Optimizing Robust Limit Cycles for Legged Locomotion on Unknown Terrain

Hongkai Dai and Russ Tedrake

Abstract— While legged animals are adept at traversing rough landscapes, it remains a very challenging task for a legged robot to negotiate unknown terrain. Control systems for legged robots are plagued by dynamic constraints from underactuation, actuator power limits, and frictional ground contact; rather than relying purely on disturbance rejection, considerable advantage can be obtained by planning nominal trajectories which are more easily stabilized. In this paper, we present an approach for designing nominal periodic trajectories for legged robots that maximize a measure of robustness against uncertainty in the geometry of the terrain. We propose a direct collocation method which solves simultaneously for a nominal periodic control input, for many possible one-step solution trajectories (using ground profiles drawn from a distribution over terrain), and for the periodic solution to a jump Riccati equation which provides an expected infinite-horizon cost-to-go for each of these samples. We demonstrate that this trajectory optimization scheme can recover the known deadbeat open-loop control solution for the Spring Loaded Inverted Pendulum (SLIP) on unknown terrain. Moreover, we demonstrate that it generalizes to other models like the bipedal compass gait walker, resulting in a dramatic increase in the number of steps taken over moderate terrain when compared against a limit cycle optimized for efficiency only.

I. INTRODUCTION

Legged animals exhibit an extraordinary ability to adapt to variable terrain elevation during walking, inspiring engineers to bring about legged robots that are comparably robust on unknown terrain. A wide variety of legged robots capable of overcoming rough terrain have been built to examine diverse locomotion strategies. Usually, the control strategy for these robots have been tailored to the specific robot model [20], [18], [2], [6]. This work presents a new control strategy which applicable to a more general robot model, with simulation validation on two canonical robot models. Legged robots are subjected to many sources of uncertainty, including for instance unmodeled disturbances like a push from a human or a gust of wind, or parameteric uncertainty from unmodeled frictional forces. Among all possible sources of uncertainty for a walking robot, terrain uncertainty is of particular interest, as it pertains directly to the hybrid dynamical system structure which makes walking robots unique relative to many other types of robots. Furthermore, changes in the schedule or location of a ground impact can have a large deviation from a nominal desired walking pattern. Underactuated systems like legged robots often do not possess enough control authority to cancel out the state
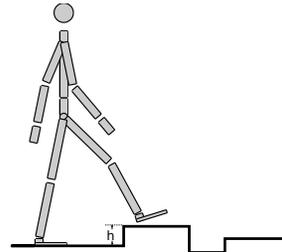
Fig. 1: A planar humanoid walks on the rough terrain, with the incoming terrain height equals to h. Although the terrain is represented by flat stairs, it actually incorporates most of the terrain profiles, since by ignoring ground clearance for the swing foot, it only matters with the difference in height between the stance foot and the swing foot to to be touching the ground.

error arising from unscheduled ground impacts; these can easily cause the robot to fall down.

Formal approaches to control design for walking robots (e.g. [26], [12]) are often separated into (1) designing a nominal periodic motion/trajectory/gait and (2) (local) feedback stabilization of that gait. Robust design techniques can be injected into either of these pieces, but in practice we have found that the selection of a robust nominal can have a more dramatic effect on the overall robustness of the systems. In this paper we formulate the problem of designing a nominal gait which is intristically easy to stabilize, and provide a transcription of this problem to a nonlinear optimization problem.

In Section II we present the general scheme for optimizing the robust limit cycle. In Section III, we examine the method on both SLIP and a simple passive walker, the compass gait, and compare the performance of the robust limit cycles to the existing periodic orbits. In Section IV, we discuss the possible alternations to our approach. In the last section we draw the conclusion and briefly introduce our future work.

A. Related work

Humans and animals maintain periodic gaits while walking and running. Such periodic trajectories, called the limit cycles, are also discovered in various walking models, such as the stable apex-to-apex return map for Spring Loaded Inverted Pendulum (SLIP) model [21], and the passive limit cycle for the bipedal walker stepping down a shallow slope [14], [10]. Those periodic orbits attract special attention among robotics researchers to study their properties. Instead of constructing a disturbance rejection controller, Mombaur exhibits that by minimizing the spectral radius of the mon-

odromy matrix of the return map, the optimized limit cycle achieves improved stability without the aid of feedback controller [16]. The success of her work motivates our interest in optimizing the nominal limit cycle to be more robust to terrain changes, rather than purely relying on disturbance rejection.

Trajectory optimization can improve the intrinsic robustness of trajectories. By employing Differential Dynamic Programing, Morimoto designed a trajectory that is robust to disturbances through minimizing H_∞ cost iteratively, and simultaneously came up with a robust controller [17]. This is the most relevant approach to our work, although he mostly focuses on the disturbances in the continuous dynamics, while the terrain disturbances are not addressed explicitly in his scheme.

The trajectories robust to terrain changes have been discovered for the SLIP model. Ernst presented the deadbeat feedforward controller for such model through exhaustive simulations [6]. Such trajectory can completely reject unknown terrain elevations, and will achieve a perfect apex-to-apex return map. Although his approach is model-specific, the result motivates us to design a general framework of computing such robust limit cycles for the whole family of legged robots.

II. METHODOLOGY

A. Problem Formulation

As is commonplace in optimization-based trajectory design for legged robots (e.g., [26], [15], [23], [24], [19]), we describe the dynamics of a legged robot as a hybrid system, with rigid-body collisions between the robot and the environment modeled as impulsive transitions between hybrid modes. For simplicity we focus our presentation on the case where hybrid system has only one mode and a single transition function, which maps the prior-impact state to a new state in the same mode. Without much loss of generality, we describe the unknown terrain as drawn from a distribution parameterized one each footstep by a single random variable describing the height, h, of the terrain on the next step[3]. This random terrain height adds uncertainty to the hybrid guard which detects the collision event (e.g. between a foot and the ground), and to the hybrid transition function. The resulting dynamics are summarized as:

$$\dot{x} = f(x,u) \quad \text{if } \phi(x,u) > h \tag{1}$$
$$x^+ = \Delta(x^-,u^-) \quad \text{if } \phi(x^-,u^-) = h \tag{2}$$
$$x \in \mathscr{X} \quad u \in \mathscr{U} \tag{3}$$
$$h \text{ drawn from a random distribution} \tag{4}$$

where f is the continuous dynamics in the mode, and ϕ represents the height of the foot, and it is the guard function of the hybrid system. When the foot touches the terrain with height h, namely $\phi(x^-,u^-) = h$, the guard function is triggered, and the transition function Δ maps the prior-impact state x^- to a post-impact state x^+ in the same mode. $\mathscr{X}$
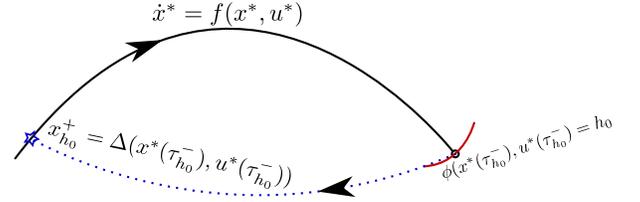
Fig. 2: Nominal trajectory on the nominal terrain height h_0
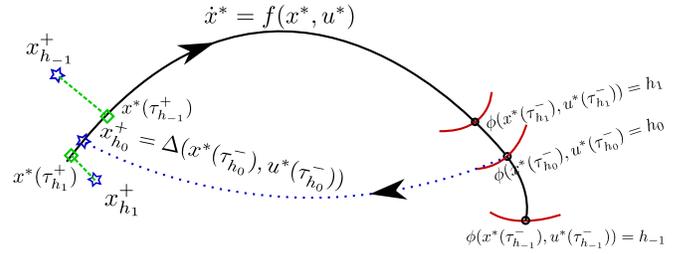
Fig. 3: Trajectory on terrain height h_{-1}, h_1

and $\mathscr{U}$ are the admissible sets of states and control actions respectively.

In the sequel, we will assume that the robot walks blindly on the rugged terrain, but has access to perfect state information. Therefore, the controller does not know the height of the terrain before making contact, but could infer it perfectly upon contact. The blindness assumption is valid for robots without terrain perception; for robots with a terrain sensor, the uncertainty description can be used to capture bounds on the error from a perceptual system.

B. Intuition

Our goal is to design a one-step limit cycle that can minimize the effect of terrain disturbances while traversing all possible terrain heights within a certain range. The propagation of the state x^* on such limit cycle is depicted in Fig 2 and 3 for three different terrain heights h_{-1}, h_0, h_1, with h_0 being the nominal terrain. Notice that in Fig 2, the post-impact state $x_{h_0}^+$ lands exactly on the nominal trajectory, due to the periodic orbit on the nominal terrain h_0. For the terrain height h_1 higher than the nominal height h_0, the nominal pre-impact phase $\tau_{h_1}^-$ is earlier than the nominal pre-impact phase $\tau_{h_0}^-$, and the post-impact state $x_{h_1}^+$ is projected to $x^*(\tau_{h_1}^+)$ on the nominal trajectory, and resets the tracked phase to $\tau_{h_1}^+$. A time varying controller will steer the post-impact state $x_{h_1}^+$ back to the nominal trajectory by tracking the phase starting at $\tau_{h_1}^+$. Similarly for the terrain height h_{-1} lower than the nominal terrain h_0, we extend the trajectory to $x^*(\tau_{h_{-1}}^-)$, and reset the post-impact phase to be $\tau_{h_{-1}}^+$.

One of reasons why a robot easily falls down after hitting unexpected terrain is that the post impact states are far

(a) A bad limit cycle   (b) A good limit cycle
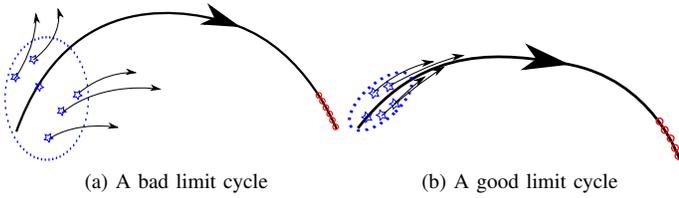
Fig. 4: Squeeze post-impact region

away from the nominal trajectory, leaving little hope for the controller to capture those post-impact states within the narrow regions of attraction; these post-impact states will diverge from the nominal trajectory. Intuitively, we want to squeeze the regions of the possible post-impact states, and constrain them to be in the vicinity of the nominal trajectory. Fig 4 illustrates this idea. Each post-impact states (blue stars) corresponds to a ground impact at a specific terrain height. When the regions of all possible post impacts (the blue circle) spans a large region away from the nominal cycle (blue ellipsoid), those post-impacts will not track the nominal cycle, and the robot will fall down. On the contrary, when the possible post-impact states are all close to the nominal cycle, they will be stabilized by the tracking controller. The location of the post-impact states are determined by the pre-impact states, and thus by the whole nominal trajectory. We note that the improvement from Fig 4a to 4b can also be accomplished by elegant controller design, but in practice we have found that the design of the nominal trajectory has a greater effect.

C. Trajectory Optimization

Based on the intuition discussed above, our goal is to squeeze the region of the post-impact states so as to make those states 'close' to the nominal limit cycle. Thus we need to define a metric quantifying the 'closeness' in the state space. A simple distance metric for 'closeness' is the (weighted) Euclidean distance between two points. But such metric performs poorly in the state space, as the system dynamics determines that it is almost impossible for the straight line connecting those two states to be a feasible path. On the other hand, the cost-to-go in optimal control theory provides a good notion to measure the difficulty of steering a state to the its goal, and it inherently indicates the long-term performance of the system, thus it is an ideal metric in the state space [4]. In the case of rough terrain locomotion, we expect it takes as little effort as possible to regulate the post-impact states back to the nominal limit cycle, so that the effect of terrain disturbances is alleviated. In our work, we minimize the expected cost-to-go to steer the post-impact states to the nominal trajectory in our trajectory optimization program.

Since the robot should maintain periodic walking on the nominal terrain h_0, the periodicity constraints is also imposed for the post-impact states $x_{h_0}^+$ and the cost-to-go function $J(x_{h_0}^+)$. Suppose function Π projects the post-impact state on the nominal trajectory and resets the tracked phase τ, the

trajectory optimization program is formulated as follows:

$$\min_{x^*(.),u^*(.),J(.)} E(J(x_h^+)) \qquad (5)$$

$$s.t \quad \dot{x}^*(\tau) = f(x^*(\tau), u^*(\tau)) \quad \text{if } \phi(x^*(\tau), u^*(\tau)) > h$$

$$\dot{\tau} = 1 \qquad\qquad\qquad \text{if } \phi(x^*(\tau), u^*(\tau)) > h$$

$$x_h^+ = \Delta(x^*(\tau_h^-), u^*(\tau_h^-)) \quad \text{if } \phi(x^*(\tau_h^-), u^*(\tau_h^-)) = h$$

$$\Pi(x_h^+, x^*(\tau_h^+)) = 0$$

$$x_{h_0}^+ = x^*(\tau_{h_0}^+)$$

J subject to the cost-to-go dynamics

$J(x_{h_0}^+)$ subject to the periodicity constraint

$x \in \mathcal{X} \quad u \in \mathcal{U}$

h drawn from a random distribution

For the optimization problem, there are two viable approaches to parameterize the decision variables [25]. The first one, the indirect method (for example, single shooting), only optimizes over the initial state and control actions. The explicit form of the cost function and constraints can be obtained by numerically integrating the dynamics function forward and computing the gradient of states w.r.t decision variables. Although straightforward, such a method suffers from the 'tail wagging the dog' effect, i.e, a small perturbation of the initial state and control action can produce large changes on the final states, resulting in a slow convergence rate [1]. The second approach, the direct method (for example, direct collocation), formulates both control and states as decision variables, with the dynamics functions to be the nonlinear constraints in the optimization program [5]. This approach is advantageous in convergence rate, but requires the optimization program to handle more variables and constraints. In our work, we adopt the second approach, and formulate states, control actions and cost-to-go matrices as decision variables.

The true optimal cost-to-go is prohibitively difficult to compute for a general nonlinear system, thus the infinite-horizon Linear Quadratic Regulator (LQR) cost is chosen as an approximation, namely, $J(x) = x'Sx$, with optimal cost-to-go matrix S to be the solution to the differential Riccati equation. LQR cost-to-go proves to be a valid and convenient metric in state space for expanding Rapidly-explored Random Tree (RRT) [9], and is also a good approximation of the true optimal cost-to-go for the nonlinear system. The physical interpretation of such metric is the optimal cost that the LQR will take to track the nominal trajectory, if the following terrains were nominal terrain and if the linearized dynamics is a valid approximation of the nonlinear system in the neighborhood around the nominal trajectory. Meanwhile, optimizing such cost function also provides a simple LQR controller for stabilizing the nominal trajectory, so that numeric integration error and external disturbances other than terrain changes can be rejected in the simulation.

The traditional manner of solving periodic LQR cost-to-go matrices is to integrate the differential Riccati equation backward along a given periodic trajectory in the state space until the solution converges. This approach is not viable in

our optimization program, as we do not know the periodic state-space trajectory beforehand, and we need to solve the periodic solution explicitly, rather than an approximated matrix converging to the real periodic solution. Instead, we parameterize the cost-to-go matrices as decision variables in our optimization program, and we treat the differential Riccati equation as dynamics constraints on those variables. Together with the constraints that the cost-to-go matrices at the tailing points of the limit cycle are consistent through the jump-Riccati equation, the periodic LQR cost-to-go matrices and the state/control sequences are simultaneously solved by the optimization program below

$$\min_{\substack{x^*(.),u^*(.),S^*(.),x_h^+ \\ \tau_h^-,\tau_{h_0}^-,\tau_h^+,\tau_{h_0}^+}} E[(x_h^+ - x^*(\tau_h^+))'S(\tau_h^+)(x_h^+ - x^*(\tau_h^+))] \quad (6)$$

$$\begin{aligned}
\text{s.t} \quad & \dot{x}^* = f(x^*,u^*) \quad \forall \tau < \tau_h^- \\
& \phi(x^*(\tau),u^*(\tau)) > h \quad \forall \tau < \tau_h^- \\
& x_h^+ = \Delta(x^*(\tau_h^-),u^*(\tau_h^-)) \\
& \phi(x^*(\tau_h^-),u^*(\tau_h^-)) = h \\
& -\dot{S}(\tau) = A(\tau)'S(\tau) + S(\tau)A(\tau) + Q \\
& \qquad\qquad - S(\tau)B(\tau)R^{-1}B(\tau)'S(\tau) \quad \forall \tau < \tau_{h_0}^- \\
& S(\tau_{h_0}^+) = T_x'S(\tau_{h_0}^-)T_x \\
& \Pi(x_h^+,x^*(\tau_h^+)) = 0 \\
& x_{h_0}^+ = x^*(\tau_{h_0}^+) \\
& h \sim \text{ a given distribution} \\
& x \in \mathscr{X} \quad u \in \mathscr{U}
\end{aligned}$$

The formulation above cannot be processed directly by the optimization solver, since it has infinite decision variables. The remedy is to sample the terrain height and discretize the state, control and cost-to-go matrix, and approximate the continuous differential equation by numerical approximation of the discretized values. We take K samples of the terrain height h with impulsive probability $pr(h = h_n) = p_n, n = 1,\ldots,K$, and discretize the state, control, and cost-to-go matrix tape. In the discretization of the tape, we break the time domain into N intervals with the interval duration between knot i and knot $i+1$ to be dt_i. We then sample the state, control and cost-to-go matrix trajectories at those knot points as x_i, u_i and S_i respectively. To rebuild continuous trajectories from those samples, we use cubic Hermite interpolation for states and cost-to-go matrices, and linear interpolation for control actions as in [11]. We prescribe some sampled knot points $\{x_{p_n}\}$ on the nominal trajectory that are projected by post-impact states $\{x^+(h_n)\}$ through function Π respectively. Due to the monotonicity relationship between the terrain height and the impact time, i.e., a given trajectory will hit the impact surface related to higher terrain prior to it hitting the impact surface with lower terrain, we can also prescribe a sequence of knot point x_{c_n} such that x_{c_n} is the prior-impact state for the terrain height

h_n. The optimization program is formulated as follows:

$$\min_{x_i,u_i,S_i,x_{h_n}^+,dt_i} \sum_n pr(h_n)\left[(x_{h_n}^+ - x_{p_n})'S_{p_n}(x_{h_n}^+ - x_{p_n})\right] \quad (7)$$

$$s.t \ x_i^c = (x_i + x_{i+1})/2 + \frac{dt_i}{8}[f(x_i,u_i) - f(x_{i+1},u_{i+1})] \quad (8)$$

$$u_i^c = (u_i + u_{i+1})/2 \quad (9)$$

$$\dot{x}_i^c = -3(x_i - x_{i+1})/(2dt_i) - [f(x_i,u_i) + f(x_{i+1},u_{i+1})]/4 \quad (10)$$

$$\dot{x}_i^c = f(x_i^c,u_i^c) \quad (11)$$

$$\dot{S}_i = -(A_i'S_i + S_iA_i - S_iB_iR^{-1}B_i'S_i + Q) \quad (12)$$

$$S_i^c = (S_i + S_{i+1})/2 + dt_i[\dot{S}_i - \dot{S}_{i+1}]/8 \quad (13)$$

$$\dot{S}_i^c = -3(S_i - S_{i+1})/(2dt_i) - [\dot{S}_i + \dot{S}_{i+1}]/4 \quad (14)$$

$$\dot{S}_i^c = -(A_i^{c'}S_i^c + S_i^cA_i^c - S_i^cB_i^cR^{-1}B_i^{c'}S_i^c + Q) \quad (15)$$

$$h_n = \phi(x_{c_n},u_{c_n}) \quad (16)$$

$$x_{h_n}^+ = \Delta(x_{c_n},u_{c_n},h_n) \quad (17)$$

$$\Pi(x_{h_n}^+,x_{p_n}) = 0 \quad (18)$$

$$x_{h_0}^+ = x_{p_0} \quad (19)$$

$$S_{p_0} = T_{c_0}'S_{c_0}T_{c_0} \quad (20)$$

$$x_i \in \mathscr{X} \quad u_i \in \mathscr{U} \quad (21)$$

Here x^c,u^c,S^c are the states, control actions and cost-to-go matrices at the collocation point. A_i,B_i,A_i^c,B_i^c are the Jacobians of dynamics function f w.r.t the sampled state x_i, the sampled control u_i, the collocated state x_i^c and the collocated control u_i^c respectively. T_{c_0} is the Jacobian of transition function Δ with respect to the pre-impact state x_{c_0} that triggers the ground collision guard for nominal terrain h_0. Q and R are the fixed instantaneous cost matrices for state error and control actions respectively.

Equations 8, 9, 10 and 11 altogether represent the state dynamics constraints on the collocation point x^c,u^c. Equation 12 is the explicit form of differential Riccati equation. Equations 13, 14 and 15 represent the cost-to-go dynamics constraints on the collocation point x^c,u^c,S^c. Equations 19 and 20 represent the periodicity constraints on states and cost-to-go matrices respectively.

To improve the speed of the optimization program, the gradients of the cost and constraints are computed explicitly. Note that the second order derivatives of the dynamics must be available for calculating the gradients of the constraints involving the cost-to-go matrices. The nonlinear optimization program (NLP) can be solved by software package like SNOPT [8], which handles this NLP efficiently due to the sparsity of the Jacobian matrix.

III. RESULTS

A. Spring Loaded Inverted Pendulum

The Spring Loaded Inverted Pendulum (SLIP) model, as shown in Fig 5, consists of a point mass attached to a massless spring. It is widely known that appropriate leg angle of attack and/or spring stiffness can achieve a perfect apex-to-apex return map on the flat terrain. On rough terrain, by
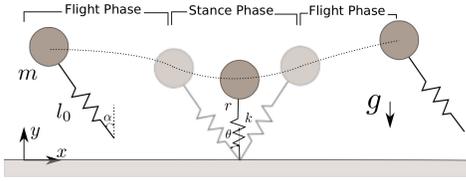
Fig. 5: Spring Loaded Inverted Pendulum (SLIP) [7], α in the flight phase is the actuated angle of attack.
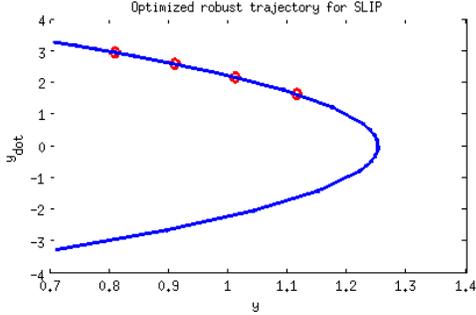
Fig. 6: Phase plot of SLIP in the flight mode

Fig. 7: Robust leg angle sequence computed from optimization versus deadbeat controller.

Fig. 8: Compass Gait Walker [24]

exhaustive simulation, Ernst finds the deadbeat controller encoding a time law of leg angle and the spring stiffness so as to achieve a perfect return map [6]. We show their model-specific approach can merge into our more general scheme.

The SLIP model has two modes, the flight mode and the stance mode. We suppose the leg angle is actuated during flight phase. The post-impact (post-take-off) state tracks the state on the nominal trajectory with the same vertical position. It is worth noting that the mode transition function Δ which maps the pre-impact (pre-landing) state to the post-impact (post-take-off) state is not instantaneous, but rather governed by the dynamics in the stance phase. Such mapping, together with its first and second order derivatives, are derived through adjoint method as follows [24]. We denote the mapping from pre-landing to post-landing to be Δ_1 and the mapping from pre-take-off to post-take-off to be Δ_2, the continuous dynamics in stance phase to be f_2, pre-landing state to be x_l^-, pre-landing angle of attack to be u^-, and $\alpha = [x_l^{-'} \ u^{-'}]'$ define a matrix P to store the gradient of the state

$$P(t) = \frac{\partial x(t)}{\partial \alpha} \tag{22}$$

The dynamics of P are

$$\dot{P} = \frac{\partial f_2}{\partial x} P \tag{23}$$

By numerically integrating this P matrix, and multiplying the Jacobians of the functions Δ_1 and Δ_2, we can compute the first derivatives of Δ. Special attention should be paid that the take-off time is an implicit function of α. So implicit function theorem should be used when computing gradients at the time of taking off the ground. The second derivatives can be computed in the similar manner. Fig 6 is the phase plot of the SLIP in the flight mode by following the robust trajectory.

The red circles are the post-impact states for five sampled terrain heights. As can be seen on the plot, those post-impact states land exactly on the nominal trajectory, indicating that whatever the terrain height is, the robot can always follow exactly the desired gait after the ground impact, thus the terrain disturbance is completely rejected. Fig. 7 shows the robust open loop leg angle sequence computed from the optimization program versus the deadbeat controller obtained from exhaustive simulation [6]. The nearly perfect match between those two suggests that the model-specific approach for SLIP can merge into our more general optimization scheme.

B. Compass Gait Walker

The configuration of the compass gait walker is shown in Fig. 8. This underactuated robot can maintain a passive limit cycle when walking down a shallow slope due to gravitational force only. Although self-stable, such passive limit cycle is fragile under terrain disturbances, even with a stabilizing controller (Fig 10). We search over the robust limit cycle through optimization. In the optimization scheme, we set the function Π to project the post-impact states to states on the nominal trajectory with the same stance leg angle. Since we require such a function to be injective, to this end we constrain the stance leg angle to be monotonic in the limit cycle, which is a valid requirement for a natural-looking gait. The terrain disturbance is the angle of the virtual slope from step to step, equivalently, the terrain slope changes after every ground impact.

The initial input to the optimization program is the passive limit cycle. After optimization, the expected LQR cost-to-go drops from 45.1462 to 0.7859. The optimized limit cycle
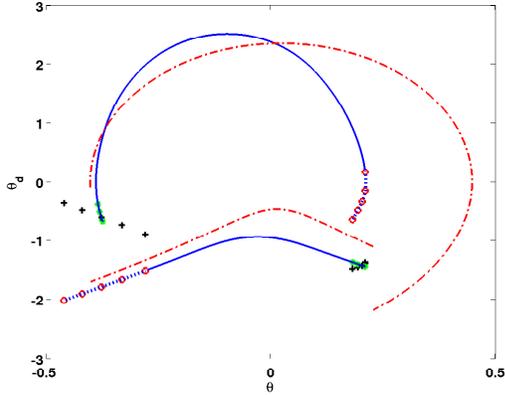
Fig. 9: The two red dashed curves are phase plots of the compass gait's passive limit cycle, with the upper curve being swing leg angle, and the lower curve being stance leg angle. The blue curves are the phase plots of the optimized robust limit cycle, the red curves are the initial input to the optimization program.

for the compass gait is depicted in the blue phase plot in Fig. 9. We suppose the virtual terrain slope is drawn from a uniform distribution within $[2° 8°]$. The dotted blue curve is the segment of the trajectory that might hit the impact surface with terrain slope within the given range. We take five sampled terrain slopes $[2° 3.5° 5° 6.5° 8°]$. The five red circles on each blue curve represent the five prior-impact states corresponding to five sampled terrain slopes. The black plus signs near each curve represent the post-impact states corresponding to the sampled terrain slopes. The green asterisks on each curve represent the knot points on the nominal trajectory projected by each post-impact state. Note that one of the black plus sign and green asterisk exactly overlap, indicating periodic solution on the nominal terrain.

We simulate the compass gait robot to walk on rugged terrain by tracking the passive limit cycle and the optimized robust limit cycle respectively. Both trajectories are stabilized by an infinite-horizon LQR controller with the same instantaneous state and control cost matrices. The phase plots of the LQR-stabilized trajectories for two cases are shown in Fig 10 and 11. In the simulation, the compass gait walker remains stable even after walking 10,000 steps on the stochastic uneven terrain, with the virtual one-step slope γ drawn from uniform distribution within $[2° 8°]$, by tracking the robust limit cycle (Fig. 11). On the other hand, tracking passive limit cycle ends up with failure in averagely less than 10 steps on the same stochastic terrain (Fig 10). From Fig 11 it is also clear that with the stabilizing controller, the perturbed trajectories converge to the nominal robust limit cycle. The comparison between optimized limit cycle and passive limit cycle proves that our scheme finds an inherently robust trajectory that alleviates the effect of terrain disturbances.

IV. DISCUSSION

In the current work we do not develop a general form of the projection function that determines the state on the
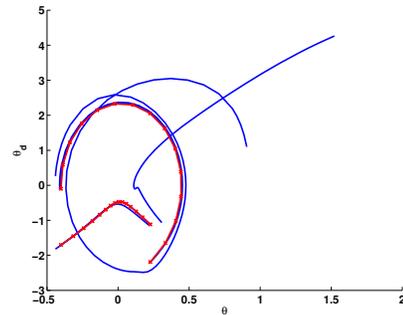
Fig. 10: Phase plot of tracking passive limit cycle, stabilized by LQR. The walker falls down after 3 steps on rough terrain. The red marked plot is the passive nominal limit cycle.
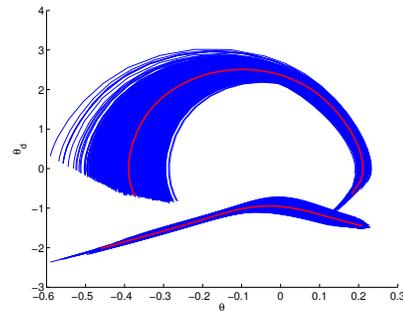
Fig. 11: Phase plot of tracking robust limit cycle for 1000 steps, stabilized by LQR. The system does not fail during even 10000 steps on the same rough terrain. The red curves are the nominal limit cycle, and the blue curves are the simulated trajectories on the rough terrain.

nominal trajectory to be tracked by the post-impact states. Inspired by controlling dynamics in the transversal coordinate [22], we might choose the transversal plane projection, although designing transversal coordinates of the trajectory requires extra effort, and might also be model-specific.

Every sample of the cost-to-go matrix is parameterized as decision variables in the optimization program by using the direct collocation approach. Although convenient in formulation, such parameterization greatly increases the size of the optimization program, from $O(nN)$ to $O(n^2N)$, with n as the dimension of the system and N as the the number of samples. When systems with high DOFs are considered, it might be difficult for the optimizer to handle a huge number of decision variables and constraints. In contrast, the indirect shooting method parameterizes only the ending sample as the decision variables, thus reduces the size of the optimization program significantly. When the size of the problem becomes a concern , we might need to switch to the indirect method for the cost-to-go matrices. Still, we want to keep direct collocation parameterization of the state and control to achieve a good convergence rate. Moreover, an alternative approach to reduce the size of the program is to choose higher order interpolation method instead of cubic Hermite interpolation, thus retain desirable accuracy while taking fewer samples.

The cost function is chosen as the expected value of the

cost-to-go of the post-impact state. We can also adopt the maximum cost-to-go and work with the worst case scenario. But the maximum function can hardly be approximated by a simple smooth function. Such a non-smooth function cannot be handled by the gradient-based optimization method.

V. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

In this paper, we have proposed a novel approach planning a limit cycle that is robust to unknown terrain changes for legged locomotion, such limit cycle can be easily stabilized by simple linear controllers. We show that the expected LQR optimal cost-to-go of the post-impact state is a good metric for the robustness to terrain disturbances. We present a novel direct collocation formulation to solve simultaneously the state/control sequences and the LQR cost-to-go matrices through our optimization program.

To the best of the authors' knowledge, this is the first general approach to search for an intrinsically robust trajectory that handles hybrid impact disturbances explicitly. Also notable is the fact that this was the first time that we have produced a simple linear controller capable of stabilizing the compass gait model on rough terrain in the full coordinates (as compared to Manchester's work, which also utilizes linear controller, but in the transversal coordinates [13]). The proposed technique can recover the known open-loop control strategy for the Spring Loaded Inverted Pendulum. Moreover, this scheme generalizes to a more complicated system like the compass gait walker, and achieves dramatic increase in number of steps taken over rough terrain compared to the passive limit cycle.

B. Future Work

Further investigation will be to investigate the formulation so that the trajectory optimization scales well for a more complicated system. Moreover, we will seek to construct a robust controller for limit cycles, so as to further reject terrain disturbances.

VI. ACKNOWLEDGMENTS

REFERENCES

[1] John T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. SIAM, 2009.

[2] K. Byl and R. Tedrake. Approximate optimal control of the compass gait on rough terrain. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1258 –1263, may 2008.

[3] Katie Byl and Russ Tedrake. Metastable walking machines. *International Journal of Robotics Research*, 28(8):1040–1064, August 1 2009.

[4] Peng Cheng and S.M. LaValle. Reducing metric sensitivity in randomized trajectory design. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 1, pages 43 –48 vol.1, 2001.

[5] P.J. Enright and B.A. Conway. Optimal Finite-Thrust Spacecraft Trajectories Using Collocation and Nonlinear Programming. In *Proceedings of the AAS/AIAA Astrodynamics Conference*, volume 1, pages 479–490, 1990.

[6] Michael Ernst, Hartmut Geyer, and Reinhard Blickhan. Spring-Legged Locomotion on Uneven Ground: a Control Approach to Keep the Running Speed Constant. In *International Conference on Climbing and Walking robots*, pages 639–644, 2009.

[7] Hartmut Geyer, Andre Seyfarth, and Reinhard Blickhan. Spring-mass running: simple approximate solution and application to gait stability. *Journal of Theoretical Biology*, 232(3):315 – 328, 2005.

[8] Philip E. Gill, Walter Murray, Michael, and A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, 12:979–1006, 1997.

[9] E. Glassman and R. Tedrake. A quadratic regulator-based heuristic for rapidly exploring state space. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 5021 –5028, may 2010.

[10] A. Goswami, B. Espiau, and A. Keramane. Limit cycles and their stability in a passive bipedal gait. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 1, pages 246 –251 vol.1, apr 1996.

[11] C. R. Hargraves and S. W. Paris. Direct trajectory optimization using nonlinear programming and collocation. *J Guidance*, 10(4):338–342, July-August 1987.

[12] Ian R. Manchester, Uwe Mettin, Fumiya Iida, and Russ Tedrake. Stable dynamic walking over uneven terrain. *The International Journal of Robotics Research (IJRR)*, 30(3):265–279, January 24 2011.

[13] Ian R Manchester, Uwe Mettin, Fumiya Iida, and Russ Tedrake. Stable dynamic walking over uneven terrain. *The International Journal of Robotics Research*, 30(3):265–279, 2011.

[14] Tad McGeer. Passive dynamic walking. *The International Journal of Robotics Research*, 9(2):62–82, 1990.

[15] K. Mombaur. Performing open-loop stable flip-flops an example for stability optimization and robustness analysis of fast periodic motions. In M. Diehl and K. Mombaur, editors, *Fast Motions in Biomechanics and Robotics: Optimization and Feedback Control*, volume 340 of *Lecture Notes in Control and Information Sciences*, pages 253–275. Springer, 2006.

[16] Katja D. Mombaur. *Stability Optimization of Open-Loop Controlled Walking Robots*. PhD thesis, Heidelberg, August 2001.

[17] J. Morimoto, G. Zeglin, and C.G. Atkeson. Minimax differential dynamic programming: application to a biped walking robot. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pages 1927–1932 vol.2, oct. 2003.

[18] H.W. Park, Koushil Sreenath, Alireza Ramezani, and JW Grizzle. Switching Control Design for Accommodating Large Step-down Disturbances in Bipedal Robot Walking. In *International Conference on Robotics and Automation*, 2012.

[19] Michael Posa and Russ Tedrake. Direct trajectory optimization of rigid body dynamical systems through contact. In *Under review*, 2012.

[20] Marc Raibert, Kevin Blankespoor, Gabriel Nelson, Rob Playter, and the BigDog Team. Bigdog, the rough-terrain quadruped robot. *Proceedings of the 17th World Congress, The International Federation of Automatic Control*, 2008.

[21] R.Blickhan. The spring-mass model for running and hopping. *Journal of Biomechanics*, 22(11–12):1217–1227, 1989.

[22] A. Shiriaev, L. Freidovich, and I. Manchester. Can we make a robot ballerina perform a pirouette? orbital stabilization of periodic motions of underactuated mechanical systems. *Annual Reviews in Control*, 32(2):200–211, December 2008.

[23] Manoj Srinivasan and Andy Ruina. Computer optimization of a minimal biped model discovers walking and running. *Nature*, 439:72–75, January 5 2006.

[24] Russ Tedrake. *Underactuated Robotics: Learning, Planning, and Control for Efficient and Agile Machines: Course Notes for MIT 6.832*. Working draft edition, 2012.

[25] O. von Stryk and R. Bulirsch. Direct and indirect methods for trajectory optimization. *Annals of Operations Research*, 37:357–373, 1992. 10.1007/BF02071065.

[26] Eric R. Westervelt, Jessy W. Grizzle, Christine Chevallereau, Jun Ho Choi, and Benjamin Morris. *Feedback Control of Dynamic Bipedal Robot Locomotion*. CRC Press, Boca Raton, FL, 2007.