Switching Policies for Metastable Walking

Cenk Oguz Saglam and Katie Byl

Abstract—In this paper, we study the underactuated five-link biped walking on stochastically rough terrain. We propose a simple and powerful Sliding Mode Control scheme. By taking Poincaré sections just before the impact, we accurately represent ten dimensional system dynamics of metastable walking as a Markov process. By switching between two qualitatively different controllers, we show that the number of steps before failure can be increased by more than 10 million times compared to using either one of the controllers only. To achieve this, only the current state and approximate terrain slope for a one step lookahead on geometrically rough terrain is needed. The analysis techniques in this paper are also designed for future application to a range of other simulated or experimental walkers.

I. INTRODUCTION

Bipedal locomotion control is essential for humanoid robots to operate effectively in real-world environments. Research has produced a variety of capable legged machines to date. However, the control problem for human-like walking stands as a big challenge. This is not a result of lack of interest or attention.

Since McGeer introduced the passive walkers to the robotics world [1], researchers have demonstrated a range of powered walkers based on exploiting natural dynamics [2], and the approach has led to record-breaking performance in walking with only onboard power (energetically autonomous) [3]. However, a focus on energy efficiency has also resulting in designs that are sensitive to perturbations, thus performing poorly on rough terrain.

An alternative direction in biped locomotion research using the Zero Moment Point (ZMP) concept has become extremely popular [4]. Unlike passive-based approaches, which exploit an underactuated ankle, classic ZMP techniques rely on an actuated ankle, which in turn requires control to keep the center of pressure inside the support polygon of the feet. Here, ensuring that a ZMP constraint is not violated makes control design and analyses challenging for real-world environments, where perturbations due to rough terrain occur at every step.

Our work joins a body of work focusing on the control aspects of walking with an unactuated ankle. For example, inverted pendulum similarity has been widely used [5], and the promising Hybrid Zero Dynamics (HZD) has been proposed [6], to name some major studies. However, these studies often concentrated on constant slopes or flat ground, whereas the obvious advantage of walking over use of rolling of wheels is on rough terrain with discontinuities, such as rocky obstacles, ditches, or stairs. In order to make use of this advantage, stochastic terrains should be considered [7], [8], [9], [10]. On stochastic terrains, instead of adopting a black-white stability notion, with absolute guarantees the system will never fall down, considering metastability is much more useful. The mean first passage time (MFPT), also commonly called the Mean Time To Failure, gives the expected number of steps before falling. Thus, the higher MFPT is, the more stable the robot is, and control can then be optimized in terms of minimizing the frequency of falls.

What is very intuitive but has lacked sufficient attention is that humans do not walk the same way on every ground type. They modify their walk depending on various conditions, such as whether the surface is pavement or clay, whether the ground is triangular (slopes), or rectangular (stairs), whether it’s uphill or downhill, and whether there are obstacles on the way, just to name some. It is not possible or necessary to design a controller for all cases. However, even partly addressing the problem of using limited lookahead information on terrain can provide dramatic improvements in MFPT in the rough terrain examples we have studied.

The methods and ideas presented in this paper can be applied to any walker and any controller set. To illustrate our results, we adopted the five-link biped based on the RABBIT [6] because of its popularity and accessibility. We designed and used only two controllers to show the clear advantages of switching and to present the methodology most clearly, with a simple case example. To study the metastable dynamics and calculate MFPTs, we created a mesh that captures the step-to-step dynamics of walking. The switching policies we proposed increase stability dramatically. Although switching with the existing controllers is extremely useful on its own, we also anticipate that our analysis illuminates a path toward designing new controllers, in terms of the local, terrain-dependent stability characteristics each of a family of controllers should demonstrate to enable walking on across a broader range of terrain variability.

The rest of the paper is organized as follows. It first defines the necessary terminology and studies the dynamics. Secondly, it proposes a simple control scheme to easily get qualitatively different controllers. Thirdly, it explains the meshing procedure and presents pseudocode to encourage researchers to adopt our techniques toward the problem of more realistic walking. Then the paper illustrates the success of our mesh capturing the dynamics by sample cases.
Finally, it talks about switching policies depending on the information available and how to obtain the optimal policy. The results reveal that by using two controllers, the robot can walk much longer on a much wider range of slopes.

II. MATHEMATICAL MODEL

As the name dictates, the five-link biped consists of five-links connected by four joints. We assume the links are rigid and all internal joints are actuated, while the pivot contact with the ground is unactuated. We consider only the planar and all internal joints are actuated, while the pivot contact links connected by four joints. We assume the links are rigid with the ground. These cases are named single support and double support phases, respectively. A single and a double support phase form a step. The single support phase has instantaneous impact event. The impact event, denoted as presented in [11] except the following. The coefficient of ground friction is assumed to be $\mu_s = 0.6$, and the torque saturation limit is chosen to be 50 Nm, which corresponds to 1 Nm on the motor side due to the gear ratio of 50.

$$\Delta$$

is an instantaneous impact event. The impact event, denoted to equivalently write

$$\Delta$$

is obtained by the conservation of energy and the principle of virtual work [6], [12]. Here we just note that the instantaneous changes are only assumed to be in velocities, and the angles undergo a relatively simple mapping.

$$\begin{bmatrix} q_1^+ & q_2^+ & q_3^+ & q_4^+ & q_5^+ \end{bmatrix} = \begin{bmatrix} q_2^- & q_1^- & q_4^- & q_3^- & q_5^- \end{bmatrix}$$

This mapping is actually a relabeling and lets us always call the leg in contact with the ground the “stance leg”. The other leg is the “swing leg” as shown in Figure 1. The impact event occurs when the swing foot touches the ground. In order for the impact model to be valid, we need the former stance leg to lift from ground with no further interaction with the ground. For a step to be successful, the swing foot needs to be in front of the stance foot (walking “forward”) at the impact. Otherwise, or if any other part of the robot than the foot touches the ground, we say the conditions are violated. Finally, the force on the tip of the stance leg during the swing phase, and the force on the swing tip at the impact event must satisfy the following equation.

$$F_{\text{friction}} = F_{\text{normal}} \mu_s > |F_{\text{transversal}}|$$

We note all of the above-mentioned conditions as “validity conditions”. When any of those are violated, we say the robot has failed. Fig. 2 depicts this and the hybrid (continuous and discontinuous) dynamics of walking.

As the figure illustrates, we assume the robot cannot continue walking after falling, so that failure is the absorbing state. For all but trivial cases of variability in terrain, as we will see in Section V, the robot will eventually fall, although failure rates can still be quite rare. Note that the latter also holds for humans.

III. CONTROL

The most important limitation to control is underactuation, which is a result of the point-foot model. There are five angles but four motors. It is a straightforward idea to select four linear combinations of these five angles to control. In this paper we chose the angles to be controlled as

$$q_c := [\theta_2 \ q_3 \ q_4 \ \theta_5]^T = [q_2 + q_5 \ q_3 \ q_4 \ q_5]^T.$$
Note that $\theta_2 = q_2 + q_5$ and $\theta_3 = q_5$ are absolute angles, whereas $q_3$ and $q_4$ are relative angles. This choice is a result of our experiments with a three-link walker, which we controlled using its absolute angles of the swing leg and the torso. We decided to use the additional two actuations in the five-link biped to control the relative angles of the tips. Selecting the control input $u$ as

$$u = (SD^{-1}B)^{-1}(v + SD^{-1}(Cq + G)),$$

where $S = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$, (7)

leads to following simple structure

$$\dot{q}_c = v.$$ (8)

There can be different approaches to determine $v$. For finite time convergence, we used the sliding mode controller (SMC) scheme as explained in [13] and outlined next. The error is defined as

$$e = q_c - q_c^{ref}.$$ (9)

And the generalized error $\sigma$ is given by

$$\sigma_i = \dot{e}_i + \frac{e_i}{\tau_i}, \quad i = \{1, 2, 3, 4\}.$$ (10)

We then obtain $v$ in (7) as

$$v_i = -k_i|\sigma_i|^{2\eta_i-1}\text{sign}(\sigma_i), \quad i = \{1, 2, 3, 4\}.$$ (11)

Table I lists the controller parameters used.

<table>
<thead>
<tr>
<th>$i$</th>
<th>$\tau$</th>
<th>$k$</th>
<th>$\alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/10</td>
<td>50</td>
<td>0.7</td>
</tr>
<tr>
<td>2</td>
<td>1/10</td>
<td>100</td>
<td>0.7</td>
</tr>
<tr>
<td>3</td>
<td>1/20</td>
<td>75</td>
<td>0.7</td>
</tr>
<tr>
<td>4</td>
<td>1/5</td>
<td>10</td>
<td>0.7</td>
</tr>
</tbody>
</table>

In this paper, we will use two controllers, named $\zeta_1$ and $\zeta_2$. Thus, the set of controllers $Z$ is of the following form:

$$Z = \{\zeta_1, \zeta_2\}.$$ (12)

Both controllers use the same control parameters, but they have different reference sets as shown in Table II.

<table>
<thead>
<tr>
<th>Controller</th>
<th>$\theta_2^{ref1}$</th>
<th>$\theta_2^{ref2}$</th>
<th>$\theta_3^{ref1}$</th>
<th>$\theta_3^{ref2}$</th>
<th>$\theta_4^{ref1}$</th>
<th>$\theta_4^{ref2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\zeta_1$</td>
<td>225</td>
<td>204</td>
<td>0</td>
<td>-60</td>
<td>-21</td>
<td>0</td>
</tr>
<tr>
<td>$\zeta_2$</td>
<td>230</td>
<td>210</td>
<td>0</td>
<td>-45</td>
<td>-25</td>
<td>-15</td>
</tr>
</tbody>
</table>

$q_c^{ref}$ in (9) is obtained from Table II using

$$q_c^{ref} = [\theta_2^{ref}, \theta_4^{ref}]^T,$$

where $(\theta_2^{ref1}, q_4^{ref1}) = \left\{ \begin{array}{ll} (\theta_2^{ref2}, q_4^{ref2}) & \text{condition} \\ (\theta_2^{ref2}, q_4^{ref2}) & \text{otherwise.} \end{array} \right.$ (13)

We selected the condition in (13) as "if $\theta_2 \leq \pi"$ to reduce the risk of the stance feet slipping, by increasing $F_{\text{normal}}$.

We note that the two controllers have common qualities. They are verified to exhibit stable walking on flat ground by simulation, and they satisfy the following:

$$q_3^{ref} = 0$$

$$q_4^{ref1} < q_4^{ref2} < 0$$

$$\theta_2^{ref1} > \theta_2^{ref2} > \pi.$$ (14)

The first two lines are for ground clearance, and the last one is especially for the friction requirements. It is intuitive to verify these are anthropomorphic choices. However, the two controllers also have qualitatively different behaviors. $\zeta_2$ leans the torso relatively more forward, which lets the robot add energy more effectively and walk more easily on high slopes. $\zeta_1$ makes the robot take smaller steps. The advantage of having different controllers will be become clear in section V.

IV. DISCRETE POINCARÉ MAP

Next, we create a discrete Poincaré map using states just before the impact, i.e. $x^-$. From this point on, we will refer to them just as states. We first define the distance of a state $x$ from the set of states $P$.

$$d(x, P) := \min_{p \in P} \left\{ \sum_{i=1}^{10} \left( \frac{x_i - p_i}{r_i} \right)^2 \right\}.$$ (15)

We will refer to the vector containing $r_i$ values as weight vector $R$. In this paper, we will assume the ground profile is triangular, and that the slope may change only after an impact. We select slope set $S$ to be

$$S = \{-10 + k/4 \mid k \in \mathbb{Z}, \quad 0 \leq k \leq 80\}.$$ (16)

For each controller from controller set $Z$ and each slope from slope set $S$, we tried walking the biped 100 steps. If the robot did not fall, we added the final state to the set of initial states $P_f$, which in the end had 65 points. We then use Algorithm 1 to obtain a larger set of states which aims to capture the dynamics of the robot when the slope and controller may change at each step. While doing so, we also create the corresponding transition matrix. The transition matrix is a map: Given controller, slope, and initial condition, it tells where the robot will go in the configuration space. During the meshing, extra values such as the Cost of Transport, step width, or controller history can be also stored.

We first mesh using $r_i = 1$ in (15). Then we repeat meshing using the standard deviation of the resulting mesh as $R$ to get our final mesh, $P_f$, which consisted of 139,241 states.
Algorithm 1 Meshing algorithm

**Input:** Initial set of states $P_i$, Slope set $S$, Controller set $Z$, weight vector $R$ and maximum distance $d_{\text{max}}$

**Output:** Final set of states $P_f$ and Deterministic state-transition map $T_d$

1. $Q \leftarrow P_i$
2. $P_f \leftarrow P_i$
3. define failure state $x_1$
4. while $Q$ is non-empty do
5. $Q_2 \leftarrow Q$
6. empty $Q$
7. for each state in $q \in Q_2$ do
8. for each slope $x \in S$ do
9. for each controller $\zeta \in Z$ do
10. simulate to get the corresponding state $x$
11. if fell then
12. $x \leftarrow x_1$
13. else
14. if $d(x,P_f) < d_{\text{max}}$ then
15. add $x$ to $Q$
16. else
17. add $x$ to $P_f$
18. end if
19. end if
20. store in $T_d$ that robot went to $x$
21. end for
22. end for
23. end for
24. end while
25. return $P_f$ and $T_d$

The methodology explained here can be used with different initial sets, controller sets, ground sets, and distance definitions. We verify the success of the mesh covering the step-to-step dynamics with Table III. The “Simulation” and “Estimation1” columns are obtained by simulating the full model and using the transition matrix as explained in [11], respectively. Each element in these two columns has a relative error of 0.001. “Estimation2” will be explained in the following section. In the light of this table, we conclude that the mesh covers the dynamics of the full model adequately.

**Table III**

<table>
<thead>
<tr>
<th>Controller</th>
<th>Slope</th>
<th>Simulation</th>
<th>Estimation1</th>
<th>Estimation2</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.3</td>
<td>(20.9,20.5)</td>
<td>(21.4,20.9)</td>
<td>21.5</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.2</td>
<td>(140,141.8)</td>
<td>(141.7,139.6)</td>
<td>139.2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.15</td>
<td>(515,1,509)</td>
<td>(471,461.1)</td>
<td>468.4</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.3</td>
<td>(15,1,14.5)</td>
<td>(15.5,15)</td>
<td>15.4</td>
<td></td>
</tr>
<tr>
<td>Random</td>
<td>-1.2</td>
<td>(5.1,3.9)</td>
<td>(5.4)</td>
<td>4.6</td>
<td></td>
</tr>
</tbody>
</table>

V. SWITCHING

A. Mean First Passage Time

The mean first-passage time (MFPT) is a useful metric to quantify the stability of metastable (i.e., “rarely falling”) walking machines [14]. In this paper, we extend the results for switching. The next state of the robot, $x[n+1]$, is a function of the current state $x[n]$, slope $\gamma[n]$, and controller $\zeta[n]$, i.e.,

$$x[n+1] = h(x[n], \gamma[n], \zeta[n])$$  \hspace{1cm} (17)

We define our control policy as the set of rules determining $\zeta[n]$. We will consider policies that are functions of the current state, and/or information about the ground. In general,

$$\zeta[n] = \pi_{\text{policy}}(x[n], \gamma_{\text{all}}),$$  \hspace{1cm} (18)

where $\gamma_{\text{all}}$ gives probabilistic information on all ground slopes. When a policy is applied, (17) becomes

$$x[n+1] = h_{\text{policy}}(x[n], \gamma_{\text{all}}).$$  \hspace{1cm} (19)

Using the deterministic state-transition map $T_d$ for state, action, and upcoming slope, and given a policy and the terrain distribution for $\gamma_{\text{all}}$, we obtain the stochastic state-transition matrix $T_s$ defined by

$$T_s(ij) = Pr(X[n+1] = x_j \mid X[n] = x_i).$$  \hspace{1cm} (20)

Noting that $x_1$ is the failure state and using the matrix $T_s$ we write the metastable state distribution defined by

$$\phi_i = \lim_{n \to \infty} Pr(X[n] = x_i \mid X[0] \neq x_1).$$  \hspace{1cm} (21)

We then get the mean passage time vector

$$\Psi_i = \begin{cases} 0, & \text{if } i = 1 \\ 1 + \sum_j T_s(ij) \Psi_j, & \text{otherwise}. \end{cases}$$  \hspace{1cm} (22)

Then the (system-wide) MFPT, $\Psi$, equals

$$\Psi = \sum_i \psi_i \phi_i,$$  \hspace{1cm} (23)

and is approximated to be

$$\Psi \approx \frac{1}{1 - \lambda_2}$$  \hspace{1cm} (24)

where $\lambda_2$ is the second largest eigenvalue of $T_s$ [14]. Approximation in (24) is the procedure for $\text{Estimation}_2$ of Table III, and for the graphs we will be presenting in the following sections.

B. The Idea of Switching

Consider the toy example of Figure 3. Say a robot is at state A, there are two possible ground profiles ahead, and for the graphs we will be presenting in the following sections.

$$M = \{-8 + k/4 \mid k \in \mathbb{Z}, 0 \leq k \leq 64\}.$$  \hspace{1cm} (25)

Different standard deviations or mean sets are possible, but these values are sufficient to show advantages in switching.
**C. Simple Switching**

The simplest policy is to use just one controller, i.e.,

\[
\zeta[n] = \zeta_i
\]  

(26)

Much research on bipedal walking has concentrated on finding one optimal \( \zeta_i \), agnostic to local upcoming terrain information (i.e., "blind but robust" walking). However, a particular \( \zeta_i \) might be optimal only for a small region in the slope set, meaning local terrain features can be better negotiated through switching control. In addition, the optimal \( \zeta_i \) is different for different cost definitions.

The simplest switching policy depends on one-step lookahead, i.e., \( \zeta[n] \) only.

\[
\zeta[n] = \pi_{\text{simple}, n} \]  

(27)

Figure 4 shows MFPT data for several control policies on terrain with mean as given on the x-axis label and \( \sigma = 1 \) degree (as earlier discussed). Looking at this data, we see that controller one (\( \zeta_1 \)) works better for mean smaller than 0.218. Thus, we first consider the following, simple policy:

\[
\zeta_s(\gamma')[n] = \begin{cases} 
\zeta_1, & \text{if } \gamma[n] < \gamma' \\
\zeta_2, & \text{otherwise.} 
\end{cases}
\]  

(28)

As expected, the robot works at least as good as any fixed controller, set for any mean. In addition, the robot is more stable between the peaks of the two fixed controllers.

**D. Avoiding Guaranteed Failures**

We add another simple idea to the simple switching: if \( \zeta[n] = \zeta_s(\gamma')[n] \) will cause an immediate failure for the current state and the following state, use the other controller:

\[
\zeta_o(\gamma')[n] = \begin{cases} 
\zeta_1, & \text{if } \gamma[n] < \gamma', \text{ and } f(x[n], \gamma[n], \zeta_1) \neq x_1 \\
\zeta_2, & \text{if } \gamma[n] < \gamma', \text{ and } f(x[n], \gamma[n], \zeta_2) = x_1 \\
\zeta_1, & \text{if } \gamma[n] \geq \gamma', \text{ and } f(x[n], \gamma[n], \zeta_1) \neq x_1 \\
\zeta_2, & \text{if } \gamma[n] \geq \gamma', \text{ and } f(x[n], \gamma[n], \zeta_2) = x_1, 
\end{cases}
\]  

(31)

where \( x_1 \) is the failure state. Compared to simple switching, note that the policy is now also a function of the current state \( x[n] \).

\[
\zeta[n] = \pi_{\text{avoid}}(\gamma[n], x[n])
\]  

(32)

\( \zeta_{\text{av}} \) is obtained from \( \zeta_s(\gamma')[n] \) by using the \( \gamma' \) that maximizes the MFPT for given \( \gamma_{\mu} \).

\[
\zeta[n] = \pi_{\text{avoid}}(\gamma[n], x[n], \gamma_{\mu})
\]  

(33)
This paper studies underactuated bipedal locomotion control with partial terrain information. As the model, a five-link walker is adopted. We present a simple and powerful control with partial terrain information. As the model, a five-link walker is adopted. We present a simple and powerful control scheme. By changing just a few parameters, we are able to obtain qualitatively different controllers with relative ease.

We then aim to find the optimal switching given the state, mean of the long-term ground slope $\gamma_\mu$, and one-step look-ahead of slope $\gamma[n]$. The intuitive idea is to decide on the controller by calculating $\Psi$ (MFPT) in (24) for every possible $T_i$ defined in (20). In the toy example of Figure 3, there were 4 choices for each point. If there are, say 10 points in the mesh, there are more than a million different policies possible. In general, number of possible policies is $d^{abc}$, where $a$, $b$, and $c$ are the sizes of the controller set, mean set, and transition matrix (number of points in the mesh) respectively. The computational complexity makes this idea infeasible. Instead, we use a numerical approach to find the optimal policy: Bellman’s Principle of Optimality [15]. We first assign initial conditions for approximate mean passage time vector.

$$v_i[0] = \begin{cases} 0, & \text{if } i = 1 \\ 1, & \text{otherwise} \end{cases}$$

(35)

Then write the recursive algorithm for $i > 1$

$$v_i[n+1] = 1 + \alpha \sum_j Pr(\gamma[n] = \gamma_j) \max_k (h(x_i, \gamma_j, \zeta_k))$$

(36)

where $\gamma_j \in S$, $0 < \alpha < 1$ is the discount factor, and $h$ is as defined in (17). Note that the optimization is done for each $\mu \in M$. We iterated for one thousand steps with $\alpha = 0.95$, to achieve policy convergence. We call the resulting control $\zeta_{opt}$ and present the results in Figure 7. In this Figure we also plot $\zeta_{opt}(0)$, which is the policy optimized only for $\mu = 0$.

Looking at the graph and noting the logarithmic scale, we immediately observe there is an extreme improvement. To illustrate, we define “safe mean range” (SMR) to be the mean range over which the robot has MFPT more than $10^6$ and present Table IV.

<table>
<thead>
<tr>
<th>$\zeta[n]$</th>
<th>MFPT for $\mu = 0$</th>
<th>SMR</th>
<th>SMR Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\zeta_1$</td>
<td>9.857x10^5</td>
<td>(-2.107, -0.002)</td>
<td>2.105</td>
</tr>
<tr>
<td>$\zeta_2$</td>
<td>9.548x10^4</td>
<td>(0.657, 1.939)</td>
<td>1.282</td>
</tr>
<tr>
<td>$\zeta_{opt}$</td>
<td>5.118x10^13</td>
<td>(-2.107, 1.960)</td>
<td>4.067</td>
</tr>
<tr>
<td>$\zeta_{opt}(0)$</td>
<td>5.118x10^13</td>
<td>(-2.106, 1.945)</td>
<td>4.051</td>
</tr>
</tbody>
</table>

We note that knowing the long-term mean of the ground might or might not be easily done in practice but that $\zeta_{opt}(0)$ does almost as good as $\zeta_{opt}$, so we suggest the following policy:

$$\zeta[n] = \pi_{suggested}(x[n], \gamma[n]) = \pi_{optimal}(x[n], \gamma[n], \gamma_{peak}).$$

(37)

where $\gamma_{peak}$ is the mean at which the MFPT of $\zeta_{opt}$ is maximum.

VI. CONCLUSIONS

Figure 6 depicts $\zeta_{opt}(\gamma^*)$ with the optimal $\gamma^*$. Figure 6 reveals two important results. First, the MFPT greatly increases just by selecting the other controller, if the baseline switching policy would otherwise fail in the next, immediate step. Secondly, knowing the mean does not help as much as it did in Figure 5.

E. Optimal Switching

We then aim to find the optimal switching given the state, mean of the long-term ground slope $\gamma_\mu$, and one-step look-ahead of slope $\gamma[n]$. The optimization goal is to maximize the MFPT. Of particular interest are the extents to which the optimal policy improves performance over either the simple switching strategies previously presented or a “robust but blind” (non-switching) strategy, more generally.

The intuitive idea is to decide on the controller by calculating $\Psi$ (MFPT) in (24) for every possible $T_i$ defined in (20). In the toy example of Figure 3, there were 4 choices for each point. If there are, say 10 points in the mesh, there are more than a million different policies possible. In general, number of possible policies is $d^{abc}$, where $a$, $b$, and $c$ are the sizes of the controller set, mean set, and transition matrix (number of points in the mesh) respectively. The computational complexity makes this idea infeasible. Instead, we use a numerical approach to find the optimal policy: Bellman’s Principle of Optimality [15]. We first assign initial conditions for approximate mean passage time vector.

$$v_i[0] = \begin{cases} 0, & \text{if } i = 1 \\ 1, & \text{otherwise} \end{cases}$$

(35)

Then write the recursive algorithm for $i > 1$

$$v_i[n+1] = 1 + \alpha \sum_j Pr(\gamma[n] = \gamma_j) \max_k (h(x_i, \gamma_j, \zeta_k))$$

(36)

where $\gamma_j \in S$, $0 < \alpha < 1$ is the discount factor, and $h$ is as defined in (17). Note that the optimization is done for each $\mu \in M$. We iterated for one thousand steps with $\alpha = 0.95$, to achieve policy convergence. We call the resulting control $\zeta_{opt}$ and present the results in Figure 7. In this Figure we also plot $\zeta_{opt}(0)$, which is the policy optimized only for $\mu = 0$.
The paper then explains a meshing technique that dramatically simplifies the analysis of the stability of walking on stochastically rough terrain. This meshing importantly exploits the fact that impact states lie essential on a 2D manifold within a higher dimensional state space, which helps us study the problem more easily. We also observe that stability for walkers is not a 1 or 0 (yes/no) result; statistics of failures, when rare (e.g., 10^{-8} events), still result in exceptional long-term behavior. When \( \mu = 0 \), both \( \zeta_1 \) and \( \zeta_2 \) are numerically stable for \( \sigma = 0 \), but failure events are eventually guaranteed for any unbounded noise. We conclude that we need to consider probabilistic metrics for walking, because every non-trivial ground has a deviation, and present optimization results for a MFPT metric.

Our results in Section V reveal that even simple switching helps by increasing the safe mean range. We also note that, while long-term mean information is very helpful for simple switching, it is not so for the optimal switching policy. Not needing the mean information is a highly desired result, since knowing the mean of the ground slope ahead is often not as practical as knowing the state or the slope ahead for one step. Our main conclusion is that switching between two qualitatively different controllers improves the stability dramatically: the range of mean locomotion-capable terrain slopes is much wider and the peak (in MFPT) is much higher.

### VII. DISCUSSIONS AND FUTURE WORKS

In this paper, we assume no noise on the one-step look ahead to the ground slope. Our initial results indicate that the optimal policy may not be particularly robust (to noise in the one-step lookahead). Finding an optimally robust controller policy is a central focus for our future work.

After finding a robust policy, the next main goal of the writers is to understand how to design a third controller to further improve the MFPT of the optimal switching policy using all three controllers. We are also interested how much a two-, three- and infinite-step lookahead increases the stability, beyond one-step knowledge.

We plan to improve the meshing technique in several aspects. We believe we are using a denser mesh than necessary. By improving our weighting in distance calculation, we are hoping to capture the dynamics adequately with a smaller mesh. Also, the mesh we present covers a region in 10-dimensional space from which the robot will not escape (except by falling) after entering the region. We hope to cover all initial conditions that leads to the same region.

In this paper we use same controller parameters for both controllers. We are interested in finding a mapping from a given reference set to the optimal controller parameter set.

The ground profile in this paper was triangular, each step experienced a slope. In future, we plan to use more realistic ground types. The challenging part is meshing. We also want to consider cases when there are bad spots on terrain that we do not want robot to step on.

Finally, we are interested in coming up with more complicated cost functions. In this paper we optimized for MFPT only, but we may also consider, for example, the Cost of Transport (COT), and speed.

### REFERENCES