Proceedings of the ASME 2020 Dynamic Systems and Control Conference
DSCC 2020
October 4-7, 2020, Pittsburgh, Pennsylvania, USA

DSCC2020-3314

RECEDING HORIZON CONTROL FOR A 2D POINT-MASS HOPPING MODEL NAVIGATING ON TERRAIN WITH STEPPING STONES AND STAIRS

Ali Zamani
Robotics and Motion Laboratory
Dept. of Mechanical and Industrial Eng.
University of Illinois at Chicago
842 W. Taylor St. Chicago, IL 60607
Email: alizamani.mecheng@gmail.com

Pranav A. Bhounsule
Robotics and Motion Laboratory
Dept. of Mechanical and Industrial Eng.
University of Illinois at Chicago
842 W. Taylor St. Chicago, IL 60607
Email: pranav@uic.edu

ABSTRACT

We consider the problem of a 2D point mass model navigating a complex terrain comprising of stepping stones and stairs while optimizing an energy metric. We solve the problem using receding horizon control as follows. We preview a fixed distance ahead at mid-flight. Then we optimize the number of steps, the controls, and foot placement location, choosing the solution with least energy cost. However, we implement only the solution for the first step which takes the model to the next mid-flight. This process continues until the model reaches the end of the terrain. We improve on past approaches by (1) considering a fixed distance preview as done by humans instead of fixed time or fixed steps, and (2) adding obstacles as a cost and elevation change as a condition within the model dynamics, thus avoiding mixed-integer formulations which are computationally expensive. The resulting problem is solved using constrained nonlinear programming. We demonstrate that the approach works for randomly chosen terrain consisting of stepping stones and stairs.

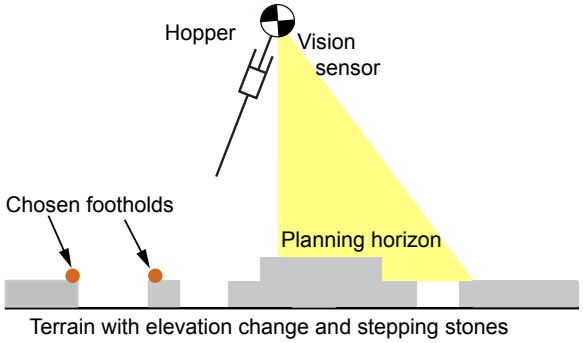

FIGURE 1: **Problem conceptualization:** The hopper has to negotiate a terrain consisting of stairs and stepping stones (gray rectangles). At mid-flight, the hopper uses a vision sensor to preview a fixed distance ahead. Then the hopper plans the optimal steps and strategy to navigate the fixed distance. The hopper then executes the optimum strategy for the first step until the next mid-flight. Then the hopper replans as before continuing the process until it crosses the terrain.

1 Introduction

Dynamically balancing robots have small feet or point feet and have to continue moving to stabilize themselves. Because of their small footprint and dynamic nature, they are able to navigate complex terrains consisting of ditches, stairs, and obstacles. However, planning and control of dynamically balancing robots on such complex terrain presents a formidable computational challenge because of the discretely changing dynamics (i.e., dynamics during touchdown are different than those in free flight). Moreover, if an objective function has to be minimized then there are additional computational challenges. Here, we address the problem of navigating a terrain consisting of stepping stones and stairs, a benchmark problem in dynamic legged locomotion, while optimizing an energy metric while taking into

account the robot dynamics.

We consider a realistic scenario that a robot might be subjected to and is shown in Fig. 1. Here the robot can see a fixed distance ahead with a vision sensor (e.g., an RGB-D camera) in the flight phase. The robot then plans the optimum number of steps and controls for each step over the fixed distance ahead (planning). Next, the robot implements the control strategy for only the first step to reach the next flight phase (control). The process continues until the robot reaches the end of the terrain. This formulation of the problem that uses a model to plan ahead based on sensory data is known as model predictive control (MPC) or receding horizon control (RHC). Although RHC is popular in robotic applications such as cars and drones, it is fairly new in the area of legged locomotion.

2 Background and related work

The simplest approach to motion planning is to first plan foot step locations from start to goal, followed by creating a controller that will enable the robot to meet those foot step locations. The foot step planning may be achieved with A-star planner with heuristics such as effort, risk, and/or number and complexity of steps taken to plan footsteps from start to goal [1] or energy estimates obtained from human movement data [2]. The main issue with this approach is that the planner is usually based on kinematics and conservative estimates of possible motion leading to sub-optimal solutions.

A more complete approach is to do foot step planning by considering the dynamics. One way to do this is to precompute all feasible solutions for a single step considering the dynamics. Then these feasible solutions can then be searched using A-star planners [3] and probabilistic road maps [4]. Although these approach considers the dynamics, these methods work with discrete controls and states, thus they are not able to handle boundary conditions (e.g., step length, velocity constraint, final state specified).

Boundary value problems are much easily handled with continuous-optimization methods (e.g., trajectory optimization). Given a set of footholds, presumably from A-star planner, one needs to solve for a trajectory optimization problem that minimizes a suitable cost while constraining the feet to line up with the chosen footholds [5]. Alternately, the foot step location can be included as a cost function using control barrier function [6].

Continuous optimization problems where foothold locations are optimization variables lead to an OR constraint [7, 8]. That is, step one can only take place at one point in the terrain and no other place. This is formulated as a mixed integer optimization problem that is computationally more challenging than a traditional optimization problem with only real numbers.

While most past approaches consider stepping on (e.g., elevation) or away (e.g., stepping stones) from an obstacle, there are problems where it is best to step over the obstacle. In such case, one needs to consider the possibility that the leg would collide with the obstacle in mid-air. To tackle this problem, one can discretize the state at multiple points in space and obstacles as polytopes. Then the optimization problem is to plan the motion of the robot to be in the free space polytopes using a mixed integer programming approach [9].

Receding horizon control is gaining popularity as tool for robust trajectory optimization. The key idea is to preview and consequently plan for fixed time ahead [10, 11]. But only some portion of the plan is implemented while the rest is discarded. As the robot moves to a new position, new data is available and the process continues.

Our approach is also based on receding horizon control as follows. The robot scans a fixed distance ahead, then plans footholds, controls, and number of steps, implements the solution for the first step and continues the process until it reaches the end of the terrain. Our work is novel from previous works in the following ways: (1) we plan for a fixed distance ahead, instead of fixed time or fixed steps ahead, as this is more similar to human behavior [12] and (2) we incorporate obstacles (here stepping stones) as a cost and elevation change in the dynamics of touchdown thus avoiding mixed-integer programming formulation.

3 Model

Fig. 2 shows a 2D point mass model [13]. The model consists of a point mass m_0 at the hip and a massless leg with a maximum leg length ℓ_0. There are two actuators, a prismatic actuator that generates an axial force F along the leg during the support phase and a rotary actuator that servos the leg at a desired angle θ during the flight phase. The variables of the model are non-dimensionalized by dividing them to terms given as follows: time by $\sqrt{\ell_0/g}$, distance and leg length by ℓ_0, velocity by $\sqrt{\ell_0 g}$, acceleration by g, and force by mg.

The non-dimensionalized states of the model are represented by $\{x,\dot{x},y,\dot{y}\}$, where x and y are the positions of the center of mass, and $\dot{x}$ and $\dot{y}$ are the corresponding velocities. The model starts at the apex (see Fig. 2(a)), where the state vector is $\{x_i,\dot{x}_i,y_i,\dot{y}_i=0\}$, and then falls under gravity given by

$$\begin{aligned} \ddot{x} &= 0, \\ \ddot{y} &= -1, \end{aligned} \tag{1}$$

until the contact with the ground is detected by the touchdown event $y-\cos\theta-h(x_c)=0$, where x_c is the x-position of the contact point and $h(x)$ is the terrain profile. Thereafter, the hopper enters the stance phase represented by

$$\begin{aligned} \ddot{\ell} &= \ell\dot{\theta}^2-\cos\theta+F, \\ \ell\ddot{\theta} &= -2\dot{\ell}\dot{\theta}+\sin\theta, \end{aligned} \tag{2}$$

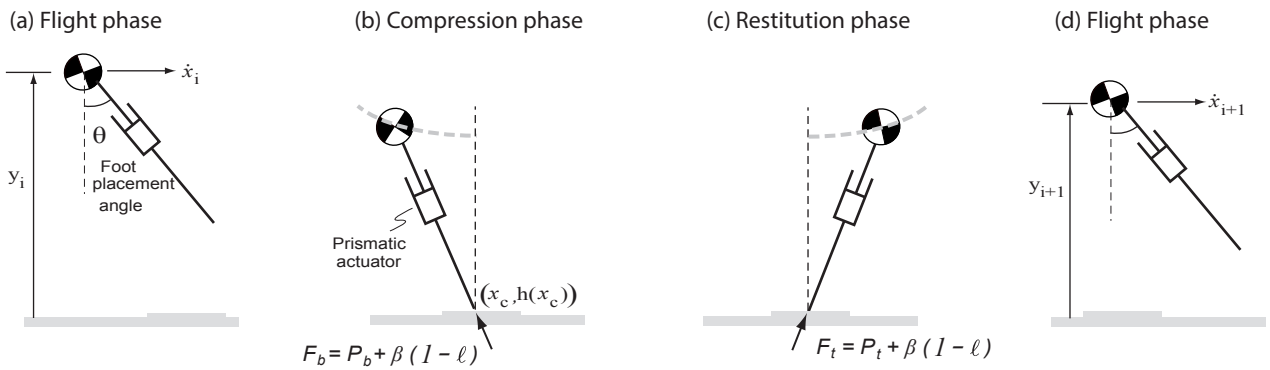

FIGURE 2: **A complete step for the hopping model:** The model starts in the flight phase at the apex position (vertical velocity is zero), followed by the stance phase, and finally ending in the flight phase at the apex position of the next step. The hopping model has a prismatic actuator that is used to provide an axial braking force F_b in the compression phase and an axial thrust force F_t in the restitution phase, and a hip actuator (not shown) that can place the swinging leg at an angle θ with respect to the vertical as the leg lands on the ground.

where F is a positive axial force along the leg. The first half of the stance phase from touchdown to the maximum compression of the leg length (defined by $\dot{\ell} = 0$) is called compression phase (Fig. 2(b)) and the second half of the stance phase from the maximum compression of the leg length to takeoff (defined by $\ell - 1 = 0$) is called restitution phase (Fig. 2(c)). The axial force during the compression phase is $F = P_b + \beta(1-\ell)$ and during the restitution phase is $F = P_t + \beta(1-\ell)$. In these equations, $\beta = \frac{k\ell_0}{mg}$ is a constant, k is a constant (fixed) gain analogous to the spring constant, $\ell = \frac{\sqrt{(x-x_c)^2+(y-h(x_c))^2}}{\ell_0}$ is the instantaneous leg length measured with respect to the contact point x_c, and P_b and P_t are constant braking and thrust forces, respectively. After the takeoff, the model enters the flight phase and ends up in the next apex state, $\{x_{i+1}, \dot{x}_{i+1}, y_{i+1}, \dot{y}_{i+1} = 0\}$, (Fig. 2(d)).

Conceptually it is much easier to use step-to-step model as the building block for optimization rather than the continuous time model based on equations of motion. If the model state at the apex is $\mathbf{z}_i = \{x, \dot{x}, y, \dot{y} = 0\}_i$ and the controls are $\mathbf{u}_i = \{\theta, P_b, P_t\}_i$ then one can define a function $\mathbf{F}$ such that

$$\mathbf{z}_{i+1} = \mathbf{F}(\mathbf{z}_i, \mathbf{u}_i, h(x_{ci})). \tag{3}$$

The function $\mathbf{F}$ defines the step-to-step map or the Poincaré map and includes the terrain profile $h(x_c)$. We obtain it by integrating the equations of motion for a given initial conditions $\mathbf{z}_i$ and controls $\mathbf{u}_i$.

4 METHODS

4.1 Terrain with ditches and elevation changes

The terrain profile considered here consists of ditches and elevation changes (step-up and step-down). An example terrain is shown in Fig. 3 (a).

The terrain is divided into permissible and forbidden regions for foothold positions. The permissible regions are shown in gray color and the forbidden regions are shown in red-dashed color. The latter includes (1) the ditches and (2) the flat regions near the elevation change. One way to incorporate the forbidden regions in the optimization is to specify the feasible regions as constraints and then find foothold positions within those feasible regions. This formulation leads to a mixed-integer programming problem [7, 9], which is harder to solve especially when the objective function is non-linear as it is the case here.

We avoid mixed-integer formulations by incorporating the forbidden regions as a cost function. To do so, we define a terrain cost as shown in Fig. 3 (b). The terrain cost is zero at all permissible regions, but is sufficiently high for all forbidden region. Then we fit a cubic spline to ensure that the cost is a smooth function of the terrain. Thus, given a foothold position x_c, the terrain cost $C_{\text{terrain}}(x_c)$ is known. The elevation change is incorporated in the function h(x) by fitting a piecewise cubic Hermite interpolating function (*pchip* in MATLAB) to the terrain.

4.2 Receding horizon control problem

The model starts at the apex of flight phase with initial position $x = 0$ with apex height and apex horizontal speed specified $y(x = 0)$ (given) and $\dot{x}(x = 0)$ (given). The terrain profile given is specified by fitting a cubic spline $h(x)$ where x specifies the x-position. At each mid-flight the model can see and hence plan a fixed distance d_{horizon}. Finally, we also impose that the end of the terrain is $x = D$, and the apex height and apex horizontal speed are the same as the beginning. That is, $y(x = D) = y(x = 0)$ and $\dot{x}(x = D) = \dot{x}(x = 0)$.

We solve a receding horizon control problem at each apex,

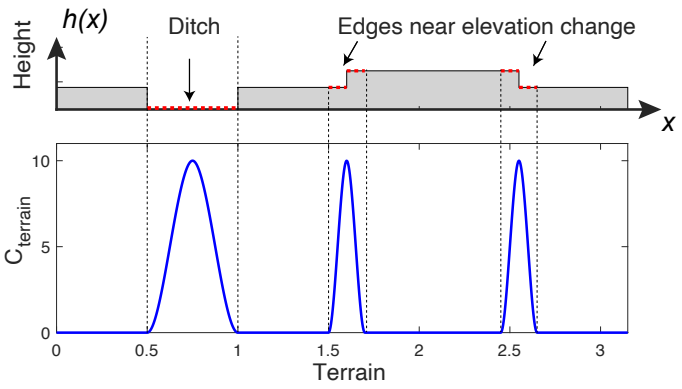

FIGURE 3: **Terrain profile:** (a) The terrain is shown with gray rectangles and ditches are in the white spaces between the stepping stones. For feasible movement all footholds (x_c) should be on the gray rectangles. We model the terrain with piecewise cubic polynomial, $h(x)$. Thus, given a foot placement x-position to be x_c, the vertical position is found as $h(x_c)$. (b) To avoid stepping in the ditch and on edges near elevation change (shown as red dashed line), we assign a terrain cost that increases sharply as shown.

where we re-initialize the step number $i = 0$. The model's current x-position x_0, apex height y_0, and horizontal speed $\dot{x}_0$ are known. The vision sensor enables the robot to preview the terrain for a distance $x_0 \leq x \leq x_0 + d_{\text{horizon}}$. Now we formulate a trajectory optimization problem

$$\begin{aligned}
\min_{N,\mathbf{z}_i,\mathbf{u}_i,x_{ci}} \quad & \frac{\sum_{i=0}^{i=(N-1)} E_i}{(x_{N-1}-x_0)} + \sum_{i=0}^{i=(N-1)} C_{\text{terrain}}(x_{ci}) \\
\text{subject to:} \quad & \mathbf{z}_{i+1} = \mathbf{F}(\mathbf{z}_i, \mathbf{u}_i, h(x_{ci})); \\
& \mathbf{u}_{\min} < \mathbf{u}_i < \mathbf{u}_{\max}; \\
& D_{\min} < x_{i+1} - x_i < D_{\max}; \\
& x_{N-1} = d_{\text{horizon}}.
\end{aligned} \tag{4}$$

where $i = 0, 1, ..., (N-1)$ indicates the planning horizon up to N steps noting that N is 1 parameter, $\mathbf{z}_i = \{x, \dot{x}, y, \dot{y} = 0\}$ are $3(N+1)$ parameters, $\mathbf{u}_i = \{\theta, P_b, P_t\}_i$ is the set of controls at step i, thus totaling $3N$ parameters, and x_{ci} is the foot location at step i thus a total of N parameters. Thus, the total decision variables are $1 + 3(N+1) + 3N + N = 7N + 4$ optimization parameters. The mechanical energy usage at each step, $E_i = E_k + E_{P_b} + E_{P_t} = \int(|k(\ell - \ell_0)d\ell| + |P_t d\ell| + |P_b d\ell|)$ is the mechanical energy used per step, $C_{\text{terrain}}(x_{ci})$ is the terrain cost as described in Sec. 4.1, $D_{\min}$ and $D_{\max}$ are minimum and maximum step length. The absolute value is a non-smooth function of its argument, so we can smooth it out using square-root smoothing [14].

This optimization problem can be solved using NPL solvers such as SNOPT [15]. It should be noted that in this optimization problem the decision variables depend on N, so it is not possible to simultaneously optimize N and the other decision variables. So there are two loops, an inner and an outer loop. The decision variable N is optimized in the outer loop and the other decision variables are optimized in the inner loop for a given N. This optimization problem can be solved quite fast by suitable choice of planning horizon, d_{horizon}.

5 Results

We present optimization results for: (1) terrain with randomly placed stepping stones, (2) terrain with stairs going incrementally up and down, and (3) terrain consisting of randomly placed stepping stones and stairs. For all these optimizations, unless otherwise noted, we set up the numerics as follows.

We describe the model parameters and solution. We chose the free parameter $\beta = 40$. We obtain the step-to-step or Poincaré map $\mathbf{F}$ by numerically integrating the equations of motion using dop853, a Runge-Kutta method of order $8(5,3)$ with adaptive step size [16]. The integrator also has built-in capability to detect events such as touchdown, mid-stance, take-off, and mid-flight. The integrator is written in C++ and called from MATLAB 2019b using a mex interface. The relative and absolute tolerances for the integrator are set to 10^{-12}.

We describe the optimization parameters. The terrain starts at $x = 0$ and terrain ends at $x = D = 22$. The initial apex height is $y(x = 0) = 1.2$ and the initial horizontal speed is $\dot{x}(x = 0) = 1.1$. The minimum and maximum step lengths are $D_{\min} = 1.05$, $D_{\max} = 2.2$, respectively. The bounds for the controls, $\mathbf{u}_{\min} = \{\theta, P_b, P_t\}$ are: $\mathbf{u}_{\min} = \{5, 0, 0\}$, and $\mathbf{u}_{\max} = \{30, 5, 5\}$ where θ bounds are in degrees. The planning horizon is $x_{N-1} = d_{\text{horizon}} = 4$. The planning horizon and the bounds on the step lengths enable us to compute the minimum and maximum number of steps. Thus, we obtain the minimum planned steps by rounding to the nearest integer toward positive infinity, $N_{min} = \frac{d_{\text{horizon}}}{D_{\max}} = \frac{4}{2.1} = 1.9 \sim 2$. Similarly, we obtain the maximum planned steps by rounding to the nearest integer toward negative infinity, $N_{min} = \frac{d_{\text{horizon}}}{D_{\min}} = \frac{4}{1.05} = 3.81 \sim 3$. Thus, at each step at mid-flight we compute the optimal solution for $N = 2$ and $N = 3$ and use the solution that has the least cost. We use constraint nonlinear programming solver SNOPT [15] to solve the optimization problem. We now show results for the three terrains.

5.1 Terrain with stepping stones

This terrain consists of 9 stepping stones with minimum and maximum lengths of 0.7 and 3, and 8 ditches between stepping stones with minimum and maximum lengths of 0.8 and 1.3. We solve the receding horizon control problem described in Sec. 4.2

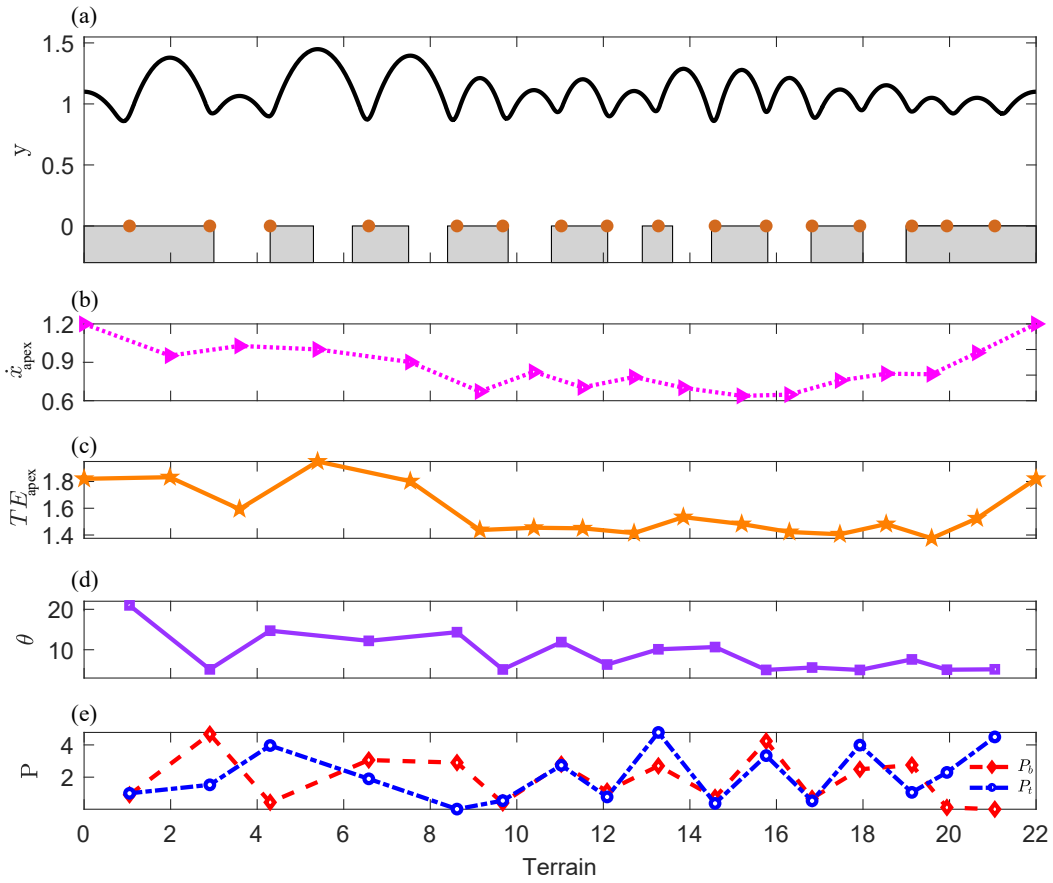

FIGURE 4: **Terrain with stepping stones:** (a) x-y trajectory of the point mass shown in black solid line and the foothold positions shown as brown solid circles, (b) the forward velocity at apex, (c) the total energy at apex, (d) foot placement angle, and (e) the braking and thrust forces.

for this terrain. Fig. 4(a) shows the stepping stones, the ditches, the foothold positions, and the x-y position of the point mass. As seen in the figure, it takes the robot 16 steps to successfully traverse the terrain. Fig. 4(b-e) illustrates the forward velocity at apex, $\dot{x}_{\text{apex}}$, the total energy at apex, $TE_{\text{apex}} = 0.5\dot{x}_{\text{apex}}^2 + y_{\text{apex}}$, and the controls, $\{\theta, P_b, P_t\}$, at each step, respectively. The first stepping stone and ditch are 3 and 1.3 in length, respectively. Since the planning horizon is 4, and the step length is bounded, the robot needs to take two steps on the first stepping stone. To do so, the forward velocity needs to be decreased at the second apex so that the robot can take a short step, but the height at the second apex needs to be increased so that the total energy stays almost the same. Taking a short step requires a small foot placement angle, but the small foot placement angle would accelerate the body forward. To compensate for this acceleration, the braking force increases at a higher rate compared to the thrust force. Similar reasoning can be applied for other steps. The robot tries to keep the total energy at apex constant if possible as seen mostly in the second half of the terrain, except for the last two steps to ensure that the robot meets the final conditions at the end of the terrain. The MCOT computed for the entire terrain is $\sum_{i=1}^{i=16} E_i / d_{\text{horizon}} = 3.51/22 = 0.1595$ which is 14.66% higher than the $MCOT = 0.139$ computed for a level ground with the same terrain length and without ditches.

5.2 Terrain with stairs

This terrain consists of 9 stairs with minimum and maximum lengths of 1.2 and 2.1, and heights of 0.1 and 0.25. We solve the receding horizon control problem for this terrain. As seen in Fig. 5 (a), it takes the robot 16 steps to successfully traverse the terrain. The height of the point mass with respect to the ground increases until the robot reaches the top most stair and then decreases until the robot reaches the end of the terrain, but the rate of height change depends on the relative elevation change of two consecutive stairs. Fig. 5 (b) shows the forward velocity at apex which mostly decreases for hopping up the stair and increases for hopping down the stairs. The total energy at apex

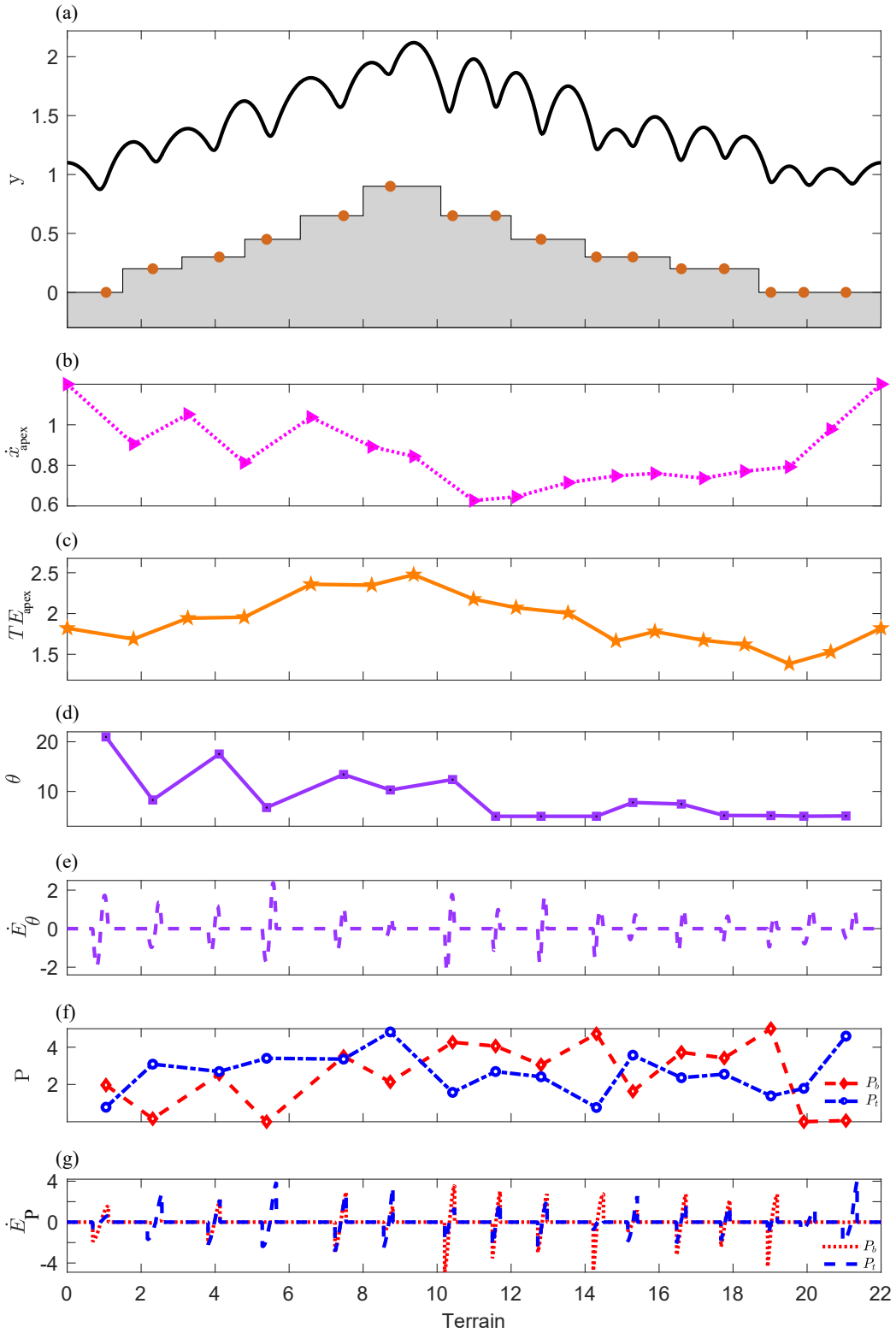

FIGURE 5: **Terrain with stairs:** (a) x-y trajectory of the point mass shown in black solid line and the foothold positions shown as brown solid circles, (b) the forward velocity at apex, (c) the total energy at apex, (d) the foot placement angle, and (e) the power consumption by the foot placement angle, (f) the braking and thrust forces, and (g) the power consumption by the braking and thrust forces.

shown in Fig. 5 (c) nearly follows the trend of the height at apex. For hopping up the stairs, the robot takes different step lengths due to the change in stair height and that is why we see fluctuations in foot placement angle shown in Fig. 5 (d). However, the foot placement angle slightly changes for hopping down the stairs since the robot mostly takes 2 steps at each stair. Fig. 5 (e) shows the power consumption of the foot placement angle which is negative during the compression phase and positive during the restitution phase. The thrust and braking forces are illustrated in Fig. 5 (f). The thrust force is greater than the braking force for hopping up the stairs to add energy to the system while it reverses for hopping down the stairs to dissipate energy from the system. Fig. 5(g) shows the power consumption of the braking and thrust forces. The power consumption of the thrust force is greater than that of the braking force for hopping up the stairs and less for hopping down the stairs. We see a different trend in the last step due to the final condition imposed on the forward velocity and height at apex. The MCOT computed for the entire terrain is $MCOT = 3.528/22 = 0.1604$ which is 15.4% higher than that computed for a level ground with the same terrain length.

5.3 Terrain with stepping stones and stairs

This terrain consists of several stepping stones, ditches, and stairs. The minimum and maximum lengths of the ditches are 0.5 and 1.2, and the height of stairs is fixed to 0.2 with respect to their base. For this particular terrain, in addition to solving the RHC problem, we performed another optimization problem, called baseline. In this optimization problem, we preview the entire terrain at mid-flight, $x_{N-1} = D = 22$, and optimize the decision variables. Then the solution for the entire terrain is implemented which takes the model to the end of the terrain. Fig. 6 shows the terrain, x-y position of the point mass, and foothold positions for both problems. As seen in this figure, for both optimizations, it takes the model 15 steps to successfully traverse the terrain. The x-y position of the point mass for both optimizations is almost similar in the first half of the terrain. However, they look different in the second half. This is because the baseline optimization has the knowledge of the whole terrain in advance so it plans the steps such that there would be smooth change between steps to meet the final conditions at the end of the terrain. For brevity, we show the rest of the results only for the RHC optimization in Fig. 7. The results in this figure can be explained similar to those in Figs. 4-5. The mechanical cost of transport for each optimization is $MCOT_{\text{RHC}} = 13.76$ and $MCOT_{\text{baseline}} = 14.07$.

6 Discussion

In this work, we have developed a control approach based on receding horizon control to solve the navigation problem of a 2D point-mass hopping model on complex terrain consisting of stepping stones and stairs. The efficacy of the approach was shown in tasks involving locomotion on various terrains including stepping stones, stairs, and the combination of the two.

Our finding are as follows:

1. For the terrain with stairs, the apex forward velocity and height generally decreases and increases for hopping up the stairs, respectively, and vice versa for hopping down the stairs. The total energy at apex is dominated by the apex height rather than the apex forward velocity. The constant thrust force is greater than the constant braking force for hopping up the stairs to add energy to the system and vice versa for hopping down the stairs. For the stairs with considerable length, the changes of the foot placement angles between steps are minimal because the model takes mostly two steps in a single stair. Also the thrust force does more work than the braking force for hopping up and less work for hopping down the stairs, but both forces do more work than the foot placement angle during the entire locomotion.
2. For the terrain with stepping stones, when the model takes a high jump to overcome a ditch, the velocity generally decreases to avoid the significant change of the total energy at apex. Similar to hopping on stairs, the apex height dominates the apex forward velocity in determining the apex total energy. The work done by the foot placement angle for hopping on stepping stones is considerable compared to that by the braking and thrust forces, and both forces have generally done similar work during locomotion.
3. For the terrain consisting of both stairs and stepping stones, the behavior of the model is almost a combination of the two behaviors described for the other two terrains.

Our control approach has several advantages over previous methods. First, the planning horizon in our control framework is a fixed distance the robot can scan ahead rather than time or number of steps. Planning based on the fixed distance ahead is more analogous with human behavior [12] and consistent with the mechanical cost of transport used as a common energy metric in the legged locomotion community.

Second, incorporating obstacles as a suitable cost and elevation change in the dynamics of touchdown allows us to use nonlinear programming (NLP) solvers to optimize decision variables. However, incorporating terrain profile and obstacles as constraints leads to a mixed-integer programming problem which requires branch and bound algorithms [17] to solve. This problem is relatively harder to deal with, especially with nonlinear cost functions. Also, solving the optimization problem using NLP as opposed to sampling-based methods [18] allows the model to meet the boundary conditions such as position and velocity constraints.

In the course of optimization, no passive solution ($P_b = P_t = 0$) is found even for parts of the terrain with no stairs/ditches. This is due to the fact that our robot model is not the spring

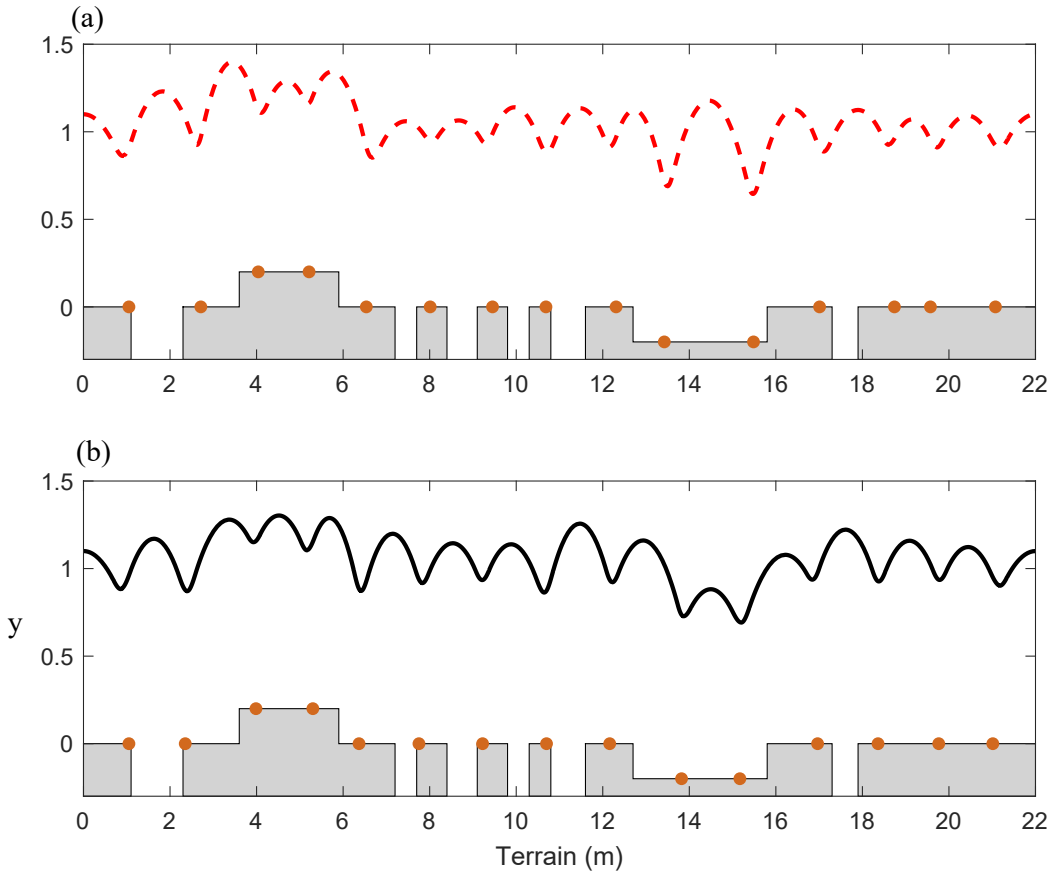

FIGURE 6: **Comparing two different optimization problems** (a) baseline optimization problem with $x_{N-1} = 22$, and (b) RHC optimization problem with $x_{N-1} = 4$

loaded inverted pendulum model in which there is a physical spring for storing and releasing energy during the stance phase.

There was insignificant difference among MCOTs for various planning horizons. We solved the RHC problem for different planning horizons $x_{N-1} = 4, 6, 8$ and performed the baseline optimization, $x_{N-1} = D = 22$. The difference among MCOTs was less than 5%.

Our work has several limitations as follows. First, our control approach may be quite challenging for real-time implementation if a long planning horizon is chosen. This increases the number of steps needed to be optimized, thereby making our approach computationally expensive. This issue can be mitigated to some extent by using approximated models (e.g., low order polynomials) for the Poincaré map [19] rather than solving for the equations of motion online. Second, we defined a high cost in the flat regions near the elevation change to avoid the leg collision with stairs. However, this solution may not work for terrain with high elevation changes. A better solution would be to include collision avoidance in the flight phase in the optimization formulation.

7 Conclusions and Future Work

The paper illustrates receding horizon control for motion planning and control of a hopping model on terrain with stepping stones and stairs. In particular, we conclude that both: (1) including the ditches as a cost function with a high penalty and (2) including elevation changes into the physics of the model, avoids mixed-integer formulations that are computationally expensive to solve.

The future work could focus on extending the approach to 3D model navigating on 3D terrain, reformulating the optimization problem to solve it in real-time, and demonstrating the scalability of the method to higher dimensional models such as models with swing leg dynamics and torso dynamics.

ACKNOWLEDGMENT

This work was supported by NSF grant IIS 1946282.

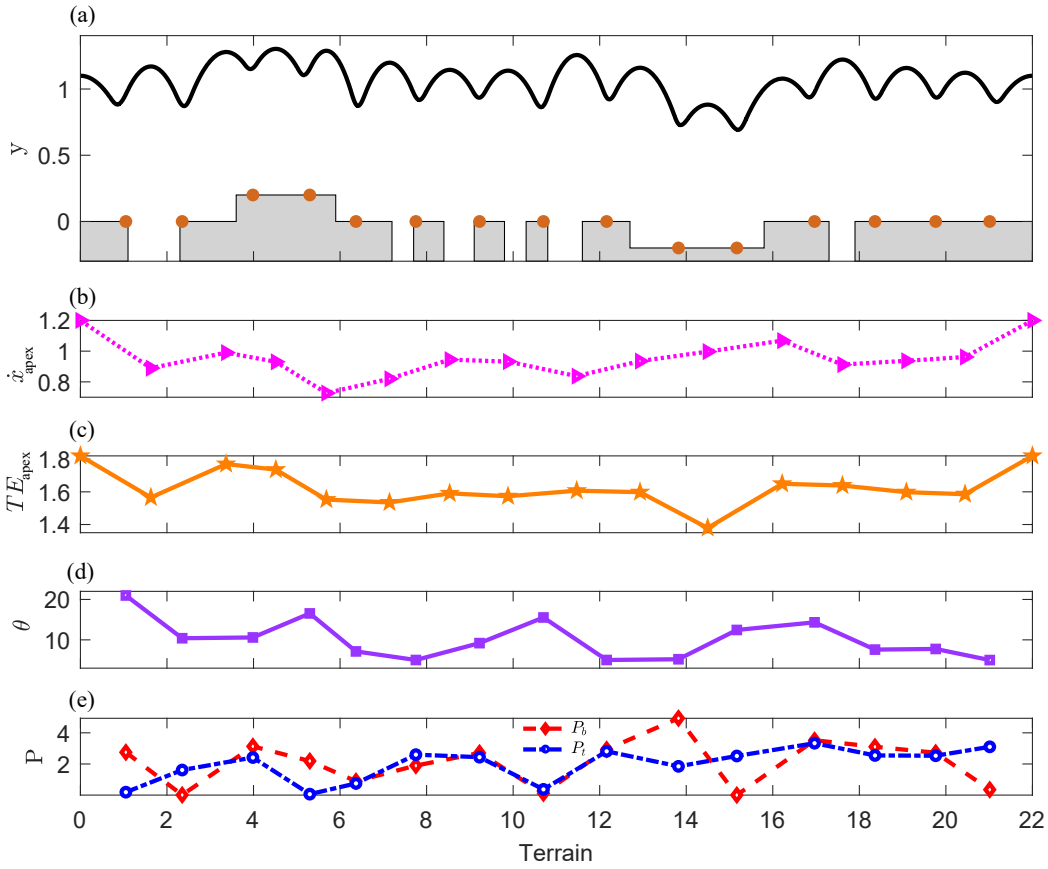

FIGURE 7: **Terrain with stepping stones and stairs:** (a) x-y trajectory of the point mass shown in black solid line and the foothold positions shown as brown solid circles, (b) the forward velocity at apex, (c) the total energy at apex, (d) foot placement angle, and (e) the braking and thrust forces.

REFERENCES

[1] Chestnutt, J., Lau, M., Cheung, G., Kuffner, J., Hodgins, J., and Kanade, T., 2005. "Footstep planning for the honda asimo humanoid". In Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, IEEE, pp. 629–634.

[2] Huang, W., Kim, J., and Atkeson, C. G., 2013. "Energy-based optimal step planning for humanoids". In Robotics and Automation (ICRA), 2013 IEEE International Conference on, IEEE, pp. 3124–3129.

[3] Paris, V., Strizic, T., Pusey, J., and Byl, K., 2016. "Tools for the design of stable yet nonsteady bounding control". In 2016 American Control Conference (ACC), IEEE, pp. 4822–4828.

[4] Campana, M., and Laumond, J.-P., 2016. "Ballistic motion planning". In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 1410–1416.

[5] Rutschmann, M., Satzinger, B., Byl, M., and Byl, K., 2012. "Nonlinear model predictive control for rough-terrain robot hopping". In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, pp. 1859–1864.

[6] Nguyen, Q., Hereid, A., Grizzle, J. W., Ames, A. D., and Sreenath, K., 2016. "3d dynamic walking on stepping stones with control barrier functions". In 2016 IEEE 55th Conference on Decision and Control (CDC), IEEE, pp. 827–834.

[7] Deits, R., and Tedrake, R., 2014. "Footstep planning on uneven terrain with mixed-integer convex optimization". In 2014 IEEE-RAS international conference on humanoid robots, IEEE, pp. 279–286.

[8] Aceituno-Cabezas, B., Cappelletto, J., Grieco, J. C., and Fernández-López, G., 2016. "A generalized mixed-integer convex program for multilegged footstep planning on uneven terrain". *arXiv preprint arXiv:1612.02109*.

[9] Ding, Y., Li, C., and Park, H.-W., 2018. "Single leg dynamic motion planning with mixed-integer convex optimization". In 2018 IEEE/RSJ International Conference on

Intelligent Robots and Systems (IROS), IEEE, pp. 1–6.
[10] Farshidian, F., Jelavic, E., Satapathy, A., Giftthaler, M., and Buchli, J., 2017. "Real-time motion planning of legged robots: A model predictive control approach". In 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids), IEEE, pp. 577–584.
[11] Ansari, A. R., and Murphey, T. D., 2016. "Sequential action control: Closed-form optimal control for nonlinear and nonsmooth systems". *IEEE Transactions on Robotics,* ***32***(5), pp. 1196–1214.
[12] Matthis, J. S., and Fajen, B. R., 2014. "Visual control of foot placement when walking over complex terrain.". *Journal of experimental psychology: human perception and performance,* ***40***(1), p. 106.
[13] Zamani, A., and Bhounsule, P., 2018. "Control synergies for rapid stabilization and enlarged region of attraction for a model of hopping". *Biomimetics,* ***3***(3), p. 25.
[14] Bhounsule, P. A., Zamani, A., Krause, J., Farra, S., and Pusey, J., 2020. "Control policies for a large region of attraction for dynamically balancing legged robots: a sampling-based approach". *Robotica*, pp. 1–16.
[15] Gill, P., Murray, W., and Saunders, M., 2002. "SNOPT: An SQP algorithm for large-scale constrained optimization". *SIAM Journal on Optimization,* ***12***(4), pp. 979–1006.
[16] Hairer, E., NORSETT, S., and Wanner, G., 2000. *Solving Ordinary, Differential Equations I, Nonstiff problems/E. Hairer, SP Norsett, G. Wanner, with 135 Figures, Vol.: 1*. No. BOOK. 2Ed. Springer-Verlag, 2000.
[17] Ebrahimi, N., Guda, T., Alamaniotis, M., Miserlis, D., and Jafari, A., 2020. "Design optimization of a novel networked electromagnetic soft actuators system based on branch and bound algorithm". *IEEE Access*.
[18] Zamani, A., Galloway, J. D., and Bhounsule, P. A., 2019. "Feedback motion planning of legged robots by composing orbital lyapunov functions using rapidly-exploring random trees". In 2019 International Conference on Robotics and Automation (ICRA), IEEE, pp. 1410–1416.
[19] Zamani, A., and Bhounsule, P. A., 2020. "Nonlinear model predictive control of hopping model using approximate step-to-step models for navigation on complex terrain". In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE.