2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)
Toronto, Canada. October 15-17, 2019

Unified Foothold Selection and Motion Planning for Legged Systems in Real-Time

Steven Crews, Sapan Agrawal, and Matthew Travers

***Abstract*— This work presents a novel architecture that unifies footstep planning, motion planning, and online feedback control for legged robots moving through complex environments. Our approach contrasts related prior works that treat planning and control as separate components in a hierarchical framework (first plan, then control). Though prior works have demonstrated success, existing state-of-the-art planning and control architectures for legged robots quickly become brittle in highly uncertain environments due to an inherent inability to dynamically and decisively react to unplanned events. To address this, this work presents a novel framework that uses modeling and analysis tools from the hybrid systems and nonlinear control communities to reformulate planning footsteps and dynamic trajectories as well as deriving closed-loop controllers as a single trajectory optimization problem. By combining these previously disparate steps we empirically show that we can remove much of complexity that underlies the hierarchical decision posed by conventional approaches, making it possible to dynamically and safely react to large external disturbances in sub-real-time. We present results that highlight the reactive and robust nature of the unified framework developed.**

I. Introduction

While traversing complex terrains, legged animals constantly adjust where they place their feet while maintaining balance and promoting overall safety. To achieve similar types of "safe adaptation" in complex terrains, this work supports the belief that walking robots need to be able to easily reason over the multiple levels of decision making (where to place feet, how to place feet at desired locations, how to control, *etc.*) to ultimately decide what behavior to execute. To this end, this work develops a framework that enables legged robots to execute complex locomotive behaviors that rapidly adapt to unexpected environmental scenarios by simultaneously adjusting the placement of their feet, planning whole body dynamic trajectories, and deriving feedback controllers that stabilize online.

This work is inspired by the issues related to the difficulty of dynamically and safely adapting prior approaches to account for unexpected changes in the environment. This work firmly supports the belief that this difficulty is directly related to treating footstep location selection, motion planning, and control as three independent steps. For example, consider a bipedal robot that experiences a perturbation and must instantaneously decide how to react (e.g., the system's stance foot begins slipping while walking over uneven terrains). Given a relatively large perturbation, a conventional planning and control framework would first re-plan desired feet locations, then plan motions that move the system's body correspondingly, and finally derive controllers that stabilize the system about the planned motions. However, in practice these individual steps are actually coupled: the choice of footstep location can result in either a relatively high or low cost motion plan, it may be arbitrarily difficult or easy to stabilize given the plan, *etc.* Most existing approaches simply ignore this coupling and thus end up executing reactive policies that are likely sub-optimal and possibly unsafe. Another possibility would be to evaluate several candidate planning and control sequences (e.g., with different footstep locations, motion plans, *etc*) and select the best. Though likely to produce better overall performance this approach is computationally prohibitive with respect to dynamic, real-time performance.

Biorobotics Lab, Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave, USA `(screws, sapana, mtravers)@andrew.cmu.edu`

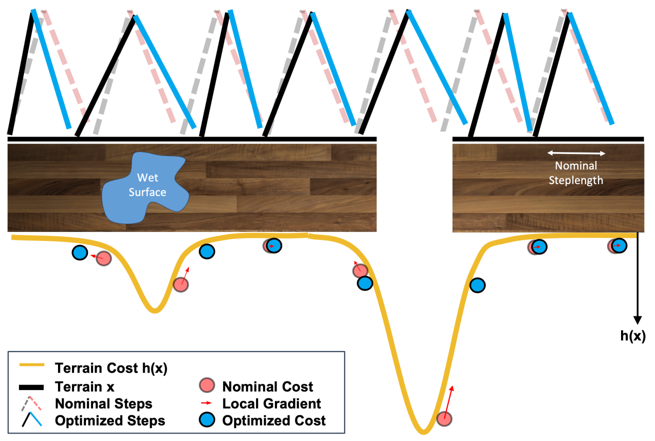

Fig. 1. Schematic diagram for a legged robot navigating rough terrain, shown with a mirrored terrain cost. Nominal foothold positions are iteratively adjusted toward attainable areas of lower terrain cost by a method of gradient descent. Overall multi-step cost outweighs local cost, evident when Step 4 slightly ascends the slope to assist with the steeper Step 5 descent.

To address this, this work presents a strategy that works conversely to prior methods. Specifically, we develop a novel approach for composing all three steps of planning and control (footsteps, motion, and feedback) concurrently by way of online quadratic programming, a smart discretization of walking motion, and imposing soft penalties on foothold locations directly into our cost function. The footstep soft penalties presented are based on continuous parameterized functions that represent terrain cost at foothold locations, given an environmental model that properly conveys the desirability of stepping at different terrain features and surfaces. By embedding terrain information into a numerical optimization tool that has proven successful for real-time planning, our method is able to adapt to new terrain information and avoid areas in rough terrains that are likely to cause the legged systems being investigated to fail. The resulting approach plans aggressive trajectories that avoid obstacles and safely adapts nominal walking motions to environments with significant mobility challenges.

The paper is structured as follows. Section II describes the related works. We lay out our method for the control of a hybrid system in Section III-A. Section III-B follows with the linearization of the hybrid dynamics between footsteps. The terrain cost is defined and this method is assembled in Section III-C using an iterative optimization approach. We present results for this method using a Compass Gait Walker (CGW) model in Section IV with proposed future work in V. Section VI derives the full linearization of a projected hybrid system using the analytical solution, along with the necessary computation sequence.

II. Background

Conventional approaches for legged robot planning and control in unstructured terrain can be broken into the three fundamental sub-problems: perception, planning, and control.

A. Terrain Perception / Map Modeling

Terrain perception involves sensing the environment to determine a terrain map and translate physical suitability into terrain cost map usable by the planner(s). Perception research has conventionally focused on the classification of terrains to determine suitability of foothold positioning. Significant work has advanced terrain classification [1] and the ability to generate terrain cost maps.

Researchers [2]–[5] demonstrated stereo camera and LIDAR sensor fusion to obtain terrain height maps. The terrain information then can be processed as an occupancy grid map [3], [4] to generate a terrain cost map or a binary confidence map [6], [7] based on terrain features (standard deviation of height, slope, roughness and height relative to the center of mass) and its interaction with the system (stability, safety, usability, etc). After a map is generated, a terrain cost generator is necessary to translate physical suitability into a measurable cost to be used in planning.

The terrain cost map can be turned into a set of constraints that a footstep planner must satisfy for path feasibility, which, in current optimization methods, undesirable states are modelled as barrier-like constraints, i.e. they are explicitly avoided [8]–[10]. This work represents such desirable and undesirable foothold locations using a continuous function that portrays areas of low- and high-cost locations, respectively, as depicted in Fig. 1. This model grants more flexibility in the depiction, not only addressing classic terrain features like friction and slope, but also capturing factors such as loudness of the step or associated degrees of probability based on visibility or measurement errors. Therefore, we introduce a "terrain cost" as a soft constraint in our cost function, enabling the planner to balance dynamics, energy consumed, and the environment all at once.

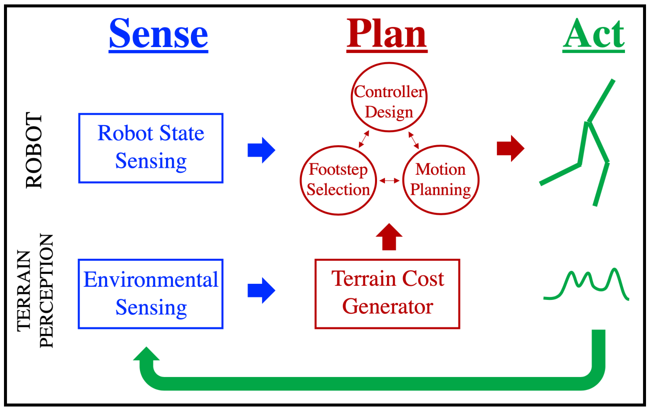

Fig. 2. Possible architecture for a legged robot. This work focuses on footstep selection, motion planning and feedback controller design, all three simultaneously and in real-time.

B. Hierarchical Planning and Control

Once terrain information is received, conventional planning and control frameworks then begin planning the system's desired motions This typically involves three subproblems: (i) footstep locations are first located, (ii) the robot coordinates its motion to place its feet at the locations, and (iii) an online controller regulates the system around the planned motions.

Footstep selection is most commonly solved using search-based methods [3], [4], [7], [11]–[14]. These discrete methods first determine suitable foothold positions given the robot's start and desired terminal positions. Search-based planners typically find the locally-optimal foothold positions by searching reachable spaces defined in part by the quasi-static stability of the system.

Though search-based planning has been successfully applied to different legged systems, there are several well-known issues with this approach. For improved speed, many prior works have focused on coarse discretizations of the search space used to find good foothold locations. However, coarse discretization leads to non-smooth body trajectories that require some degree of smoothing after the initial planning procedure. On the other hand, for fine discretizations it is difficult to determine foothold locations in real-time. In general, the main practical issue with this inherent tradeoff is that the appropriate level of representational fidelity for a given system and specific environment may be very difficult to determine in real-time. Once the footstep plan culminates, the hierarchical approaches proceed on to the motion-planning and controller derivation problems, an approach with inherent issues as described in Section I.

C. Combining Footstep Selection, Motion Planning, and Feedback Control

Recently a number of approaches [3], [15], [16] have combined footstep selection and motion planning into trajectory optimization formulations by directly incorporating the dynamics into the footstep planning. [17] showed the approach can work in near-real-time, but require a priori knowledge of the environment (analytical solutions of constraints). While these methods can yield both foothold and motion plans, they still do not consider the amount of feedback control effort it takes to control around the motion plan.

Differential Dynamic Programming (DDP) [18] is a class of approaches that natively produces not only the trajectory, but also the controller to track them. These methods typically function by making quadratic approximations of the cost-to-go function and then apply approximate dynamic programming to solve for trajectories and controllers to stabilize them using several iterative forward and backward passes. Numerical optimization via DDP and its computationally simpler adaptation, iterative linear quadratic regulator (iLQR) [19], have presented promising results in terms of solving planning problems in real-time [20]. [21]–[23] demonstrated the ability to fuse environmental information into their numerical optimization using DDP variants by planning with pure-state and state constraints. [24] extended hard-constrained DDP to legged locomotion, but additionally solves for switching time parameters, a step unnecessary in our formulation.

By using an iLQR formulation as the basis for our planner, we are able to simultaneously solve for foothold locations, dynamically feasible trajectories, and controllers that stabilize around that trajectory (see Fig 2). Our main innovation is a clever linearization of the hybrid system, allowing this work to do two things that has not been done in prior DDP/iLQR formulations for dynamic walking systems: (i) for an N-footstep sequence, decrease the backward pass from a fine discretization of NS segments to just N steps, speeding up the backwards pass by a factor of S (likely orders of magnitude faster depending on how continuous discretization S is chosen), (ii) project the system states onto a lower dimensional manifold for quicker calculations, (iii) put special emphasis on foothold selection such that "every" discrete point can be evaluated against the terrain cost map in real-time. We use an iLQR framework as the basis for a model predictive control (MPC) approach that allows "once-per-step" re-planning in real-time.

III. Technical Approach

The basis for the unified planning and control framework presented in this work consists of modeling techniques from the hybrid systems community, nonlinear analysis, and nonlinear control.

A. Hybrid Systems

We assume a hybrid system that can handle once-per-step control parameter updates (e.g. impulses $\mathbf{u}_\mathrm{I}$, gain selection $\mathbf{u}_\mathrm{K}$, *etc.*). We also assume that we can apply discrete impulsive control actions at the instant impact occurs. Examples of impulsive control includes instantaneous adjustment of joint angles and instantaneous forces created by toe push-off. We refer to the discrete toe-ground collision event as an impact map M, consisting of both non-conservative energy loss due to collision with the ground and the applied impulses to generate the post-collision state

$$\mathbf{q}^+ = M(\mathbf{q}^-, \mathbf{u}_\mathrm{I}). \tag{1}$$

We assume the hybrid system can be stabilized throughout its continuous swing phase by feedback control. Function ϕ will represent the stepping foot touching the terrain and therefore denote the end of continuous integration. We assume we have direct control over the gains of a continuous feedback controller, similar to a proportional-derivative (PD) control as per

$$\begin{aligned} u_i &= -k_p(\theta - \theta^d) - k_d(\dot{\theta} - \dot{\theta}^d) \\ &= -\begin{bmatrix} k_p & k_d \end{bmatrix} \begin{bmatrix} \bar{\theta} \\ \dot{\bar{\theta}} \end{bmatrix}. \end{aligned} \tag{2}$$

For state space $\mathbf{q} = \begin{bmatrix} \boldsymbol{\theta} & \dot{\boldsymbol{\theta}} \end{bmatrix}^\mathrm{T}$ of dimension n, the gain matrix will be denoted as $\mathbf{u}_\mathrm{K} = \begin{bmatrix} k_p & k_d \end{bmatrix}^\mathrm{T}$ for continuous dynamics f

$$\dot{\mathbf{q}} = f(\mathbf{q}, \mathbf{u}_\mathrm{K}). \tag{3}$$

Control $\mathbf{u}_{[\mathrm{n}]}$ is the vector of m once-per-step control parameter changes over a single step n, as depicted for a simple hybrid system in Fig. 3.

$$\mathbf{u}_{[\mathrm{n}]} = \begin{bmatrix} \mathbf{u}_\mathrm{I} & \mathbf{u}_\mathrm{K} \end{bmatrix}^\mathrm{T} \tag{4}$$

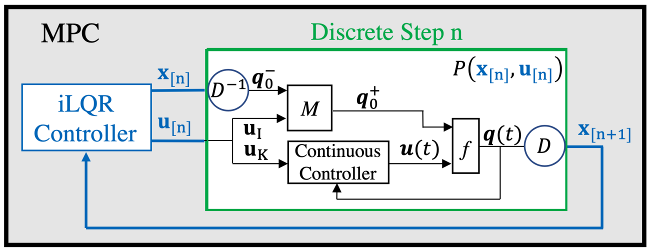

Fig. 3. Hybrid Dynamics with discrete controllable parameters $\mathbf{u}_{[\mathrm{n}]}$ tuned by an iLQR controller operating under an Model Predictive Control architecture. Discrete actions $\mathbf{u}_\mathrm{I}$ represent impulses or other such instantaneous actions occurring at impact map M. Adjustable gains $\mathbf{u}_\mathrm{K}$ are used to control the continuous phase f, initiated after impact mapping. Dimensionality reduction D and its respective inverse D^{-1} encountered leaving and entering the mapping. The entire discrete step n can be denoted by discrete projection mapping P. While N steps are planned by iLQR controller, only a single step is shown for clarity.

B. Nonlinear Analysis

For a system experiencing a periodic orbit, the entire trajectory can be captured by a single instant along a carefully-selected slice of the trajectory. This allows us to project the nonlinear dynamics of the the system with n states $\mathbf{q}$ onto the $n-1$ dimensional *surface of section* (*s.o.s.*) $\mathbf{x}$. The dimensionality reduction mapping from $\mathbf{q}$ to $\mathbf{x}$ will be called function D, with a defined inverse D^{-1}

$$\mathbf{x} = \mathrm{D}(\mathbf{q}) \qquad \mathbf{q} = \mathrm{D}^{-1}(\mathbf{x}). \tag{5}$$

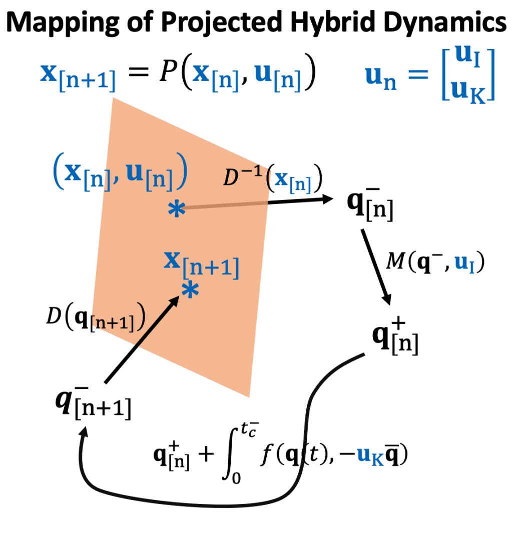
Mapping of Projected Hybrid Dynamics
$\mathbf{x}_{[n+1]} = P(\mathbf{x}_{[n]}, \mathbf{u}_{[n]})$
$\mathbf{u}_n = \begin{bmatrix}\mathbf{u}_I \\ \mathbf{u}_K\end{bmatrix}$
$(\mathbf{x}_{[n]}, \mathbf{u}_{[n]})$
$D^{-1}(\mathbf{x}_{[n]})$
$\mathbf{q}^-_{[n]}$
$M(\mathbf{q}^-, \mathbf{u}_I)$
$\mathbf{x}_{[n+1]}$
$D(\mathbf{q}_{[n+1]})$
$\mathbf{q}^+_{[n]}$
$\mathbf{q}^-_{[n+1]}$
$\mathbf{q}^+_{[n]} + \int_0^{t_c^-} f(\mathbf{q}(t), -\mathbf{u}_K\bar{\mathbf{q}})$

Fig. 4. Example projection mapping for a hybrid system with surface of section immediately before impact: dimensionality reduction D, impact mapping M, continuous dynamics f, and inverse dimensionality D^{-1}.

We define a projection mapping from one *s.o.s.* $\mathbf{x}_{[n]}$ to the next $\mathbf{x}_{[n+1]}$ as

$$\mathbf{x}_{[n+1]} = \mathrm{P}\left(\mathbf{x}_{[n]}, \mathbf{u}_{[n]}\right), \tag{6}$$

sometimes referred to as the return map or Poincare Map. This work defines the return map at the time immediately before contact t_c^- for the nth step with control action determined by (4). When $\mathbf{u}_{[n]}$ influences $\mathbf{x}_{[n]}$ such that $\mathbf{x}_{[n]} = \mathbf{x}_{[n+1]}$, the motion is at a fixed point of the return map associated with the closed-loop system dynamics, experiencing periodic behavior on the *s.o.s.*. An example mapping for a system defined with a reduced state on the *s.o.s.* (5), a single impact map (1), and a single continuous dynamics phase through integration of (3) is shown in Fig. 4 and described by

$$\begin{aligned} P\left(\mathbf{x}_{[n]}, \mathbf{u}_{[n]}\right) =& D\Bigg[M\left(D^{-1}\left(\mathbf{x}_{[n]}\right), \mathbf{u}_{[n]}\right) \\ &+ \int_{t^+_{c[n]}}^{t^-_{c[n+1]}} f\left(\mathbf{q}\left(\mathrm{t}, \mathbf{x}_{[n]}, \mathbf{u}_{[n]}\right), \mathbf{u}_{[n]}\right) dt\Bigg]. \end{aligned} \tag{7}$$

It is important to note that (7) is not a closed-form solution of the hybrid dynamics, but rather involves an integration until next contact, determined by

$$\phi\left(q\left(t^-_{c[n+1]}, \mathbf{x}_{[n]}, \mathbf{u}_{[n]}\right)\right) = 0. \tag{8}$$

For a walking system, (7) is equivalent to propagating the dynamics forward until the swing foot comes in contact with the terrain map (8) at $t^-_{c[n+1]}$.

We derive a discrete-time feedback controller to "regulate the system on the *s.o.s.*." The controller is returned from a modified iLQR algorithm. The iLQR approach functions by first computing the linearization of (6) about $\left(\mathbf{x}^*_{[n]}, \mathbf{u}^*_{[n]}\right)$

$$\begin{gathered} \mathbf{x}_{[n+1]} = \mathbf{x}^*_{[n+1]} + \mathbf{A}^*\left(\mathbf{x}_{[n]} - \mathbf{x}^*_{[\mathbf{n}]}\right) + \mathbf{B}^*\left(\mathbf{u}_{[n]} - \mathbf{u}^*_{[\mathbf{n}]}\right) \\ \mathbf{A}^* = \frac{\partial P\left(\mathbf{x}^*_{[n]}, \mathbf{u}^*_{[n]}\right)}{\partial \mathbf{x}_{[n]}}, \quad \mathbf{B}^* = \frac{\partial \mathrm{P}\left(\mathbf{x}^*_{[n]}, \mathbf{u}^*_{[n]}\right)}{\partial \mathbf{u}_{[n]}} \end{gathered} \tag{9}$$

and later use iLQR to tie multiple steps together. In (9), we denote the point evaluated at with *. Evaluation of the gradients of the projected dynamics (A^*, B^*) using the mapping shown in (7) is derived as

$$\frac{\partial P}{\partial \mathbf{x}_{[n]}} = \frac{\partial D}{\partial \mathbf{q}}\frac{\partial M}{\partial \mathbf{q}}\left[f(q(t))\frac{\partial t^-_{c[n+1]}}{\partial \mathbf{x}_{[n]}} + \frac{\partial \mathbf{q}(\mathrm{t})}{\partial \mathbf{x}_{[n]}}\right]\Bigg|_{\iota} \tag{10}$$

$$\frac{\partial P}{\partial \mathbf{u}_{[n]}} = \frac{\partial D}{\partial \mathbf{q}}\left[\frac{\partial M}{\partial \mathbf{q}}\left[f(q(t))\frac{\partial t^-_{c[n+1]}}{\partial \mathbf{u}_{[n]}} + \frac{\partial \mathbf{q}(\mathrm{t})}{\partial \mathbf{u}_{[n]}}\right] + \frac{\partial M}{\partial \mathbf{u}_{[n]}}\right]\Bigg|_{\iota} \tag{11}$$

and evaluated at $t = t^-_{c[n+1]}$, time of next ϕ contact. Full derivation and method for solving is shown in Section VI for the interested reader.

C. Nonlinear Planning and Control

To produce a set of appropriate footholds, a footstep selection planner first needs terrain cost information regarding desirability of stepping locations (as shown in Fig. 2). Locally-continuous terrain cost maps not only can incorporate more information into the terrain than categorical classifications (as discussed in Section II-A), but also enable methods such as gradient descent. A terrain cost point cloud could be generated for a landscape from some terrain-perception based on desirability of footstep locations. Local approximations of continuous cost functions are created using subsets of this data. Data is selected in the region of footstep impact and footstep adjustments will be made based on the local gradient. The creation of a terrain cost point cloud based on environmental information gathered is dependent upon the robot and task, which is beyond the scope of this paper. We assume that the terrain perception model is fast enough to generate the terrain cost map at the translational speed of the walking system.

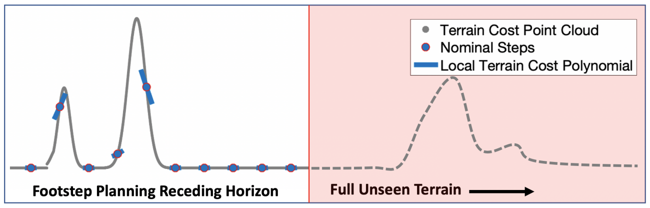

Fig. 5. Local 2^{nd} order terrain cost polynomials overlaid onto 1D terrain cost point cloud shown with a receding horizon.

A footstep cost polynomial is a local representation of the point cloud shape at the point of touchdown for a nominal step. Although many techniques can be used to generate the polynomial p_n, a custom least-square fit of form

$$x = g(\mathbf{x}) \tag{12}$$

$$H = \texttt{sign}\left(\zeta^*_{[n]}\right) \psi \, h_{[n]}\left(x^*_{[n]}\right) \tag{13}$$

$$\text{where } \zeta^*_{[n]} = \frac{\partial}{\partial x^*_{[n]}} h_{[n]}\left(x^*_{[n]}\right)$$

$$p_{[n]}(x) = x^2 + 2(H - x^*_{[n]})x + (x^*_{[n]} - 2H)x^*_{[n]} + h(x^*_{[n]}) \tag{14}$$

is used in this paper for simplicity and speed. (12) extracts step size x our of the state $\mathbf{x}$. The variables ψ and $\zeta^*_{[n]}$ and function $h_{[n]}(x^*_{[n]})$ shown in (13) represent an amplifying factor for the slope ($\psi > 0$), the derivative of the terrain cost and local fitted cost, respectively, evaluated at the nominal step size $x^*_{[n]}$. The amplifying factor ψ represents the relationship between the order of magnitude of a quadratic cost function (without including terrain cost) to that of the terrain cost. This work uses $\psi = 1$, but further refinement to include more restrictive barrier-like terrain boundaries is possible by increasing the slope of the local step polynomial by increasing $\psi > 1$.

Fig. 5 shows second order polynomials generated using data at $\pm 20\%$ of the nominal step size, but size of the region will be dependent upon the task and quality of the data. Polynomials must be at least 2nd order, as shown in (14), such that the terrain cost does not disappear at the Hessian of the iLQR value function.

Next, we combine both the discrete linearization of the hybrid dynamics (9) and the terrain cost (14) into a single iLQR optimization that will tune the gains of the feedback controller while simultaneously planning feasible foothold locations over the terrain map. The system is linearized about an initial sequence of trajectories and then new control increments are solved. This process is iterated until convergence on a local minima. Discrete iLQR is meant to solve problems of the form

$$\min_{\mathbf{u}(\cdot)} \left\{ \bar{\Phi}(\mathbf{x}_\mathrm{N}) + \sum_{\mathrm{n}=0}^{\mathrm{N}-1} \bar{\mathrm{L}}_\mathrm{n}\left(\mathbf{x}_\mathrm{n}, \mathbf{u}_\mathrm{n}\right) \right\}$$

$$\text{subject to: } \mathbf{x}_{\mathrm{n}+1} = \mathrm{P}(\mathbf{x}_\mathrm{n}, \mathbf{u}_\mathrm{n}) \;\; \mathbf{x}_0 = \mathbf{x}_{init}, \tag{15}$$

where $\mathrm{P}(\mathbf{x}_\mathrm{n}, \mathbf{u}_\mathrm{n})$ represents the nonlinear Poincare map integrated until contact similar to the representative example in (7). A quadratic terrain cost term is added to the original final and running costs of (15): $\bar{\Phi}_N(\mathbf{x}_\mathrm{N}) = \Phi_\mathrm{N}(\mathbf{x}_\mathrm{N}) + p_\mathrm{N}(\mathbf{x}_\mathrm{N})$, $\bar{\mathrm{L}}_\mathrm{n}(\mathbf{x}_\mathrm{n}, \mathbf{u}_\mathrm{n}) = \mathrm{L}_\mathrm{n}(\mathbf{x}_\mathrm{n}, \mathbf{u}_\mathrm{n}) + p_\mathrm{n}(\mathbf{x}_\mathrm{n})$. For legged locomotion, we found that heavy weighting of the final cost Φ_N (relative to intermediate costs L_n) is very important, allowing dynamic motion of the intermediate $N-1$ steps.

The iLQR derivation is a straight-forward minimization of the cost-to-go value function completed in several works [19], [25]. For our legged system, discrete point "n" represents a footstep and the entire projected hybrid step (7) is linearized per (9). Thus, $N = 10$ represents a 10-step look-ahead optimization. The iLQR feedforward control increment and feedback gain term are used to update the nominal control trajectory (gains, impulses, etc.) of the projected hybrid dynamics (7) using a line search scheme. This entire method is wrapped in an MPC scheme similar to [20] for fast updates while running online. From $\mathbf{x}_0$, plan N steps using a nominal sequence $(\mathbf{U}^*, \mathbf{X}^*) = (\mathbf{u}^*_0, \mathbf{x}^*_1), ..., (\mathbf{u}^*_{\mathrm{n}-1}, \mathbf{x}^*_\mathrm{n}), ...(\mathbf{u}^*_{\mathrm{N}-1}, \mathbf{x}^*_\mathrm{N})$, use iLQR to conform to terrain, and replace nominal with minimized plan $(\mathbf{U}, \mathbf{X})_{\min} \rightarrow (\mathbf{U}^*, \mathbf{X}^*)$. Move forward 1 step: replace $\mathbf{x}_1 \rightarrow \mathbf{x}_0$ and replan for N steps at every step.

This method does a few important things. (i) We use a method that does not require solving for time of contact for each incremental control adjustment [26], but instead solves for the gradient of the hybrid step directly. (ii) By discretizing iLQR over the projected dynamics (7), our method is able to tune the controllers for both the discrete (1) and continuous (3) dynamics of the hybrid system at once. (iii) These projected control updates (4) change the nature of the nominal gait, simultaneously adjusting footstep locations and the motion plan, to conform to the terrain map.

IV. Results

In this section, we complete an example of a Compass Gait Walker (CGW) [27], [28] walking over a terrain map. While simplified walking models [29], [30] could greatly reduce the computation time while fairly approximating the system dynamics, the assumption of massless legs oversimplifies most humanoid robots. Thus, our system has masses at the hip and knees with continuous PD-controlled torque in the hip and discrete toe-off impulses. We feel this combination properly demonstrates the use of iLQR feedforward control and feedback gains to optimally mix the discrete and continuous control efforts in a way to bring a controlled hybrid system to a locally-optimal and dynamically-feasible path.

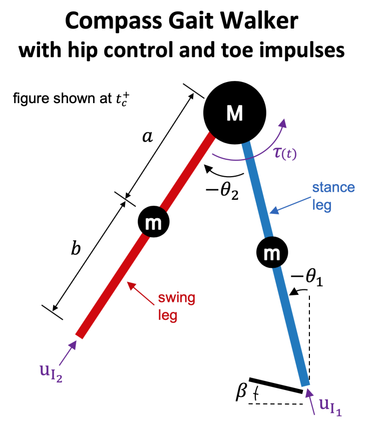

Fig. 6. Compass Gait Walker with hip torque and toe-off impulses. Parameters: $m = 5kg$, $M = 10kg$, $a = b = 0.5m$. Slope of terrain at stance foot: $\beta = 3°$. Front toe impulse $\mathrm{u}_{\mathrm{I}_1}$ occurs at t_c^- time instantaneously before touchdown (remove energy from system) and rear toe impulse $\mathrm{u}_{\mathrm{I}_2}$ occurs at t_c^+ time instantaneously after touchdown (add energy to system). Together, these impulses form the control actions to the discrete collision map M as in (1). Hip torque $\tau(t) = \begin{bmatrix} k_{p_1} & k_{p_2} & k_{d_1} & k_{d_2} \end{bmatrix} \begin{bmatrix} \bar{\theta}_1 & \bar{\theta}_2 & \dot{\bar{\theta}}_1 & \dot{\bar{\theta}}_2 \end{bmatrix}^T$ offers continuous control throughout the step as per (2). Together, these form the admissible once-per-step control actions $\mathbf{u}_\mathrm{n} = \begin{bmatrix} \mathrm{u}_{\mathrm{I}_1} & \mathrm{u}_{\mathrm{I}_2} & k_{p_1} & k_{p_2} & k_{d_1} & k_{d_2} \end{bmatrix}^\mathrm{T}$ shown in (4) and used in (6).

The system begins passively walking down a slope, but adds control effort to minimize the objective function (15) while maintaining dynamic stability. Given a nominal gait pattern, the nonlinear hybrid controller iteratively modifies individual step sizes to secure foothold positions in areas of lower terrain cost for a 10-step sequence. The gradient of the local cost polynomial (14) pushes the foothold location away from undesirable areas as shown in Fig. 7. The step sizes adjust as the individual costs are driven toward a local minima as shown in Fig. 8. The dynamics are preserved as the system modifies its stride length.

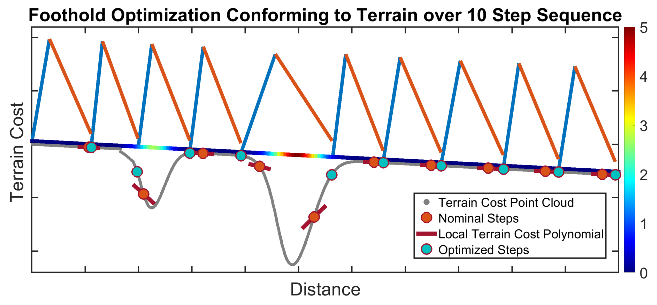

Fig. 7. Foothold optimization conforming to terrain cost map using gradient descent of local polynomials. Optimized results are shown after 4 iterations of iLQR. The controller modifies the step size to dynamically conform to the terrain, regaining the passive gait by Step 10.

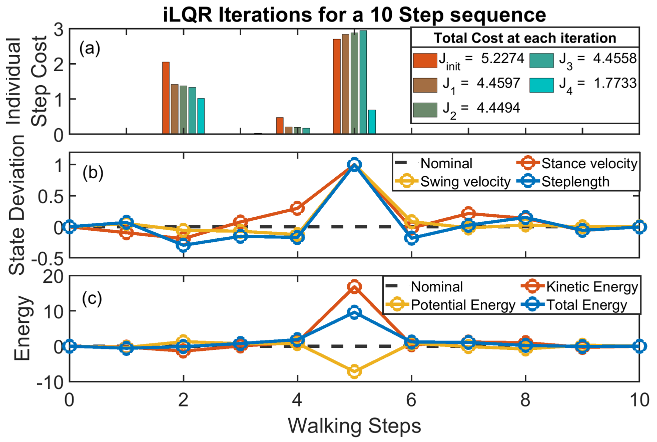

Fig. 8. (a) Footstep cost converging in 4 iterations of iLQR. In area of highest terrain cost (matching terrain in Fig. 7), local step cost increases slightly before dropping off drastically in iteration 4. (b) State deviations from nominal scaled by max deviation. This figure shows several shorter steps before a long step over a region of high local cost. Full state recovery happens at step 10. (c) Energy deviation from nominal. We see an increase in the KE of the system as it plans to take a bigger step.

For gradient computation, the analytical integration of (24)-(26) to generate the state and control gradients of the projected hybrid dynamics is on the computational complexity order of that of Forward Difference methods, but with better accuracy than that of Central Difference schemes when h_i is well-tuned. Additionally, with the use of symbolic computations of the terms in (24)-(26), common terms are not computed more than once, lending to faster overall speed. Our analytical method is 7.4× and 14.0× faster than the Forward and Central Difference algorithms, respectively, to compute the hybrid gradients. See Table I for the comparison of a CGW taking one step. The times referenced were calculated in MATLAB over an average of 10,000 runs on a computer with a 3.1 GHz Intel Core i7 processor and 16GB 2133 MHz ram.

TABLE I
COMPUTATIONAL COMPLEXITY, ACCURACY, AND TIME FOR NUMERICAL DIFFERENTIATION SCHEMES COMPARED WITH THE ANALYTICAL INTEGRATION METHOD SHOWN IN VI-C

∇ Method	A*	B*	Accuracy	Comp Time
Central Diff	$2n(n-1)$	$2nm$	$O(h^2)$	12.99 ms
Forward Diff	n^2	$n(m+1)$	$O(h)$	6.85 ms
Analytically	$(n-1)^2$	$(n-1)m$	$< O(h^2)$	0.93 ms

V. Conclusion and Future Work

In this work, we proposed an iterative optimization method that involves real-time planning over a receding horizon to address the problem of dynamic footstep planning for legged systems. Our main technical contribution was a closed-loop implementation of footstep planning and motion control in the context of control effort, facilitated by two supporting techniques. First, the projected hybrid dynamics allow for coupling of the discrete and continuous control actions, considering all motion until impact as a single event and pushes the hybrid control actions toward a common goal. By following the method in Section VI-C, we end up with both accurate gradients and very fast computations. Second, the terrain was treated as a cost map based on desirability of foothold positions. By using soft constraints, we allow the hybrid system to choose its terrain while balancing control effort through an optimization scheme utilizing gradient descent.

Although this approach is designed for generic hybrid systems, we have only tested it on a single walking robot model. More thorough simulations and experiments on different systems are necessary to verify the scalability and applications of the method onto more complex systems. The derivation shown in Section VI will be more complicated for systems involving multiple continuous phases, requiring multiple functions ϕ_i to represent the end of integration for the i^{th} continuous phase. We also believe this method is applicable to areas of rough terrain although it only has currently been tested on a uniform walking surface. We're additionally looking at including sum of squares programming to analyze the controller basin of attraction into future versions of this work. To verify the practical implementation of the methods mentioned in this paper, we intend to conduct hardware implementation on a 5-link biped and a quadraped (Minitaur).

VI. Appendix

In this section, the control gradient (11) for the projected hybrid dynamics referenced in (7) is derived. Although this derivation is specific to (7), the same method can be used generically to generate gradients for more complex hybrid systems. To keep the derivation compact, $|_\iota$ is used to represent the equation evaluation point $|_{t=t^-_{c[n+1]}}$. Parallel derivation can be done with respect to state gradient, whose results are shown in Section VI-C.

A. Projected linearization with respect to changes in $\mathbf{u}_{[n]}$

We begin by applying the chain rule to the gradient of (9)

$$\begin{aligned}\frac{\partial P\left(\mathbf{x}_{[n]},\mathbf{u}_{[n]}\right)}{\partial \mathbf{u}_{[n]}} &= \frac{\partial}{\partial \mathbf{u}_{[n]}} D\left(\mathbf{q}\left(t^-_{c[n+1]},\mathbf{x}_{[n]},\mathbf{u}_{[n]}\right)\right) \\ &= \frac{\partial D}{\partial \mathbf{q}}\left[f\left(\mathbf{q}(\mathrm{t})\right)\frac{\partial t^-_{c[n+1]}}{\partial \mathbf{u}_{[n]}} + \frac{\partial \mathbf{q}(\mathrm{t})}{\partial \mathbf{u}_{[n]}}\right]\Bigg|_\iota . \end{aligned} \tag{16}$$

Expanding $\left.\frac{\partial \mathbf{q}(\mathrm{t})}{\partial \mathbf{u}_{[n]}}\right|_\iota$ from (16)

$$\begin{aligned}\left.\frac{\partial \mathbf{q}(\mathrm{t})}{\partial \mathbf{u}_{[n]}}\right|_\iota =& \frac{\partial}{\partial \mathbf{u}_{[n]}}\Bigg[M\left(\mathbf{q}\left(\mathrm{t}^-_{c[n]},\mathbf{x}_{[n]}\right),\mathbf{u}_{[n]}\right) \\ &+ \int_{t^+_{c[n]}}^{t} f\left(\mathbf{q}(\tau),\mathbf{u}_{[n]}\right)d\tau\Bigg]_\iota \\ =& \frac{\partial M}{\partial \mathbf{u}_{[n]}} + \left[\frac{\partial}{\partial \mathbf{u}_{[n]}}\int_{t^+_{c[n]}}^{t} f\left(\mathbf{q}(\tau),\mathbf{u}_{[n]}\right)d\tau\right]\Bigg|_\iota . \end{aligned} \tag{17}$$

When applying the Leibniz integral rule to the integral in (17)

$$\begin{aligned}&\left.\frac{\partial}{\partial \mathbf{u}_{[n]}}\int_{t^+_{c[n]}}^{t} f\left(\mathbf{q}(\tau),\mathbf{u}_{[n]}\right)d\tau\right|_\iota \\ =& \Bigg[f\left(\mathbf{q}(t),\mathbf{u}_{[n]}\right)\frac{\partial t}{\partial \mathbf{u}_{[n]}} - f\left(\mathbf{q}(t^+_{c[n]}),\mathbf{u}_{[n]}\right)\frac{\partial t^+_{c[n]}}{\partial \mathbf{u}_{[n]}} \\ &+ \int_{t^+_{c[n]}}^{t}\Bigg[\frac{\partial f\left(\mathbf{q}(\tau)\right)}{\partial \mathbf{q}}\left(\frac{\partial \mathbf{q}(\tau)}{\partial \tau}\frac{\partial \tau}{\partial \mathbf{u}_{[n]}} + \frac{\partial \mathbf{q}(\tau)}{\partial \mathbf{u}_{[n]}} + \frac{\partial \mathbf{q}(\tau)}{\partial \mathbf{x}_{[n]}}\frac{\partial \mathbf{x}_{[n]}}{\partial \mathbf{u}_{[n]}}\right) \\ &+ \frac{\partial f\left(\mathbf{q}(\tau)\right)}{\partial \mathbf{u}_{[n]}}d\tau\Bigg]\Bigg]\Bigg|_\iota \\ =& \left[\int_{t^+_{c[n]}}^{t}\frac{\partial f\left(\mathbf{q}(\tau)\right)}{\partial \mathbf{q}}\frac{\partial \mathbf{q}(\tau)}{\partial \mathbf{u}_{[n]}} + \frac{\partial f\left(\mathbf{q}(\tau)\right)}{\partial \mathbf{u}_{[n]}}d\tau\right]\Bigg|_\iota , \end{aligned} \tag{18}$$

most of the expanded derivatives go to zero as seen in (18). Substituting (18) into (17) yields

$$\left.\frac{\partial \mathbf{q}(t)}{\partial \mathbf{u}_{[n]}}\right|_\iota = \left[\frac{\partial M}{\partial \mathbf{u}_{[n]}} + \int_{t^+_{c[n]}}^{t}\frac{\partial f\left(\mathbf{q}(\tau)\right)}{\partial \mathbf{q}}\frac{\partial \mathbf{q}(\tau)}{\partial \mathbf{u}_{[n]}}d\tau + \frac{\partial f\left(\mathbf{q}(\tau)\right)}{\partial \mathbf{u}_{[n]}}\right]\Bigg|_\iota . \tag{19}$$

The second fundamental theorem of Calculus states

$$\begin{aligned}\frac{d}{dt}\frac{\partial \mathbf{q}(t)}{\partial \mathbf{u}_{[n]}} &= \frac{d}{dt}\int_{t^+_{c[n]}}^{t}\frac{\partial f\left(\mathbf{q}(\tau)\right)}{\partial \mathbf{q}}\frac{\partial \mathbf{q}(\tau)}{\partial \mathbf{u}_{[n]}} + \frac{\partial f\left(\mathbf{q}(\tau)\right)}{\partial \mathbf{u}_{[n]}}d\tau \\ &= \frac{\partial f\left(\mathbf{q}(\mathrm{t}),\mathbf{u}_{[n]}\right)}{\partial \mathbf{q}}\frac{\partial \mathbf{q}(\mathrm{t})}{\partial \mathbf{u}_{[n]}} + \frac{\partial f\left(\mathbf{q}(\mathrm{t}),\mathbf{u}_{[n]}\right)}{\partial \mathbf{u}_{[n]}}. \end{aligned} \tag{20}$$

Forward integration of the form $\dot{y} = Ay$ with initial conditions $\frac{\partial \mathbf{q}(t^+_{c[n]})}{\partial \mathbf{u}_{[n]}} = \frac{\partial M}{\partial \mathbf{u}_{[n]}}$ of the form $y(0) = c_0$.

B. Solve for collision gradients

The goal in this section is to find the unknown term $\frac{\partial t^-_{c[n+1]}}{\partial \mathbf{u}_{[n]}}$ from equation (16). Set function ϕ to represent the end of integration

$$\phi\left(q\left(t^-_{c[n+1]},\mathbf{x}_{[n]},\mathbf{u}_{[n]}\right)\right) = 0. \tag{21}$$

Taking gradient of ϕ with respect to $\mathbf{u}_{[n]}$,

$$\frac{\partial}{\partial \mathbf{u}_{[n]}}\phi\left(q\left(t^-_{c[n+1]},\mathbf{x}_{[n]},\mathbf{u}_{[n]}\right)\right) = 0$$

$$\begin{aligned}0 &= \left[\frac{\partial \phi}{\partial \mathbf{q}}\left(\frac{\partial \mathbf{q}(\mathrm{t})}{\partial \mathbf{u}_{[n]}} + \frac{\partial \mathbf{q}(\mathrm{t})}{\partial \mathbf{x}_{[n]}}\frac{\partial \mathbf{x}_{[n]}}{\partial \mathbf{u}_{[n]}} + \frac{\partial \mathbf{q}(\mathrm{t})}{\partial t}\frac{\partial t^-_{c[n+1]}}{\partial \mathbf{u}_{[n]}}\right)\right]\Bigg|_\iota \\ &= \left[\frac{\partial \phi}{\partial \mathbf{q}}f(\mathbf{q}(\mathrm{t}))\frac{\partial \mathrm{t}^-_{c[n+1]}}{\partial \mathbf{u}_{[n]}} + \frac{\partial \phi}{\partial \mathbf{q}}\frac{\partial \mathbf{q}(\mathrm{t})}{\partial \mathbf{u}_{[n]}}\right]\Bigg|_\iota \end{aligned} \tag{22}$$

and rearranging (22) reveals (23)

$$\frac{\partial t^-_{c[n+1]}}{\partial \mathbf{u}_{[n]}} = -\left[\left(\frac{\partial \phi}{\partial \mathbf{q}}f(\mathbf{q}(\mathrm{t}))\right)^{-1}\left(\frac{\partial \phi}{\partial \mathbf{q}}\frac{\partial \mathbf{q}(\mathrm{t})}{\partial \mathbf{u}_{[n]}}\right)\right]_\iota . \tag{23}$$

C. Linearization of the Nonlinear Projected Dynamics

This linearization method solves for the gradients of the projected dynamics using a numerical integration of the exact analytical gradient's time derivative using an ODE solver. By doing so, this method yields the numerical equivalent to the exact solution for a system where the analytical solution is unknown, yielding a faster and more accurate result than numerical gradient methods (finite difference, automatic differentiation, etc.), as shown empirically in Section IV.

The sequence to solve for $\frac{\partial P}{\partial \mathbf{x}_{[n]}}$ and $\frac{\partial P}{\partial \mathbf{u}_{[n]}}$ can be solved in parallel using a single forward integration of the dynamics, linearized about the pair $(\mathbf{x}^*_{[n]},\mathbf{u}^*_{[n]})$. We begin by integrating

$$\dot{\mathbf{q}} = f(\mathbf{q}(\mathrm{t})) \tag{24}$$

$$\frac{d}{dt}\frac{\partial \mathbf{q}(\mathrm{t})}{\partial \mathbf{x}_{[n]}} = \frac{\partial f\left(\mathbf{q}(\mathrm{t})\right)}{\partial \mathbf{q}}\frac{\partial \mathbf{q}(\mathrm{t})}{\partial \mathbf{x}_{[n]}} \tag{25}$$

$$\frac{d}{dt}\frac{\partial \mathbf{q}(\mathrm{t})}{\partial \mathbf{u}_{[n]}} = \frac{\partial f\left(\mathbf{q}(\mathrm{t}),\mathbf{u}_{[n]}\right)}{\partial \mathbf{q}}\frac{\partial \mathbf{q}(\mathrm{t})}{\partial \mathbf{u}_{[n]}} + \frac{\partial f\left(\mathbf{q}(\mathrm{t})\right)}{\partial \mathbf{u}_{[n]}} \tag{26}$$

from initial conditions $M\left(D^{-1}\left(\mathbf{x}_{[n]}\right)\right)$, $\frac{\partial M}{\partial \mathbf{q}}\frac{\partial D^{-1}}{\partial \mathbf{x}_{[n]}}$ and $\frac{\partial M}{\partial \mathbf{u}_{[n]}}$, respectively, until the next time of impact $t^-_{c[n+1]}$ to reveal $f(\mathbf{q}(\mathrm{t}))$, $\frac{\partial \mathbf{q}(\mathrm{t})}{\partial \mathbf{x}_{[n]}}$, and $\frac{\partial \mathbf{q}(\mathrm{t})}{\partial \mathbf{u}_{[n]}}$, evaluated at $t^-_{c[n+1]}$. Second, solve

$$\xi = -\left[\left(\frac{\partial \phi}{\partial \mathbf{q}}f(\mathbf{q}(\mathrm{t}))\right)^{-1}\frac{\partial \phi}{\partial \mathbf{q}}\right]_\iota \tag{27}$$

$$\frac{\partial t^-_{c[n+1]}}{\partial \mathbf{x}_{[n]}} = \xi\left.\frac{\partial \mathbf{q}(\mathrm{t})}{\partial \mathbf{x}_{[n]}}\right|_\iota \qquad \frac{\partial t^-_{c[n+1]}}{\partial \mathbf{u}_{[n]}} = \xi\left.\frac{\partial \mathbf{q}(\mathrm{t})}{\partial \mathbf{u}_{[n]}}\right|_\iota . \tag{28}$$

Finally, solve (29) and (30) for the gradients with respect to state and control

$$\frac{\partial P}{\partial \mathbf{x}_{[\mathrm{n}]}} = \frac{\partial D}{\partial \mathbf{q}}\frac{\partial M}{\partial \mathbf{q}}\left[f(q(t))\frac{\partial t^-_{c[\mathrm{n}+1]}}{\partial \mathbf{x}_{[\mathrm{n}]}} + \frac{\partial \mathbf{q}(\mathrm{t})}{\partial \mathbf{x}_{[\mathrm{n}]}}\right]\Bigg|_{\iota} \quad (29)$$

$$\frac{\partial P}{\partial \mathbf{u}_{[\mathrm{n}]}} = \frac{\partial D}{\partial \mathbf{q}}\left[\frac{\partial M}{\partial \mathbf{q}}\left[f(q(t))\frac{\partial t^-_{c[\mathrm{n}+1]}}{\partial \mathbf{u}_{[\mathrm{n}]}} + \frac{\partial \mathbf{q}(\mathrm{t})}{\partial \mathbf{u}_{[\mathrm{n}]}}\right] + \frac{\partial M}{\partial \mathbf{u}_{[\mathrm{n}]}}\right]\Bigg|_{\iota} . \quad (30)$$

Eqns. (26),(28),(30), derived in Section VI as (19),(20),(23), yield the control gradient. A parallel set of derivations can be completed to yield the state gradient, whose results are listed in (25),(28),(29) above.

References

[1] J. Christie and N. Kottege, "Acoustics based terrain classification for legged robots," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 3596–3603.

[2] M. F. Fallon, P. Marion, R. Deits, T. Whelan, M. Antone, J. McDonald, and R. Tedrake, "Continuous humanoid locomotion over uneven terrain using stereo fusion," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 881–888.

[3] A. W. Winkler, C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini, "Planning and execution of dynamic whole-body locomotion for a hydraulic quadruped on challenging terrain," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 5148–5154.

[4] C. Mastalli, I. Havoutis, A. W. Winkler, D. G. Caldwell, and C. Semini, "On-line and on-board planning and perception for quadrupedal locomotion," in *2015 IEEE International Conference on Technologies for Practical Robot Applications (TEPRA)*. IEEE, 2015, pp. 1–7.

[5] D. Kanoulas, A. Stumpf, V. S. Raghavan, C. Zhou, A. Toumpa, O. Von Stryk, D. G. Caldwell, and N. G. Tsagarakis, "Footstep Planning in Rough Terrain for Bipedal Robots Using Curved Contact Patches," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–9.

[6] J. Chestnutt, "Navigation planning for legged robots," Ph.D. dissertation, Citeseer, 2007.

[7] A. Winkler, I. Havoutis, S. Bazeille, J. Ortiz, M. Focchi, R. Dillmann, D. Caldwell, and C. Semini, "Path planning with force-based foothold adaptation and virtual model control for torque controlled quadruped robots," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 6476–6482.

[8] B. Plancher, Z. Manchester, and S. Kuindersma, "Constrained unscented dynamic programming," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 5674–5680.

[9] A. Agrawal and K. Sreenath, "Discrete Control Barrier Functions for Safety-Critical Control of Discrete Systems with Application to Bipedal Robot Navigation." in *Robotics: Science and Systems*, 2017.

[10] J. Chen, W. Zhan, and M. Tomizuka, "Autonomous Driving Motion Planning with Constrained Iterative LQR," *IEEE Transactions on Intelligent Vehicles*, 2019.

[11] J. Z. Kolter, M. P. Rodgers, and A. Y. Ng, "A control architecture for quadruped locomotion over rough terrain," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 811–818.

[12] M. A. Arain, I. Havoutis, C. Semini, J. Buchli, and D. G. Caldwell, "A comparison of search-based planners for a legged robot," in *9th International Workshop on Robot Motion and Control*. IEEE, 2013, pp. 104–109.

[13] D. Belter, P. Łabecki, and P. Skrzypczyński, "Adaptive motion planning for autonomous rough terrain traversal with a walking robot," *Journal of Field Robotics*, vol. 33, no. 3, pp. 337–370, 2016.

[14] P. Fankhauser, M. Bjelonic, C. D. Bellicoso, T. Miki, and M. Hutter, "Robust rough-terrain locomotion with a quadrupedal robot," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[15] B. BAceituno-Cabezas, C. Mastalli, H. Dai, M. Focchi, A. Radulescu, D. G. Caldwell, J. Cappelletto, J. C. Grieco, G. Fernández-López, and C. Semini, "Simultaneous contact, gait, and motion planning for robust multilegged locomotion via mixed-integer convex optimization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2531–2538, 2017.

[16] A. W. Winkler, F. Farshidian, M. Neunert, D. Pardo, and J. Buchli, "Online walking motion and foothold optimization for quadruped locomotion," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5308–5313.

[17] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, 2018.

[18] D. Mayne, "A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems," *International Journal of Control*, vol. 3, no. 1, pp. 85–95, 1966.

[19] W. Li and E. Todorov, "Iterative linear quadratic regulator design for nonlinear biological movement systems." in *ICINCO (1)*, 2004, pp. 222–229.

[20] M. Neunert, C. De Crousaz, F. Furrer, M. Kamel, F. Farshidian, R. Siegwart, and J. Buchli, "Fast nonlinear model predictive control for unified trajectory optimization and tracking," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1398–1404.

[21] A. Sideris and L. A. Rodriguez, "A riccati approach to equality constrained linear quadratic optimal control," in *Proceedings of the 2010 American Control Conference*. IEEE, 2010, pp. 5167–5172.

[22] J. van den Berg, "Iterated LQR smoothing for locally-optimal feedback control of systems with non-linear dynamics and non-quadratic cost," in *2014 American Control Conference*. IEEE, 2014, pp. 1912–1918.

[23] M. Giftthaler, F. Farshidian, T. Sandy, L. Stadelmann, and J. Buchli, "Efficient kinematic planning for mobile manipulators with non-holonomic constraints using optimal control," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3411–3417.

[24] F. Farshidian, M. Neunert, A. W. Winkler, G. Rey, and J. Buchli, "An efficient optimal planning and control framework for quadrupedal locomotion," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 93–100.

[25] A. Sideris and J. E. Bobrow, "An efficient sequential linear quadratic algorithm for solving nonlinear optimal control problems," in *Proceedings of the 2005, American Control Conference, 2005.* IEEE, 2005, pp. 2275–2280.

[26] F. Farshidian, M. Kamgarpour, D. Pardo, and J. Buchli, "Sequential linear quadratic optimal control for nonlinear switched systems," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 1463–1469, 2017.

[27] T. Anstensrud, "2-d passive compass biped walker," Ph.D. dissertation, Masters thesis, NTNU, 2013.

[28] K. Byl and R. Tedrake, "Approximate optimal control of the compass gait on rough terrain," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 1258–1263.

[29] P. A. Bhounsule, "Control of a compass gait walker based on energy regulation using ankle push-off and foot placement," *Robotica*, vol. 33, no. 6, pp. 1314–1324, 2015.

[30] H. R. Vejdani, A. Wu, H. Geyer, and J. W. Hurst, "Touch-down angle control for spring-mass walking," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 5101–5106.