shown in Figure 10. The reference is used

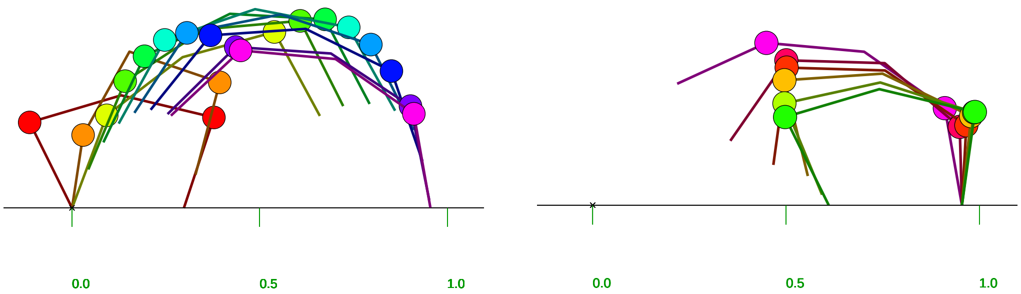

Figure 8. Animation of stable gait. Poses shown are snapshots at increments of 0.05 seconds, with a final frame included when impact of a leg with the ground occurs, ending a given phase of the gait. The left image includes the stance phase for the rear leg and the subsequent flight phase of the system, while the right image shows the stance phase for the front leg, in which the rear leg is reset for the next push-off.

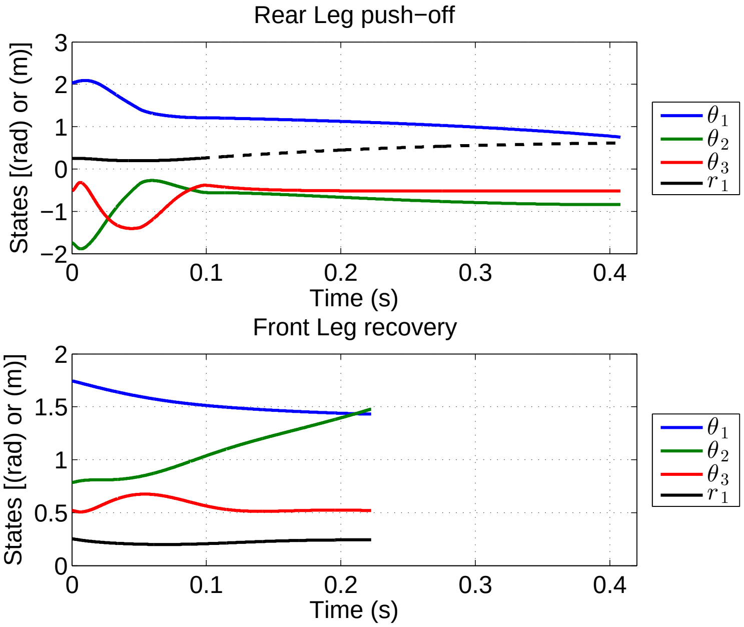

Figure 9. Joints and sprung leg length over time. Top plot corresponds to push-off and flight phases (at left in Fig. 8), and data in bottom plot corresponds to the phase in which front leg touches the ground (at right in Fig. 8). The black lines shows the distance from each respective ground contact (rear or front stance leg) to the next hip joint, and the dashed line indicates when this distance exceeds the neutral spring length, at which point the true leg is at its undeflected lengths and the system is in its flight phase.

to drive the only actuated degree of freedom, the hip of the contact leg, via τ_{hip}. The PD control law is $\tau_{hip} = -K_p(\theta_{ref} - \theta_{body}) - K_d\dot{\theta}_{body}$, with torque limits of ±50 [Nm] in τ_{hip}. ($K_p = 4000$ and $K_d = 120$ here.) The goal is to achieve a combination of a positive body angle and negative angular velocity at take-off, compatible with landing cleanly on the front legs without an excessive pitching moment that might flip the system head of heels. After the front legs touch down, the rear legs are reset as then drop back to the ground, in preparation for the next take-off phase. Both phases are depicted in Fig. 8.

The second half of the gait is much less challenging to design and tune. There is again a single actuated joint, which is now located at the hip of the new ground contact limb, i.e., the hip of the front leg. Goals during the front limb stance phase are essentially to wait for the rear leg to regain contact with the ground and to reset the posture of the robot to prepare for the next rear-stance phase. Intuitively, noise in terrain height and other perturbations cause the initial, ballistic trajectory of the center of mass to vary, while control of internal joints and the stiff passive spine together maintain relatively minimal variability in the configuration of the robot, from step to step.

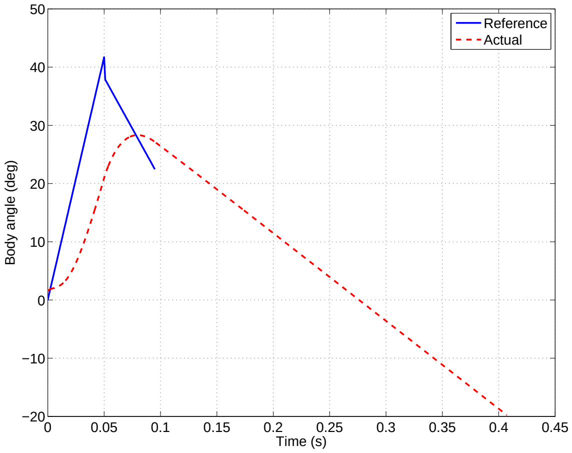

Figure 10. Reference trajectory for and actual values of body angle during the rear leg stance phase and the flight phase. The system is airborne after approximately 0.1 seconds, at which point the body rotates at nearly a constant angular velocity, i.e., with the spine remaining close to its neutral position. The front leg contacts the ground at the end of the trajectory shown.

A Jacobian analysis was performed for this gait by perturbing each of the states of the fixed point of a Poincaré snapshot in the cycle and observing the resulting magnitude of the perturbations away from the fixed point after one steps. The eigenvalues and eigenvectors of this Jacobian describe the linearized step-to-step dynamics for small perturbations about the nominal gait. Eigenvalues correspond to discrete-time poles here, so that the system is locally stable only when all eigenvalues are within the unit circle (i.e., less than one in magnitude), and the smaller in magnitude they are, the more rapidly perturbations decay away.

For the gait shown the two largest eigenvalues of the step-to-step Jacobian are $\lambda_1 = -0.45$ and $\lambda_2 = 0.23$. The third eigenvalue is about 0.01 in magnitude and the rest are much smaller. The eigenvectors associated with these small eigenvalues indicate "directions" in state space in which perturbations disappear almost entirely (i.e., to within less than 1%) in a single step. Another intuitive interpretation of this is that one would anticipate there to be essentially only two degrees of freedom of significance in the step-to-step perturbations due to environmental noise added to the system.

This observation can also be illustrated by plotting the states visited during locomotion on rough terrain. We tested the controlled bounding system in MATLAB on simulated rough terrain. Figure 5 shows a sample of this rough terrain, and Figure 11 shows a set of 100 states each recorded once per step on rough terrain. Because the full state includes 8 position and velocities values, the data are shown on 6 subplots, each with the same two values (rear hip position and rear leg velocity, which show the greatest variability with respect to one another) plotted on the x and y axes, with the remaining 6 states plotted on z. These projections show that the states lie on roughly 2D surfaces. As a result, it is practical to mesh the step-to-step dynamics of the system, following an approach developed within our group[8–11] and in turn enabling analysis of long-term stability. This also provides an important step toward enabling switching control, to improve stability on rough terrain and also to deliberately drive the system to follow a prescribed set of footholds and/or to vary speed fluidly over time through appropriate switching among a family of low-level controllers.

7. CONCLUSIONS, DISCUSSION AND FUTURE WORK

In this paper, we present an intuitive and effective control strategy for a planar model of a bounding quadruped. Toward improving the range of operability of such systems, we also introduce a new framework for qualitatively

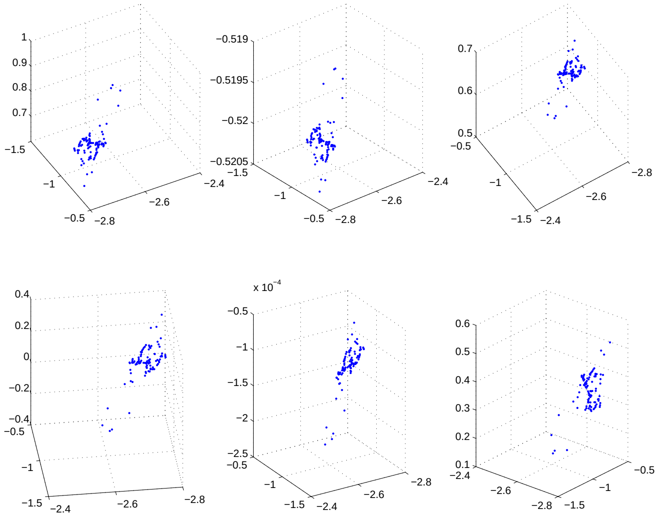

Figure 11. Step-to-step impact states plotted to visualize the low dimensionality of reachable states for our bounding gait on rough terrain.

describing "agility" and demonstrate initial steps toward a new methodology for the creation of agile legged robot performance. A key step in our approach is to first generate an intuitive and effective means of producing a family of stable motions, to achieve a range of forward speeds, ground clearance, and capability to cope with change in terrain height. We observe that the action of employing such an active controller on the system drives the system states to a relatively small-dimensional manifold within the full state space. This allows us to mesh, to approximate the step-to-step dynamics of the system. By starting with some nominal initial condition for the system, we can then deterministically generate a mesh covering the entire reachable portion of state space and can simultaneously learn the mappings between states, which depend on both the high-level controller that is selected and the upcoming terrain profile.

Once such a mesh and state-to-state mapping are both generated, the problem of switching among controllers to drive the system toward a desired state simply becomes a Markov decision process (MDP). This framework can then be employed either to reject disturbances (e.g., from terrain height variability) by driving the system back toward a steady-state motion or to intentionally exploit achievable variability to plan agile motions with variable speed and footstep length over time.

We propose that a good metric for agility is the capacity to reach a "large" set of final states within some characteristic time in a controllable way, i.e., within some desired tolerance. Note that the set of all reachable states, generated deterministically by considering all possible controller choice and possible terrain profiles over time, is in general substantially larger than the set of states one can deliberately arrive at reliability. This is because the full set of states is a result of terrain variability that is fixed by the environment and cannot be actively controlled, in addition to the active selection of high-level control at each step.

Of particular interest in this work is the ability to plan dynamic and agile maneuvers that may require multiple steps. For example, jumping a ditch may first requires some number of steps, to get up to speed. By mapping out all reachable states, given a set of controllers and finite bounds on the terrain variability, we provide one straight-forward path toward determining whether any particular goal state is reachable within n steps. This can be done by employing value iteration, with a reward given solely for arriving at the desired state. On deterministic terrain one can determine the set of initial states from which the model can arrive at a given goal in n steps, and this process can then be repeated with each state set (in turn) as the goal state, to map out the n-step reachable region. On stochastic terrain, the result instead becomes a probabilistic map, with a probability of success calculated each particular combination of initial and goal state.

It is relatively easy to vary controller parameters and generate a stable gait with a characteristic cost of transport, forward speed, and largest eigenvalue. Consider these three parameters to form a three-dimensional performance space, with achievable designs occupying a subset of this space. Several open directions of research are of interest for future work. For example, we wish to determine if bounds on the achievable design region can be determined or estimated in a systematic way. Also, we wish to evaluate whether the magnitude of the largest eigenvalue is an effective predictor of long-term stability and rough terrain performance, or if the mixing effects on variable terrain are too complex to allow such a metric of local stability to provide a useful characterization of the long-term, global performance and failure rates. Finally, work is currently underway to develop robust control for a more sophisticated, 3D model developed by Jason Pusey using LMS software. Preliminary data enable this model to bound for over a dozen steps. We aim to improve control through low-dimensional analysis of the step-to-step dynamics, expanding upon the approach outlined in this work.

Acknowledgment

This work was supported by the Army's 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. The authors would like to thank Dan Koditschek and Jeff Duperret for their technical insights and for openly sharing test data for the actual Canid robot. The authors finally wish to thanks Siemens for their generosity in providing a generous academic discount on licensed use of LMS software for Prof. Byl's research group at UC Santa Barbara, as part of this collaborative work with ARL.

REFERENCES

[1] Duperret, J. M., Kenneally, G. D., Pusey, J. L., and Koditschek, D. E., "Towards a comparative measure of legged agility," in [*Proc. Int. Symp. on Experimental Robotics (ISER 2014)*], (2015).

[2] Leeser, K., *Locomotion Experiments on a Planar Quadruped Robot with Articulated Spine*, Master's thesis, MIT (1996).

[3] Haynes, G. C., Pusey, J., Knopf, R., Johnson, A. M., and Koditschek, D. E., "Laboratory on legs: an architecture for adjustable morphology with legged robots," in [*Unmanned Systems Technology XIV*], Karlsen, R. E., Gage, D. W., Shoemaker, C. M., and Gerhart, G. R., eds., **8387**(1), 83870W, SPIE (2012).

[4] Haueisen, B., *Investigation of an Articulated Spine in a Quadruped Robotic System*, PhD thesis, The University of Michigan (2011).

[5] Culha, U., *An Actuated Flexible Spinal Mechanism for a Bounding Quadrupedal Robot*, Master's thesis, Bilkent University (2012).

[6] Poulakakis, I., Papadopoulos, E., and Buehler, M., "On the stability of the passive dynamics of quadrupedal running with a bounding gait," *The International Journal of Robotics Research* **25**(7), 669–687 (2006).

[7] Cao, Q. and Poulakakis, I., "Passive quadrupedal bounding with a segmented flexible torso," in [*IROS*], (2012).

[8] Saglam, C. O. and Byl, K., "Robust Policies via Meshing for Metastable Rough Terrain Walking," in [*Proceedings of Robotics: Science and Systems*], (2014).

[9] Saglam, C. O. and Byl, K., "Quantifying the Trade-Offs Between Stability versus Energy Use for Underactuated Biped Walking," in [*IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*], (2014).

[10] Saglam, C. O. and Byl, K., "Stability and gait transition of the five-link biped on stochastically rough terrain using a discrete set of sliding mode controllers," in [*IEEE International Conference on Robotics and Automation (ICRA)*], 5675–5682 (May 2013).

[11] Byl, K. and Tedrake, R., "Metastable Walking Machines," *The International Journal of Robotics Research* **28**, 1040–1064 (Aug. 2009).