

Jumping Over Obstacles with MIT Cheetah 2*

Hae-Won Park¹, Patrick M. Wensing¹ and Sangbae Kim

ARTICLE INFO

Keywords:
Legged Locomotion
Quadruped Robots
Sensor-Based Planning

ABSTRACT

This paper presents a planning framework for jumping over obstacles with quadruped robots. The framework accomplishes planning via a structured predictive control strategy that combines the use of heterogeneous simplified models over different prediction time scales. A receding multi-horizon predictive controller coordinates the approach before the jump using a kinematic point-mass model. Consideration of the optimal value function over different planning horizons enables the system to select an appropriate number of steps to take before jumping. The jumping motion is then tailored to the sensed obstacle by solving a nonlinear trajectory optimization problem. The solution of this problem online is enabled by exploiting the analyticity of the flow map for a planar bounding template model under polynomial inputs. By planning with this combination of models, MIT Cheetah 2 is shown to autonomously jump over obstacles up to 40 cm in height during high-speed bounding. Untethered results showcase the ability of the method to automatically adapt to obstacles of different heights and placements in a single trial.

1. Introduction

Quadrupedal animals exhibit remarkable versatility in unstructured environments. Whether cornering to evade a predator in an open field or leaping to cross a cavernous gap, biological systems display a seamless capacity to plan and execute complex motions that are tailored to the terrain and task at hand. Although quasi-static traversal strategies may be appropriate to negotiate a range of mild terrain irregularities, the most challenging terrains (e.g., gaps, large obstacles) will require future robots to skillfully plan and execute dynamic behaviors. With this motivation, this paper presents a new set of algorithms that enable dynamic and autonomous jumping over obstacles through online optimization in the MIT Cheetah 2. Key results are shown in Fig. 1.

Recent advances in a diverse set of experimental platforms encourage the use of the quadrupedal morphology to study dynamic maneuverability. Boston Dynamic's hydraulic quadrupeds BigDog, LS3, and the early version of Spot provided impressive dynamic walking on a variety of surfaces [50]. The ETH ANYmal robots have displayed a wide array of gaits and top speeds up to 1.5 m/s with the incorporation of series elastic actuation in their legs [25, 26]. IIT's HyQ [52] has shown the capacity for high-power movements via hydraulic actuation and can run with a speed of 2.0 m/s. The all-electric MIT Cheetah 1 [53] and Cheetah 2 [47] robots have shown the capabilities of electric DC motors to enable high-speed locomotion up to 6.4 m/s [47]. Follow-on platforms MIT Cheetah 3 [5], and Mini Cheetah [29] have shown speeds up to 3.0 m/s [14] and 3.7 m/s [30], respectively. Despite these more recent hardware platforms, Cheetah 2 still possesses unique advantages for jumping over large obstacles while running at speed, as Mini Cheetah has inherent



Figure 1: Autonomous jumping over obstacles with MIT Cheetah 2 using the proposed planning framework. The obstacle cleared in this picture is 40cm high, 80% of the leg length of Cheetah 2.

size limitations, while the design of Cheetah 3 is optimized for load carrying as opposed to high-speed operation.

Strategies for handling challenging terrain in legged robots have been split between perception-free reactive approaches and perception-driven planning-based approaches. Perception-free approaches present some inherent limitations, but have shown robustness benefits both individually and when combined with planning-driven gait adaptation. Park and Grizzle [45] presented a state-based approach to dynamic biped walking by switching control policies following terrain disturbances. Liu et al. [34] present a template-based gait library for humanoids to traverse unperceived uneven terrain. While these methods do not require terrain knowledge prior to contact, they do require contact detection, for which reliable algorithms are available [6]. Other methods have focused on designing gaits that are robust to terrain disturbances, for example via the design of terrain-robust trajectories [12] or robust step-to-step controllers [42]. More recently, methods have shown the promise of offline learning to transform libraries of open-loop stabilizing trajectories to terrain robust closed-loop stabilizing policies [11].

Many other studies have focused on planning-based quasi-

* This work was supported from the DARPA M3 program and from the Korean Agency for Defense Development under the contract UD140073ID.

✉ haewonpark@kaist.ac.kr (H. Park); pwensing@nd.edu (P.M. Wensing); sangbae@mit.edu (S. Kim)

ORCID(s): 0000-0002-9041-5175 (P.M. Wensing)

¹ Authors contributed equally

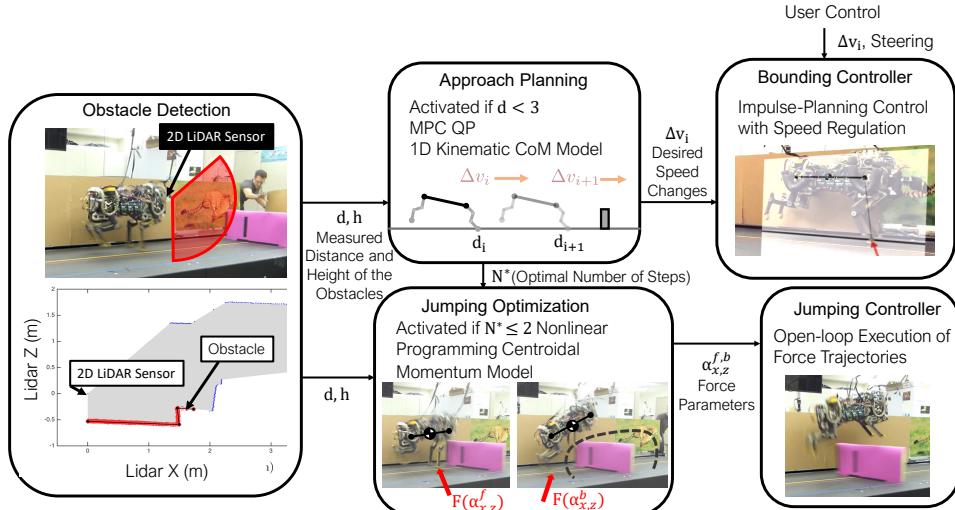


Figure 2: Overview of the framework for autonomous jumping. Obstacle detection, approach planning, and jump optimization are the three main new algorithms discussed in this work.

static locomotion strategies to traverse more challenging uneven terrains. Kalakrishnan et al. [28] proposed learning algorithms for foothold selection based on expert demonstrations using terrain templates. AnyMAL has shown an ability to handle moderately structured terrain from closed-loop perception-based gait design [17]. Likewise, researchers working on HyQ studied the use of stereo vision to build a height map of the surrounding terrain [4]. These terrain maps were used online to modify the **Central Pattern Generators (CPGs)** that governed cyclical leg motions, enabling rough terrain with obstacles up to 9.5 cm in height to be traversed. Reflexes layered on top of foot CPGs have also shown the ability to provide static ambling over obstacles up to 11cm in height [20], while other heuristic strategies have shown how perception-free reflex-based systems can be complemented with information from vision [21].

Advances in problem formulations for gait optimization [1] have allowed more complex footstep planning problems to be addressed using mixed-integer planning techniques with convex relaxations of centroidal dynamics models. Other strategies based on foothold search have shown impressive performance when ignoring dynamics during planning and using whole-body control to realize selected footholds [27]. These approaches have obvious computational benefits, but are limited when it comes to highly dynamic movements. To enable consideration of the full-body dynamics, strategies based on gait libraries (e.g., [41]) provide one alternative to simplify online computation demands.

Work in simulation has studied the capabilities of aperiodic dynamic movements to enable clearance of comparatively more extreme terrain obstacles. Wong and Orin [61] proposed the use of binary search for a small set of control parameters to produce standing jumps over obstacles. Coros et al. [10] developed an extensive trajectory dictionary to perform running jumps over a variety of large gaps. Krasny and Orin [32] applied an evolutionary algorithm to optimize running jumps during quadruped galloping. These

motions have also been studied in a limited capacity for humanoids [13, 58] by using simple models for trajectory optimization. Still, across each of these studies, the resultant controllers have required extensive offline optimization and have not demonstrated viability for use in experimental hardware. The same is true for demonstrations of reinforcement learning [24], which face challenges in sim-to-real transfer [54].

Recent demonstrations of dynamic behaviors in quadruped robots have shown the promise of model-predictive control (MPC) for online application in experimental hardware. Control based on single-rigid body models has led to a wide set of results [7, 14, 60] including diverse gaits and robust stair climbing in Cheetah 3 [14, 5]. Recent results also show early viability of solving more complex predictive control problems online through whole-body MPC [39, 40]. Optimal tradeoffs between model complexity and computational runtime yet remain an important design choice.

The main contribution of this paper is a new framework that accomplishes planning for autonomous jumping via a structured predictive control strategy combining heterogeneous simplified models over different prediction time scales. An initial version of this work appeared at RSS 2015 [46]. While the framework remains the same, this extended journal article includes an updated presentation and an expanded set of results. Of note, it features the presentation of untethered results jumping over multiple obstacles in a single trial, which were only previously shown in a YouTube video².

2. Overview of the Framework

The proposed framework consists of three main algorithms (Fig. 2): obstacle detection, approach planning, and jumping optimization. These three algorithms interface with a bounding controller [47] to provide an overall solution for autonomous jumping.

The obstacle detection algorithm perceives information

²https://youtu.be/_1uhn7TLfWU

about its surrounding environment using a Hokuyo UTM-30LX-EW laser range finder as shown in Fig. 2. This sensor provides point-cloud data in the sagittal plane with an angular resolution of 0.25° . Following each scan, a simple line segmentation algorithm based on [49, 51] is used to detect the ground plane and the front face of the obstacle. With the detected ground plane and front face of the obstacle, the distance d and height h of the obstacle can be inferred for use in the approach and jumping algorithms.

The design of the algorithms for approach planning and jumping optimization is based on mathematical models with different levels of abstraction of the robot. These models were selected to represent the robot with different degrees of complexity while satisfying stringent online computation requirements for each task. These design choices enable the algorithms to complete in the fraction of a second that is necessary to successfully jump over obstacles on the treadmill.

The Approach Planning selects changes in running speed Δv_i to position the robot into a safe region for jumping. The algorithm is activated when the distance to the obstacle becomes less than 3 meters. To select these desired changes in speed, an MPC strategy considers planning with a 1D kinematic Center of Mass (CoM) model. As a key benefit, this abstraction choice enables a convex MPC formulation that enjoys efficient and fast solution.

With the approach planning placing the robot in position for a jump, the jumping optimization then crafts a feasible jumping trajectory to clear the sensed obstacle. A planar centroidal model with a lumped mass and inertia is used to represent the evolution of the pitch and the CoM position under the application of ground reaction forces. Considering a polynomial parameterization of the force profiles with respect to time, this model admits closed-form solutions to the equations of motion. These closed-form solutions remove the need for numerical integration to evaluate the trajectory, which accelerates online trajectory optimization. The output of the optimization is a set of desired scaling coefficients $\alpha_{x,y}^{f,b}$ for ground force profiles of the front and hind legs.

As shown on the right side of Fig. 2, the planned actions from these algorithms were then realized using a bounding controller [47]. This bounding controller is able to track desired speed changes from the approach planning algorithm and to reject large perturbations in states that occur after landing a jump.

The following sections detail these algorithms further. Section 3 discusses the approach planning algorithm and its formulation as a multiple-horizon optimal control problem that is solved in an MPC fashion. Section 4 discusses the jump optimization formulation and the strategies used to solve it quickly online. Section 5 then presents the main results from experiments that are shown in the video attachment.

3. Approach Planning

This section details the model-predictive control (MPC) strategy used to get into position for a successful jump. A simple kinematic point-mass model is used for receding hori-

zon optimization to coordinate changes in speed from step to step. As opposed to conventional approaches to MPC that employ an infinite horizon or a fixed receding horizon, this work considers a family of receding horizons for the robot to determine the number of steps to take before jumping.

3.1. MPC for Approach Planning

At a given step, the distance to the obstacle is denoted as $d_0 \in \mathbb{R}$, with the initial velocity of the robot $v_0 \in \mathbb{R}$. At each following step i , the state of the robot state is similarly abstracted by $\mathbf{x}_i \in \mathbb{R}^2$

$$\mathbf{x}_i = [d_i \ v_i]^T. \quad (1)$$

In order to manage the robot's approach, we consider speed changes Δv_i at each step. It is assumed that the speed change Δv_i occurs over the duration of the step, providing an average speed of $v_i + \frac{\Delta v_i}{2}$. Under this assumption, given a nominal step period T^* , the state then evolves as

$$\mathbf{x}_{i+1} = \underbrace{\begin{bmatrix} 1 & -T^* \\ 0 & 1 \end{bmatrix}}_{\mathbf{A}} \mathbf{x}_i + \underbrace{\begin{bmatrix} -\frac{1}{2}T^* \\ 1 \end{bmatrix}}_{\mathbf{B}} \Delta v_i. \quad (2)$$

These dynamics are a drastic simplification of the full robot dynamics. However, by re-planning in a model-predictive fashion, we find that this simple abstraction is a suitable one to coordinate the approach.

Toward selecting the speed adjustments Δv_i , consider an N step approach leading up to the obstacle, for some fixed $N \in \mathbb{N}^+$. We seek adjustments $\Delta v_{0:(N-1)}$ that minimally disturb the nominal gait while landing the final state \mathbf{x}_N within a desired goal region. The goal is defined here by lower and upper bounds:

$$\underline{\mathbf{x}}_N = [d_F \ \underline{v}_F]^T, \quad \bar{\mathbf{x}}_N = [\bar{d}_F \ \bar{v}_F]^T \quad (3)$$

with a desired final state in the middle of this goal region

$$\mathbf{x}_N^d = \frac{\underline{\mathbf{x}}_N + \bar{\mathbf{x}}_N}{2}. \quad (4)$$

Given this desired end state, a simple linear MPC problem can be formulated to find velocity adjustments over the N step horizon:

$$\begin{aligned} V^*(\mathbf{x}_0, N) = \min_{\Delta v_{0:(N-1)}, \mathbf{x}_{1:N}} & (\mathbf{x}_N - \mathbf{x}_N^d)^T \mathbf{Q}_F (\mathbf{x}_N - \mathbf{x}_N^d) \\ & + \frac{1}{N} \sum_{i=0}^{N-1} r_i \Delta v_i^2 \end{aligned} \quad (5)$$

$$\begin{aligned} \text{s.t. } & \mathbf{x}_{i+1} = \mathbf{A} \mathbf{x}_i + \mathbf{B} \Delta v_i \\ & \underline{\mathbf{x}}_N \leq \mathbf{x}_N \leq \bar{\mathbf{x}}_N \\ & \underline{v} \leq v_i \leq \bar{v} \\ & |\Delta v_i| \leq \beta v_i \end{aligned}$$

where $V^*(\mathbf{x}_0, N)$ denotes the optimal cost to go, also known as the value function, for an N -step horizon starting from

state \mathbf{x}_0 . The bounds $\underline{v} > 0$ and $\bar{v} > 0$ provide velocity limits on the gait, and $\beta \in (0, 1)$ limits the relative acceleration during any given step. To penalize gait changes that occur closer to the obstacle, the cost scalars r_i are selected as

$$r_i = \max(\mu^{i-(N-1)}, r_{min}) \quad (6)$$

for some μ and r_{min} with $\mu > 1$ and $0 < r_{min} < 1$. This strategy encourages larger accelerations to occur further from the obstacle in order to provide additional time to recover from transient effects. The quadratic cost on the final state, with $\mathbf{Q}_F = \mathbf{Q}_F^T$ positive definite, rewards final states that have a margin to remain in the goal when unmodelled effects in the full system disturb the optimized trajectories. Overall, the formulation is a quadratic program (QP) with $3N$ variables and thus can be efficiently solved in a fraction of a millisecond for the planning horizons considered here.

Given the bounds on the speed changes and states, it is possible that there is no feasible approach for a given initial state \mathbf{x}_0 and horizon N . In this case, we define $V^*(\mathbf{x}_0, N) = \infty$. While most applications of MPC seek to regulate the long-term state of a system and can employ an infinite planning horizon, the horizon here represents the number of steps to take before jumping, which is not known a priori. To address this challenge, we simultaneously consider a family of receding horizons.

3.2. Multiple-Horizon MPC

Given the bounds on the forward velocities and final states, $V^*(\mathbf{x}_0, N) < \infty$ only if

$$\frac{d_0 - \bar{d}_F}{\bar{v}_F} \leq N \leq \frac{d_0 - \underline{d}_F}{\underline{v}_F}, \quad (7)$$

where the lower step number bound is derived from traveling the shortest distance to the goal region at the maximum speed, and the upper bound follows similarly. This condition introduces a finite range for N to be searched over, given the initial state of the system. The optimal approach length N^* can then be found through evaluation of $V^*(\mathbf{x}_0, N)$ for each N in this range. Figure 3 shows the optimal number of steps to be taken before the obstacle as a function of the initial state. Algorithm parameters for this example are given in Table 1.

Advances in linear MPC using interior-point [57] and active-set [18] solvers have significantly decreased the computational overhead of online MPC approaches. Here, to solve the QP for $V^*(\mathbf{x}_0, N)$, the open-source package qpOASES [19] was used. This package provides a parametric active-set solver that is particularly well suited to MPC. With this solver, QPs can be solved in 250 μ s on average for the horizons considered. The online solution, however, was largely a matter of convenience, as the low dimensionality of the state space could likewise have enabled explicit MPC solutions (see, e.g., [3]).

4. Real-time Jumping Trajectory Generation

This section presents methods for the online generation of a jumping trajectory that is tailored to the sensed obsta-

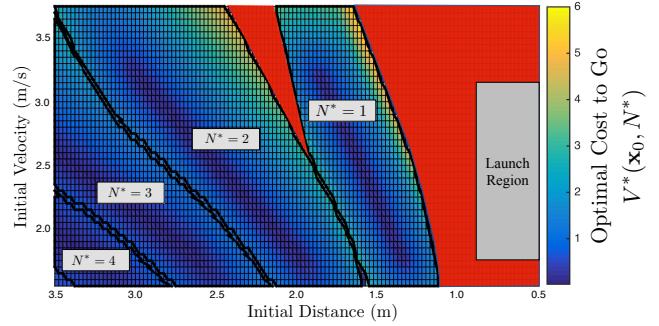


Figure 3: Optimal number of steps to the goal based on the initial state. Color indicates the optimal cost to go $V^*(\mathbf{x}_0, N^*)$. Red regions represent infeasible regions for the start state (i.e., $V^*(\mathbf{x}_0, N) = \infty$ for all N).

Parameter	β	μ	r_{min}	\mathbf{Q}_F
Value	0.3	1.1	0.4	diag(2.5, 1)
Variable Bound	d_F	v_F	v_i	
Min	0.5 m	1.8 m/s	1.0 m/s	
Max	0.9 m	3.2 m/s	4.0 m/s	

Table 1
Parameters for the approach adjustment algorithm

cle. A simplified single rigid-body model with lumped mass and inertia is considered for real-time generation of a feasible trajectory. Contact forces serve as the inputs to this model, while its configuration includes the position of Center of Mass and the body attitude. Due to the simple form of this model and its ability to describe a broad set of dynamic behaviors, it has been used widely for control design and motion planning in quadrupeds [46, 47, 44, 60, 14, 5]. In this paper, we restrict the application of the model to motion in the sagittal plane. This restriction is justified since the bounding gait used is dominated by sagittal plane dynamics.

Considering this restriction to the sagittal plane, a closed-form solution for the equations of motion can be given analytically when forces are polynomial in time. Chaining this observation across phases of stance and flight then enables the flow for the hybrid dynamics of the model to be evaluated without numerical integration. This approach accelerates online trajectory optimization of force profiles. In the hardware, these force profiles are ultimately generated using joint torques of the robot through a Jacobian Transpose mapping (e.g., as in [44]) resulting in accurate force delivery [59]. The following sections introduce the closed-form solution of the equations of motion for the planar model and describe how this solution is used for trajectory generation.

4.1. Simplified Model and Temporal Gait Pattern of Bounding

In a bounding gait, the front and hind leg pairs act in parallel. As a result, the quadruped can be modeled as a two-legged system in the sagittal plane, as shown in Figure 4. The model assumes massless legs, and thus the forces/moment

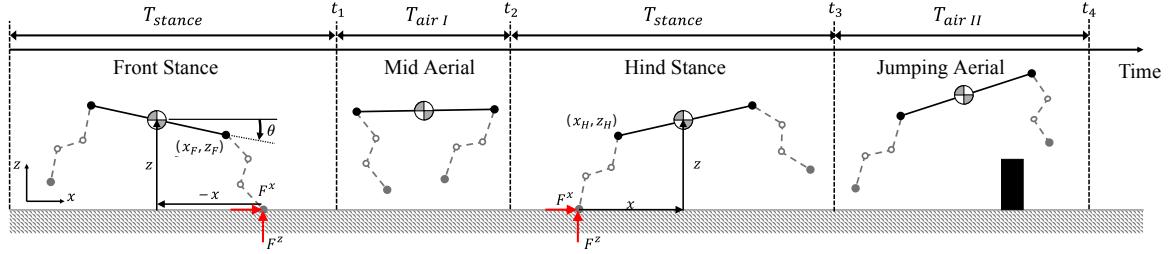


Figure 4: A simplified time-switched hybrid quadrupedal bounding model. Assuming massless legs, the quadruped is abstracted by a planar rigid body evolving under the influence of ground reaction forces and gravity. A fixed horizontal foot placement relative to the hip at the beginning of stance is used to determine the point of application for the ground reaction force.

exerted by each leg onto the body are equivalent to the horizontal and vertical ground reaction forces F^x and F^z on the foot³. In practice, these ground reaction forces can be generated by joint torques in the legs [44, 59]. The generalized coordinates of this model are $q := (x, z, \theta)$, where x is the horizontal position of the center of mass (CoM) with respect to the foot, z is the vertical position of the CoM with respect to the ground, and θ is the body pitch angle, displayed in Fig. 4.

Assuming a fixed gait timing, this simplified model is time-switched hybrid, and follows a sequential phase order of Front Stance phase, Aerial I phase, Hind Stance phase, and Jumping Aerial phase (see Figure 4). For later use, we denote t_1 , t_2 , t_3 , and t_4 as the times corresponding to the end of the Front Stance phase, the start and end of the Hind Stance phase, and the end of the Jumping Aerial phase, respectively.

During each stance phase, one pair of the legs are on the ground and the equations of motion of the robot are given by,

$$m\ddot{x} = F_x \quad (8)$$

$$m\ddot{z} = F_z - mg$$

$$I\ddot{\theta} = \gamma x F_z - \gamma z F_x$$

where, F_x and F_z are the ground reaction forces applied on the foot of the robot, m and I are the combined mass and inertia of the robot, and g is the gravitational constant. We can further simplify (8) to provide

$$\ddot{x} = u_x \quad (9)$$

$$\ddot{z} = u_z - g \quad (10)$$

$$\ddot{\theta} = \gamma x u_z - \gamma z u_x \quad (11)$$

where, $u_x = \frac{F_x}{m}$, $u_z = \frac{F_z}{m}$, and $\gamma = \frac{m}{I}$. When these scaled forces are parameterized with polynomials, the differential equations are easily integrable in a closed form. In this paper, the scaled forces u_x and u_z are chosen as n^{th} -order Bézier polynomials during the interval $t \in [t_0, t_f]$ where t_0 and t_f represent the beginning and end of the force profile.

³Note: This assumption is reasonable as the legs of the cheetah are very light, composing approximately only 10% of the total mass of the system.

A closed form solution to the differential equations proceeds as follows. Given initial conditions $[x_0, z_0, \theta_0, \dot{x}_0, \dot{z}_0, \dot{\theta}_0]$ and polynomial ground reaction forces in the interval $t \in [t_0, t_f]$, solutions to (9)-(11) can be obtained analytically. First, $x(t)$ and $z(t)$ are obtained from (9) and (10) by integrating $u_x(t), u_z(t)$ in (26) twice with respect to time. Since $u_x(t)$ and $u_z(t)$ are n^{th} -order Bézier polynomials, $x(t)$ and $z(t)$ are $(n+2)^{\text{th}}$ -order Bézier polynomials. Plugging this $x(t)$ and $z(t)$ along with $u_x(t), u_z(t)$ into (11) and integrating twice with respect to time will provide $\theta(t)$. We use product formulas for Bézier polynomials to express bi-linear terms on the right-hand side of (11). A detailed procedure is explained in the Appendix.

During the two aerial phases, Mid Aerial phase and Jumping Aerial phase, the ground reaction forces u_x and u_z become zero, and the robot follows ballistic dynamics of $\ddot{x} = \ddot{\theta} = 0$ and $\ddot{z} = -g$. A straightforward closed-form integration procedure applies for these equations as well.

These integration procedures, in sum, provide an analytical method to evaluate the body state at any given time during a candidate jump. However, successful jumping is as much about foot clearance as it is body propulsion. To simplify this consideration, the swing foot position is chosen to follow a fixed trajectory with respect to the shoulder during flight. The trajectory is shown in Figure 5. After these swing trajectories are completed, the position of the foot with respect to the shoulder is held to wait for the impact with the ground. These foot swing trajectories are designed considering the workspace of the leg through a separate one-time offline optimization procedure. Combination of these trajectories with the body state thus allows the position of the foot to be queried at any instant of a candidate jump.

Concerning the jump timing, the duration of phases T_{stance} and $T_{\text{air I}}$ are fixed at values required during normal bounding, as given in [43], while the duration of the last phase $T_{\text{air II}}$ is elongated as the robot jumps up to clear the obstacle. The value of $T_{\text{air II}}$ is left free for optimization so that the jump hang time is tailored to the sensed obstacle.

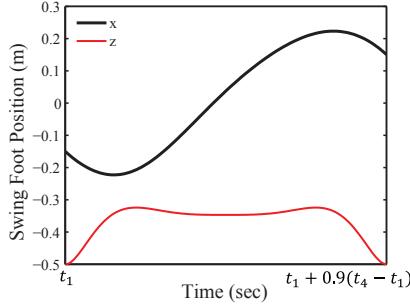


Figure 5: Swing foot trajectory for the front leg with respect to the shoulder during its aerial period of motion. The trajectory is designed to complete within 90% of the expected flight time to recover from unexpected early impact. The back leg trajectory follows a similar pattern.

4.2. Jumping Trajectory Generation via Constrained Nonlinear Programming

This section explains the new trajectory generation formulation for jumping over obstacles. A feasible jumping trajectory to clear the obstacle is obtained by solving a nonlinear programming problem. Optimization variables for this problem are:

$$\xi = [\alpha_x^f, \alpha_z^f, \alpha_x^h, \alpha_z^h, T_{air,II}]^T$$

where α_x^f and α_z^f are scaling coefficients for horizontal (x) and vertical (z) ground reaction force profiles for the front (f) legs, α_x^h and α_z^h likewise for the hind legs (h), and $T_{air,II}$ the hang time of the jump.

The ground reaction force inputs $u_x(t)$ and $u_z(t)$ are designed using Bézier polynomials to take a fixed shape both in the x and z directions and in the front and hind legs. The common shape is constructed so that it provides a single peak trajectory where the peak is located in the middle of the stance phase while the profile starts and ends at zero force. Figure 6 shows example force profiles when peak values are $\alpha_x^f = -180$ N, $\alpha_x^h = -178$ N, $\alpha_z^f = 756$ N, and $\alpha_z^h = 1000$ N. A detailed construction of this polynomial is given in [46], however, the qualitative shape is the most important takeaway.

To select values for the optimization parameters ξ , a constrained nonlinear feasibility problem was posed. Since the success of the clearance was the dominant goal in this work, an objective function of 0 was employed. This selection also served to accelerate the optimization in comparison to including a more complex objective function. Constraints over the variables were selected as described in Table 2.

In this table h is the height of the obstacle and d the distance to it at the start of the jump. The parameter w controls the fore-aft obstacle clearance width and was chosen as 15 cm in experiments. The desired clearance height $h_{obs}(x)$ around the obstacle is given as a function of forward position by multiplying two sigmoid functions:

$$h_{obs}(x) = h_0 \frac{1}{1 + e^{-\sigma(x-\underline{d})}} \frac{1}{1 + e^{\sigma(x-\bar{d})}} \quad (24)$$

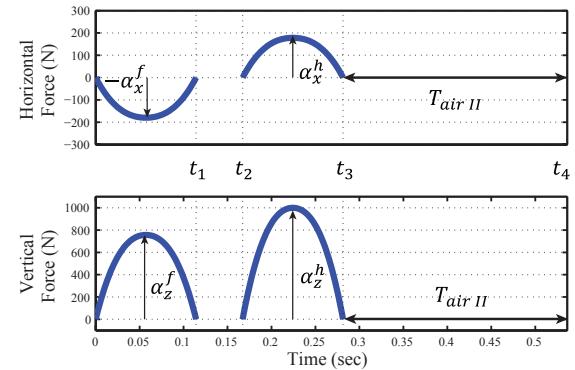


Figure 6: Example force profile for the front (f) and hind (h) legs optimized to clear an obstacle 33 cm high at a distance of 1 m.

Table 2

Constraints imposed for obstacle clearance feasibility.

$ \theta(t) < \Theta,$	for $0 \leq t \leq t_4$	(12)
$z_{foot}^f(t) > h_{obs}(x_{foot}^f(t)),$	for $t_1 \leq t \leq t_4$	(13)
$z_{foot}^h(t) > h_{obs}(x_{foot}^h(t)),$	for $t_3 \leq t \leq t_4$	(14)
$x_{foot}^f(t_4) > d + 0.5w$		(15)
$x_{foot}^h(t_4) > d + 0.5w$		(16)
$\underline{z}^f < z^f(t_1) < \bar{z}^f$		(17)
$\underline{z}^h < z^h(t_2) < \bar{z}^h$		(18)
$\underline{z}^h < z^h(t_3) < \bar{z}^h$		(19)
$z_{foot}^f(t_4) = \bar{z}^f$		(20)
$\underline{\theta} < \theta(t_4) < \bar{\theta}$		(21)
$ \alpha_x^j < \mu \alpha_z^j$	for $j \in \{f, h\}$	(22)
$0 < \alpha_z^j < \bar{F}_z$	for $j \in \{f, h\}$	(23)

with $\sigma = 200$ and $\underline{d}, \bar{d} = d \pm .5w$ respectively. As the coefficient σ is increased, the clearance constraints better approximate non-collision with a rectangular box. However this increase also causes the derivative of the constraint function to change more rapidly, and this behavior can require the optimizer to take smaller steps. The parameter choices above were found to balance conservatism with conditioning considerations for the optimizer.

The constraints in Table 2 are motivated as follows. The constraint (12) is posed to avoid excessive torso oscillation during the jumping, with the constant $\Theta > 0$ chosen to be 45°. The constraints (13) and (14) are posed to avoid a foot tripping over the obstacle, (15) and (16) ensure that the robot's foot has achieved the desired lateral clearance at the completion of the jump. The constraints (17) and (19) are used to guarantee that the robot's shoulder and hip height above the ground (z^f and z^h , respectively) at the end of each stance phase is within the range of the leg's workspace. Likewise, the constraint (18) ensures that the Hind Stance phase starts with the hip height within the range of the hind leg's workspace.

The constraints (20) and (21) are posed to provide a configuration for a safe landing. Parameters $\underline{\theta}$ and $\bar{\theta}$ are lower and upper bounds of pitch angle at the landing, and \tilde{z}^f is the fixed vertical foot position (zero in our case) at the end of the jumping. **Constraints (22) and (23) require forces to respect friction and force limits, respectively.**

The above nonlinear feasibility problem was solved using the nonlinear solver SNOPT [22]. Continuous constraints (12)-(14) were discretized in time with 50-100 time steps per gait phase. Computation of the parameters for a feasible jumping trajectory takes less than 100 ms on average using an on-board computer with an Intel Core i5-4520U CPU. Figure 6 shows an example force profile with parameters obtained solving this nonlinear programming problem for an obstacle with the distance of 1 m and height of 0.33 m.

In implementation for the experimental hardware, trajectory optimization was carried out on any step when the approach modification provided $N^* \leq 2$. The optimization was carried out when $N^* = 2$ as a precaution in case the following step placed the quadruped too close to the obstacle and an emergency jump was required. In all cases, a predicted distance to the obstacle at the next step $d = d_0 - v_0 T^* - \frac{1}{2} \Delta v_0^*$ was passed to the trajectory optimization using the results of the approach modification.

4.3. Landing Control

The purpose of the landing control is to provide a safe and robust transition back to the normal running gait by handling disturbances specific to the high impacts on the front legs at landing. We modify the horizontal and vertical force profiles of the Front Stance phase to provide additional impulse to recover the robot's velocity states to their nominal values. Because we use Bézier polynomials in (26) for force profiles $u_x(t)$ and $u_z(t)$, the impulse $\int_0^{T_{stance}} u_x(t) dt$ and $\int_0^{T_{stance}} u_z(t) dt$ can be calculated by simply averaging Bézier coefficients and multiplying the length of duration. Therefore, by uniformly scaling Bézier coefficients accordingly, we can modify the impulse created by $u_x(t)$ and $u_z(t)$.

The vertical landing speed of the CoM $\dot{z}(t_4)$ at the moment of the landing is obtained from the jumping trajectory optimization explained in Section 4.2. The required vertical impulse to steer $\dot{z}(t_4)$ to the nominal value of \dot{z}_1 can be calculated using linear impulse and momentum change relationship which is given by,

$$\int_0^{T_{stance}} u_z dt = \dot{z}_1 - \dot{z}(t_4) + g T_{stance} \quad (25)$$

On the other hand, the horizontal force is modulated to modify rotational impulse. Unlike the normal running case where the swing foot retracts fast enough to provide ground speed matching, the swing foot speed with respect to the ground will be large after the jump because the foot position with respect to the shoulder position is held in position after the completion of the desired swing foot trajectories. **Ground speed matching was not included during the jump since it is much more difficult to predict the touchdown in-**

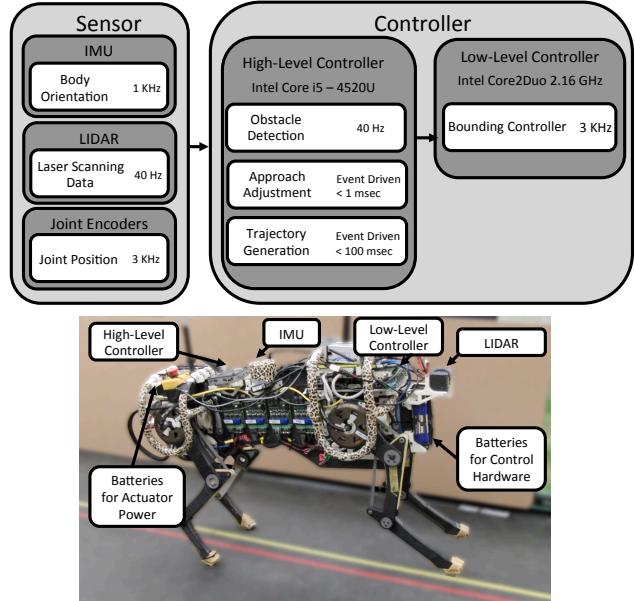


Figure 7: MIT Cheetah 2 system setup: sensors and computational architecture.

stance during jumping than it is during running, and inaccurate prediction rapidly degrades foot positioning. Due to the large relative swing foot speed at impact, excessive rotational momentum about the CoM will be generated upon impact with the ground. To handle this disturbance, the rotational velocity is sensed following impact, and the horizontal force profile is heuristically adjusted to cancel out additional rotational momentum.

5. Results

This section discusses results with Cheetah 2 jumping over obstacles. Section 5.1 first describes the experimental setup. Sections 5.2 and 5.3 then detail results on a treadmill in the lab and on a flat running surface at a gym. It is noted that the results in the laboratory were fully untethered, in the sense that all power was supplied on board and all computations took place with on-board hardware.

5.1. Experimental Setup

Figure 7 details the sensors and control computers used by the Cheetah to implement the algorithms described in the previous sections. All high-level control for obstacle detection, approach planning, and trajectory generation was performed using a single core of a 2.6 GHz Intel Core i5-54250U processor on a Nuc Mini-PC. This PC also completed a terrain scan 40 times per second using a Hokuyo UTM-30LX-EW Lidar. Lower-level bounding control was carried out on a 2.16 GHz Intel Core2Duo SpeedGoat board running compiled Simulink. The base control frequency for the lower-level bounding controller was 3 KHz with further details provided in [44].

In each of the experiments, the quadruped was brought to a steady state bounding gait of approximately 2.4 m/s. During normal running (i.e., no obstacles), a wireless link to a

Jumping Over Obstacles with MIT Cheetah 2

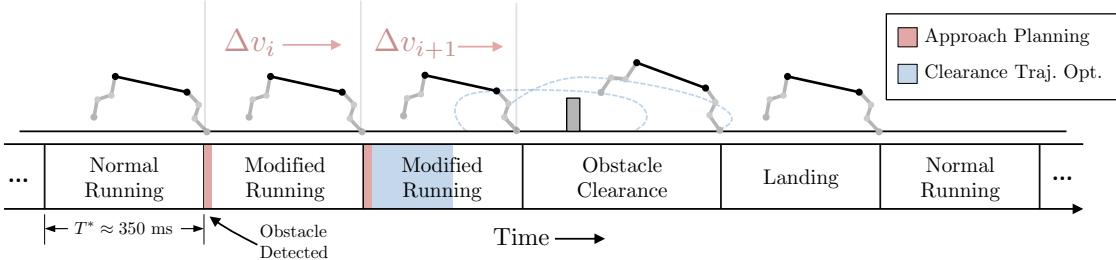


Figure 8: Procedure overview showing computation breakdown with time. Upon detecting an obstacle, the system plans a modified approach using a multiple-horizon model predictive control strategy (Sec. 3). During the step immediately before the obstacle, the system performs online trajectory optimization (Sec. 4.2) in order to select ground force profiles that will allow the quadruped to clear the obstacle.

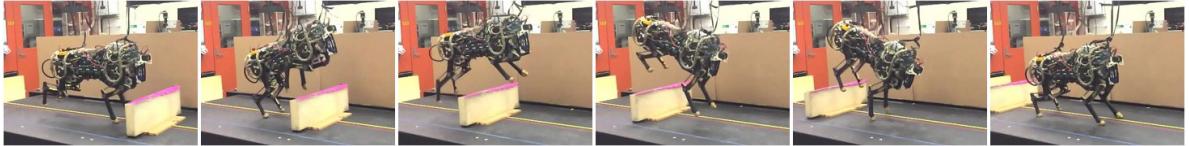


Figure 9: Motion snapshots from a successful jump over a 27cm obstacle. Frames are separated by 150 ms.

host computer was used to provide a desired running speed as well as a heading set point from an operator. This manual heading control served two purposes. First, in the laboratory, heading control was used to keep the robot on the treadmill. Secondly, beyond the lab, manual control was used to steer the robot toward the obstacles. Besides this manual heading input, all other control actions were autonomous. No information was given to the robot regarding the fore-aft placement of the obstacle, placement timing, or obstacle height.

5.2. Lab Experiments

Videos of the laboratory experiments are shown in the supplemental multimedia material that accompanies this paper. The protocol for experiments was as follows. After the Cheetah reached its steady-state speed, an operator would place an obstacle on the treadmill in front of the robot at a random time. Again, the Cheetah was given no information regarding the timing or height of the obstacle. A range of obstacles were tested from 27cm to 40cm in height.

After the obstacle was detected, the approach planning and jump optimization algorithms would take place in the fraction of a second before needing to execute the jump. Figure 8 gives an overview of the time scales over which the computations took place. Following detection, approach optimization was run each time the front feet touched down, roughly 3 times per second. Due to the limited length of the treadmill, the Cheetah often could only take two steps before needing to jump after the obstacle was placed. Thus, all computations needed to take place roughly within 650ms.

Figure 9 shows representative snapshots of jumping over a 27cm obstacle. To clear the obstacle, the Cheetah would thrust first with its front legs to pitch the body over the obstacle, and then would follow this with a propulsive thrust from the back legs to propel the system forward. Figure 10 shows

synchronized body pitch and ground force data for a similar jump over a 34cm obstacle. While the nominal pitch variation during steady state bounding was 0.25 rad, the pitch excursion during the jump exceeded 1 rad.

The force profiles needed to execute the jump were likewise substantially different from steady-state locomotion. The peak forces for the front and back legs were nominally 300N in the vertical direction during steady-state bounding. In comparison, the peak-forces for jumping were 460N and 525N for the front and back legs respectively. Further, note that during nominal running, the front legs have a force vector that is pointed toward the rear on average, while the back legs have a force that is pointed toward the front. In comparison, during the first front stance at touchdown following the jump, the fore/aft force has a significant positive component in order to assist in reversing the pitch rate at touchdown.

Still, due to the extreme nature of these behaviors, the open-loop jumping motion often requires extreme corrective actions upon landing. Although the pitch returned to its nominal range within the first step following the jump in Fig. 10, an irregular double stance period was observed. The ability of the Cheetah to recover from this situation is attributed to gait pattern stabilization advances that we have presented in a previous publication [47].

Figure 11 shows a broader set of algorithm outputs for jumping over a 40cm obstacle. The first subplot shows the output of the obstacle detection algorithm, which ran at a rate of 40Hz. The obstacle was detected when it was 2.5m away from the center of the body. Following initial detection, the mean sensed height was 40.42cm with a standard deviation of 2.1cm. The second plot shows modifications to the desired velocity as computed by the approach optimization. The stride velocity shown on this graph represents the average velocity of the hip relative to the foot during a given

Jumping Over Obstacles with MIT Cheetah 2

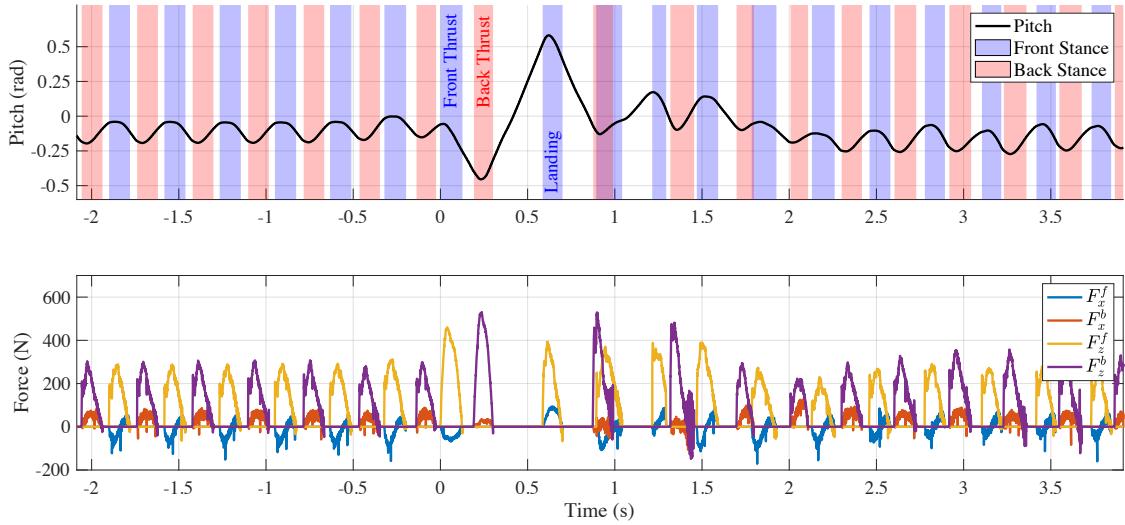


Figure 10: System outputs when jumping over a 34cm obstacle in the laboratory.

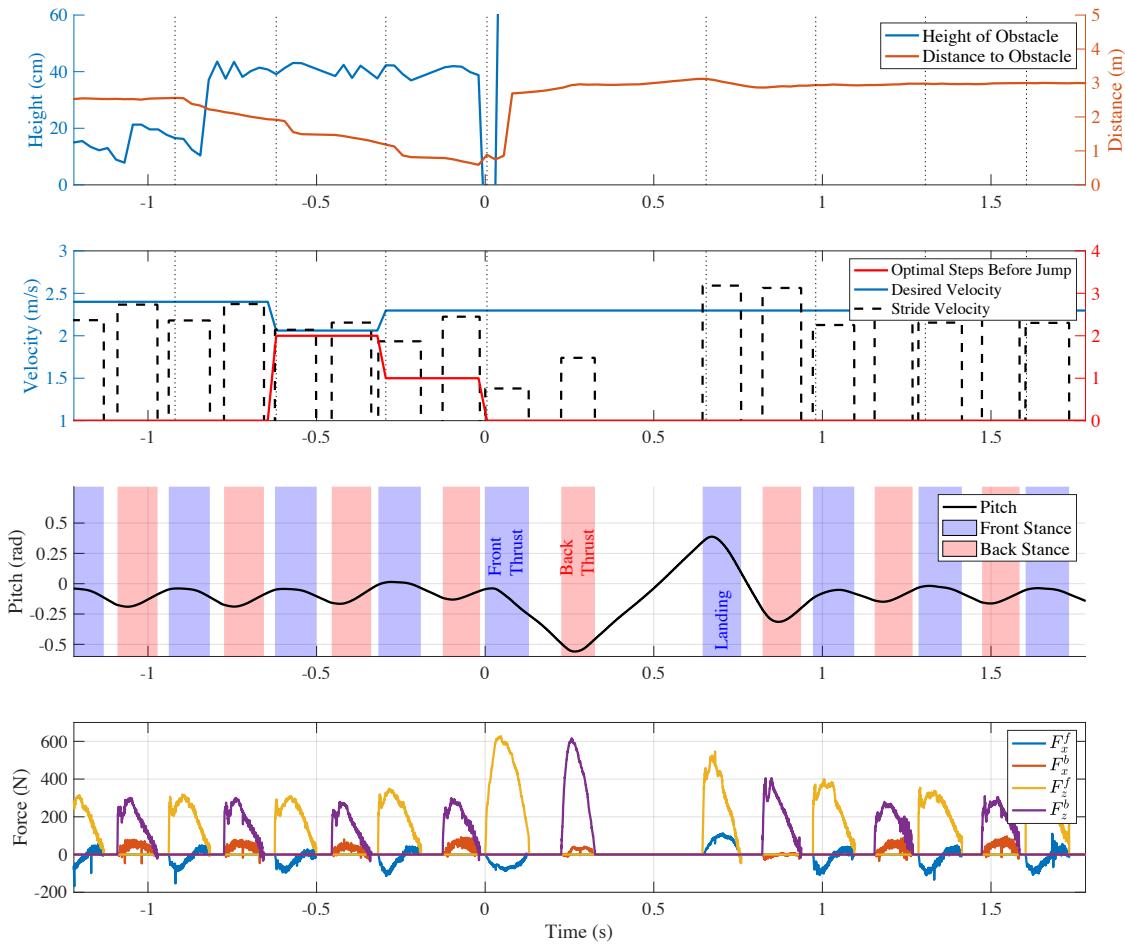


Figure 11: Key system outputs for jumping over a 40cm obstacle in the laboratory.

stance phase. It is observed that in the step immediately before jumping (i.e., around $t = -0.25s$) there is a half-stride delay in realizing the desired speed change. This delay is due, in part, to the fact that a jump trajectory was optimized during this step. The desired speed was not relayed to the

bounding controller until the completion of the jump optimization. The bottom plots show the pitch and force similar to the previous analysis for the medium obstacle. For this larger obstacle, the force profiles used had peaks 625N and 615N in order to clear the jump.

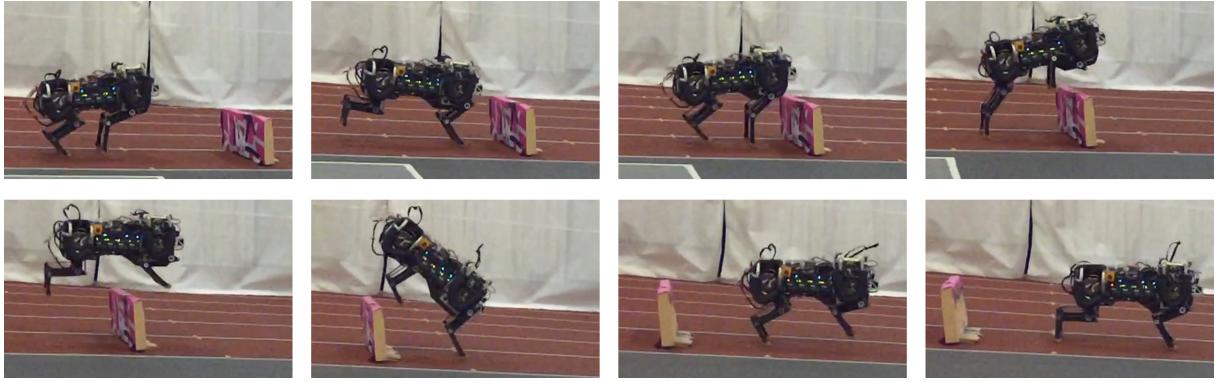


Figure 12: Snapshots from jumping over a 40cm obstacle without a safety harness.

The video attachment for this paper shows examples of other experiments in the laboratory. In one case, the Cheetah glances the obstacle during the jump. This example highlights limiting features of our framework, namely, that the Lidar scan currently only takes place in the sagittal plane and the obstacle is assumed to be placed perpendicular to this line of sight. [The video also shows experiments where the 27cm an 34cm obstacle are cleared multiple times in a row during treadmill experiments, showcasing repeatability of the tests.](#)

5.3. Fully Untethered Experiments

Following initial debugging of the algorithm, the robot was tested without its safety harness. Two main experiments are showcased in the video attachment. A first sequence shows the cheetah jumping over a 34cm, 40cm, and 27cm obstacle in order. A second sequence shows the progression 27cm, 34cm, and 40cm. These experiments highlight the ability of the Cheetah to tailor its jumps to the situation at hand. A montage from a jump over a 40cm obstacle is shown in Fig. 12.

In the fully untethered experiments, three obstacles were set up in random locations along a roughly 60m straight-away of an indoor track. The running surface provided favorable terrain conditions, in that the ground was both flat and included a rubberized surface texture. The real-time Simulink Speedgoat board recorded data at 3KHz, and thus, limited available memory prevented us from collecting log files for the full length of our experiments. Instead, only the data from the high-level planning and optimization on a Nuk PC is reported.

Figure 13 shows the planning data for the first sequence of three jumps (obstacles 34cm, 40cm, and 27 cm in height). Again, the top subfigure shows the data from the obstacle detection algorithm, with the second subfigure showing the output of the approach MPC. This data shows the diversity of strategies taken by the Cheetah to get into position for the jump; sometimes the robot speeds up during the approach, while other times it slows down. Further, due to the discounting strategy in the formulation (5), the system makes larger changes in the desired velocity at steps further from the goal.

A second sequence of jumps is also shown in the video attachment. Unfortunately, at the end of this sequence, the Cheetah unintentionally attempted to jump over a large bench at the end of the track segment, which led to an overvoltage during the crash and a loss of system data. This failure highlights a general challenge not addressed in the current work. Namely, that in the absence of directly sensing the depth of objects, context clues play a large role in our perception of the environment. The inference of geometric features more generally represents an interesting area of future work for combination with model-based planning.

6. Summary and Conclusions

This paper has developed new algorithms to dynamically jump over obstacles up to 40cm in height (80% of leg length). The quadruped reasons about its dynamics over multiple time scales to select appropriate actions to prepare for and execute the running jump. A model-predictive approach controller places the system in an optimal location relative to the obstacle to perform the running jump. By focusing on essential characteristics of the approach, this algorithm can be computed in 250 μ s, and also determines the optimal number of steps to be taken before the obstacle. In the final steps of the approach, a more detailed trajectory optimization problem is solved that determines ground force profiles to propel the system up and over the obstacle while maintaining clearance for its swing legs. [These algorithms have shown the ability to jump over obstacles from 27cm to 40cm in height while running at roughly 2.4m/s. Results with multiple jumps in a row show the ability of the robot to rapidly coordinate adjustments to its speed profiles and ground forces toward tailoring its behavior to the sensed environment.](#)

The field of legged locomotion has experienced many recent technological and algorithmic advances that inform how this work will fit into the next steps of future research. The wide commercial availability of 3D visual depth sensing options has broadened the ability for legged systems to perceive environment details well beyond the limited 2D information captured in this work. This information opens the door for making discrete decisions such as whether to run around an obstacle or to jump over it. To address these

Jumping Over Obstacles with MIT Cheetah 2

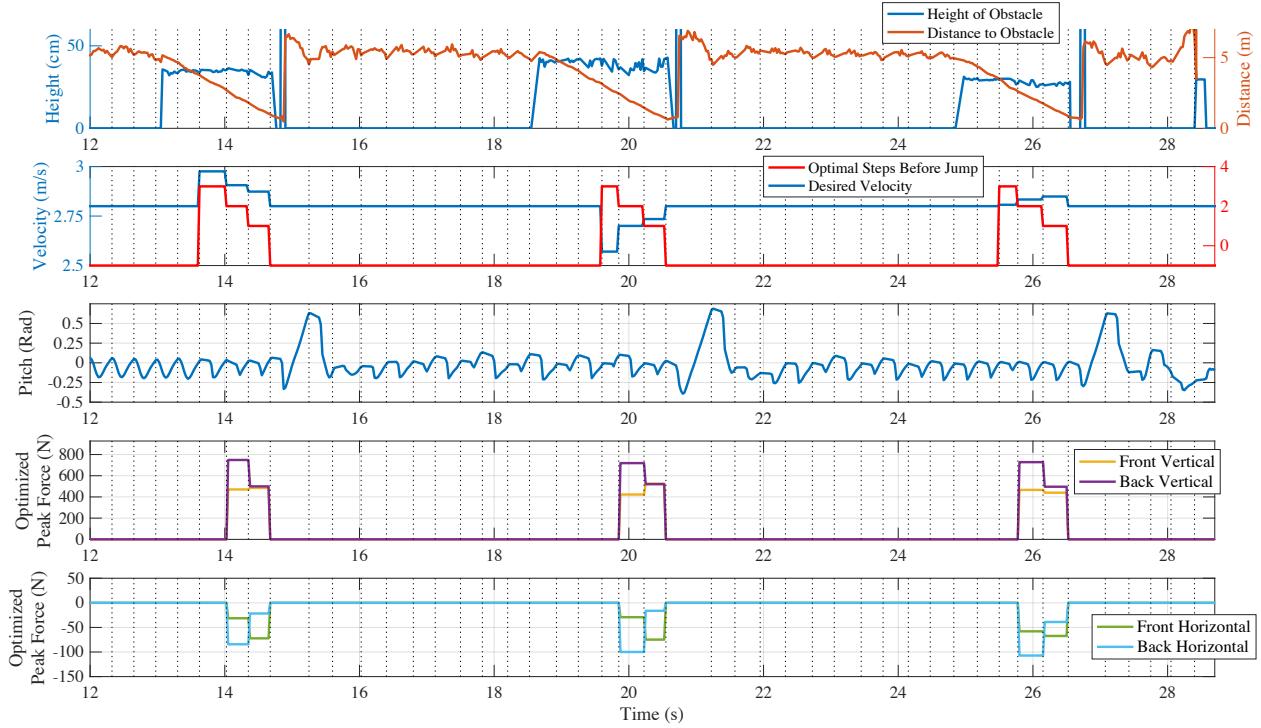


Figure 13: Sequence of three jumps in a row during experiments at the gym when operating without a safety harness.

options, the field is in need of new approaches to planning that handle discrete decisions in a computationally tractable manner, as existing approaches based on mixed integer planning [15, 55, 2] or contact implicit optimization [9, 48, 36] present computational timing and reliability challenges for use in rapid replanning. In this sense, the authors note that while great progress has been made (and continues to be made) in whole-body model-predictive control strategies for continuous planning [31, 38, 23, 33], it is likely that strategic modeling simplifications, such as those pursued here, will foreseeably continue to play a critical role in discrete high-level planning. Strategies that inject machine learning with model-based methods (c.f., [8, 37, 35, 56]) to address these computational issues are an attractive option that has shown early promise, yet warrant much further additional consideration by the community. These future advances will be critical if legged systems are to fulfill their intended roles of operating in truly novel environments as they explore, map, and operate in unstructured worlds.

References

- [1] Aceituno-Cabezas, B., Mastalli, C., Dai, H., Focchi, M., Radulescu, A., Caldwell, D.G., Cappelletto, J., Grieco, J.C., Fernandez-Lopez, G., Semini, C., 2017a. Simultaneous Contact, Gait and Motion Planning for Robust Multi-Legged Locomotion via Mixed-Integer Convex Optimization. *IEEE Robotics and Automation Letters (RAL)* doi:10.1109/LRA.2017.2779821.
- [2] Aceituno-Cabezas, B., Mastalli, C., Dai, H., Focchi, M., Radulescu, A., Caldwell, D.G., Cappelletto, J., Grieco, J.C., Fernández-López, G., Semini, C., 2017b. Simultaneous contact, gait, and motion planning for robust multilegged locomotion via mixed-integer convex optimization. *IEEE Robotics and Automation Letters* 3, 2531–2538.
- [3] Alessio, A., Bemporad, A., 2009. A survey on explicit model predictive control, in: Nonlinear model predictive control. Springer, pp. 345–369.
- [4] Bazeille, S., Barasuol, V., Focchi, M., Havoutis, I., Frigerio, M., Buchli, J., Caldwell, D.G., Semini, C., 2014. Quadruped robot trotting over irregular terrain assisted by stereo-vision. *Intelligent Service Robotics* 7, 67–77.
- [5] Bledt, G., Powell, M.J., Katz, B., Di Carlo, J., Wensing, P.M., Kim, S., 2018a. MIT Cheetah 3: Design and control of a robust, dynamic quadruped robot, in: Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, Madrid, Spain.
- [6] Bledt, G., Wensing, P.M., Ingersoll, S., Kim, S., 2018b. Contact model fusion for event-based locomotion in unstructured terrains, in: Proceedings of the 2018 IEEE International Conference on Robotics and Automation, Brisbane, Australia.
- [7] Bledt, G., Wensing, P.M., Kim, S., 2017. A predictive control framework to stabilize diverse gaits for the MIT Cheetah, in: Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, Canada. pp. 4102–4109. doi:10.1109/IROS.2017.8206268.
- [8] Carius, J., Farshidian, F., Hutter, M., 2020. Mpc-net: A first principles guided policy search. *IEEE Robotics and Automation Letters* 5, 2897–2904.
- [9] Carius, J., Ranftl, R., Koltun, V., Hutter, M., 2018. Trajectory optimization with implicit hard contacts. *IEEE Robotics and Automation Letters* 3, 3316–3323.
- [10] Coros, S., Karpathy, A., Jones, B., Reveret, L., van de Panne, M., 2011. Locomotion Skills for Simulated Quadrupeds, in: ACM SIGGRAPH 2011, ACM, New York, NY, USA. pp. 59:1–12.
- [11] Da, X., Grizzle, J., 2019. Combining trajectory optimization, supervised machine learning, and model structure for mitigating the curse of dimensionality in the control of bipedal robots. *The International Journal of Robotics Research* 38, 1063–1097.
- [12] Dai, H., Tedrake, R., 2013. L2-gain optimization for robust bipedal walking on unknown terrain, in: IEEE International Conference on Robotics and Automation (ICRA), pp. 3116–3123. doi:10.1109/ICRA.2013.6631010.

- [13] Dellin, C., Srinivasa, S., 2012. A framework for extreme locomotion planning, in: IEEE Int. Conf. on Robotics and Automation, pp. 989–996.
- [14] Di Carlo, J., Wensing, P.M., Katz, B., Bledt, G., Kim, S., 2018. Dynamic locomotion in the MIT Cheetah 3 through convex model-predictive control, in: Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, Madrid, Spain.
- [15] Ding, Y., Li, C., 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.
- [16] Doha, E., Bhrawy, A., Saker, M., 2011. Integrals of bernstein polynomials: An application for the solution of high even-order differential equations. *Applied Mathematics Letters* 24, 559 – 565.
- [17] Fankhauser, P., Bellicoso, C.D., Gehring, C., DubÃl, R., Gawel, A., Hutter, M., 2016. Free gait - an architecture for the versatile control of legged robots, in: 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), pp. 1052–1058. doi:10.1109/HUMANOIDS.2016.7803401.
- [18] Ferreau, H., Bock, H., Diehl, M., 2008. An online active set strategy to overcome the limitations of explicit MPC. *International Journal of Robust and Nonlinear Control* 18, 816–830.
- [19] Ferreau, H., Kirches, C., Potschka, A., Bock, H., Diehl, M., 2014. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation* 6, 327–363.
- [20] Focchi, M., Barasolu, V., Havoutis, I., Buchli, J., Semini, C., Caldwell, D.G., 2013. Local Reflex Generation for Obstacle Negotiation in Quadrupedal Locomotion, in: Proc. of CLAWAR.
- [21] Focchi, M., Orsolino, R., Camurri, M., Barasolu, V., Mastalli, C., Caldwell, D.G., Semini, C., 2018. Heuristic Planning for Rough Terrain Locomotion in Presence of External Disturbances and Variable Perception Quality. Springer Tracts in Advanced Robotics (STAR), Springer.
- [22] Gill, P., Murray, W., Saunders, M., 2005. SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization. *SIAM Review* 47, 99–131.
- [23] Grandia, R., Farshidian, F., Ranftl, R., Hutter, M., 2019. Feedback mpc for torque-controlled legged robots, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 4730–4737.
- [24] Heess, N., TB, D., Sriram, S., Lemmon, J., Merel, J., Wayne, G., Tassa, Y., Erez, T., Wang, Z., Eslami, S.M.A., Riedmiller, M., Silver, D., 2017. Emergence of Locomotion Behaviours in Rich Environments. arXiv e-prints , arXiv:1707.02286arXiv:1707.02286.
- [25] Hutter, M., Gehring, C., Jud, D., Lauber, A., Bellicoso, C.D., Tsounis, V., Hwangbo, J., Bodie, K., Fankhauser, P., Bloesch, M., et al., 2016. Anymal-a highly mobile and dynamic quadrupedal robot, in: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IEEE, pp. 38–44.
- [26] Hwangbo, J., Lee, J., Dosovitskiy, A., Bellicoso, D., Tsounis, V., Koltun, V., Hutter, M., 2019. Learning agile and dynamic motor skills for legged robots. *Science Robotics* 4.
- [27] Jenelten, F., Miki, T., Vijayan, A.E., Bjelonic, M., Hutter, M., 2020. Perceptive locomotion in rough terrain—online foothold optimization. *IEEE Robotics and Automation Letters* 5, 5370–5376.
- [28] Kalakrishnan, M., Buchli, J., Pastor, P., Mistry, M., Schaal, S., 2011. Learning, planning, and control for quadruped locomotion over challenging terrain. *The International Journal of Robotics Research* 30, 236–258.
- [29] Katz, B., Di Carlo, J., Kim, S., 2019. Mini cheetah: A platform for pushing the limits of dynamic quadrupedal control, in: International Conference on Robotics and Automation (ICRA), pp. 6295–6301.
- [30] Kim, D., Di Carlo, J., Katz, B., Bledt, G., Kim, S., 2019. Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control. arXiv preprint arXiv:1909.06586 .
- [31] Koenemann, J., Del Prete, A., Tassa, Y., Todorov, E., Stasse, O., Bennewitz, M., Mansard, N., 2015. Whole-body model-predictive control applied to the hrp-2 humanoid, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 3346–3351.
- [32] Krasny, D., Orin, D., 2006. Evolution of dynamic maneuvers in a 3D galloping quadruped robot, in: IEEE Int. Conf. on Robotics and Automation, pp. 1084–1089.
- [33] Li, H., Frei, R.J., Wensing, P.M., 2020. Model hierarchy predictive control of robotic systems. arXiv preprint arXiv:2010.08881 .
- [34] Liu, Y., Wensing, P.M., Schmideler, J.P., Orin, D.E., 2016. Terrain-blind humanoid walking based on a 3D actuated dual-SLIP model. *IEEE Robotics and Automation Letters* 1, 1073–1080. doi:10.1109/LRA.2016.2530160.
- [35] Magana, O.A.V., Barasolu, V., Camurri, M., Franceschi, L., Focchi, M., Pontil, M., Caldwell, D.G., Semini, C., 2019. Fast and continuous foothold adaptation for dynamic locomotion through cnns. *IEEE Robotics and Automation Letters* 4, 2140–2147.
- [36] Manchester, Z., Kuindersma, S., 2020. Variational contact-implicit trajectory optimization, in: Robotics Research. Springer, pp. 985–1000.
- [37] Mansard, N., DelPrete, A., Geisert, M., Tonneau, S., Stasse, O., 2018. Using a memory of motion to efficiently warm-start a non-linear predictive controller, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 2986–2993.
- [38] Neunert, M., De Crousaz, C., Furrer, F., Kamel, M., Farshidian, F., Siegwart, R., Buchli, J., 2016. Fast nonlinear model predictive control for unified trajectory optimization and tracking, in: 2016 IEEE international conference on robotics and automation (ICRA), IEEE, pp. 1398–1404.
- [39] Neunert, M., Farshidian, F., Winkler, A.W., Buchli, J., 2017. Trajectory optimization through contacts and automatic gait discovery for quadrupeds. *IEEE Robotics and Automation Letters* 2, 1502–1509. doi:10.1109/LRA.2017.2665685.
- [40] Neunert, M., StÃduble, M., Gifthaler, M., Bellicoso, C.D., Carius, J., Gehring, C., Hutter, M., Buchli, J., 2018. Whole-body nonlinear model predictive control through contacts for quadrupeds. *IEEE Robotics and Automation Letters* 3, 1458–1465. doi:10.1109/LRA.2018.2800124.
- [41] Nguyen, Q., Agrawal, A., Martin, W., Geyer, H., Sreenath, K., 2018. Dynamic bipedal locomotion over stochastic discrete terrain. *The International Journal of Robotics Research* 37, 1537–1553.
- [42] Park, H.W., 2012. Control of a Bipedal Robot Walker on Rough Terrain. Ph.D. thesis, University of Michigan.
- [43] Park, H.W., Chuah, M.Y., Kim, S., 2014. Quadruped bounding control with variable duty cycle via vertical impulse scaling, in: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 3245–3252.
- [44] Park, H.W., Park, S., Kim, S., 2015a. Variable-speed quadrupedal bounding using impulse planning: Untethered high-speed 3D running of MIT Cheetah 2, in: IEEE Int. Conf. on Robotics and Automation.
- [45] Park, H.W., Ramezani, A., Grizzle, J., 2013. A finite-state machine for accommodating unexpected large ground-height variations in bipedal robot walking. *IEEE Trans. Robot.* 29, 331–345. doi:10.1109/TRO.2012.2230992.
- [46] Park, H.W., Wensing, P., Kim, S., 2015b. Online planning for autonomous running jumps over obstacles in high-speed quadrupeds, in: Proceedings of Robotics: Science and Systems, Rome, Italy. doi:10.15607/RSS.2015.XI.047.
- [47] Park, H.W., Wensing, P.M., Kim, S., 2017. High-speed bounding with the mit cheetah 2: Control design and experiments. *The International Journal of Robotics Research* 36, 167–192. URL: <https://doi.org/10.1177/0278364917694244>, doi:10.1177/0278364917694244, arXiv:<https://doi.org/10.1177/0278364917694244>.
- [48] Patel, A., Shield, S.L., Kazi, S., Johnson, A.M., Biegler, L.T., 2019. Contact-implicit trajectory optimization using orthogonal collocation. *IEEE Robotics and Automation Letters* 4, 2242–2249.
- [49] Pavlidis, T., Horowitz, S., 1974. Segmentation of Plane Curves. *IEEE Transactions on Computers* C-23, 860–870.
- [50] Raibert, M., Blakespoor, K., Nelson, G., Playter, R., the Big-Dog Team, 2008. BigDog, the Rough-Terrain Quaduped Robot, in:

- Proceedings of the 17th World Congress, pp. 10822–10825.
- [51] Ramer, U., 1972. An iterative procedure for the polygonal approximation of plane curves. Computer Graphics and Image Processing 1, 244 – 256.
- [52] Semini, C., Tsagarakis, N.G., Guglielmino, E., Focchi, M., Cannella, F., Caldwell, D.G., 2011. Design of HyQ – a hydraulically and electrically actuated quadruped robot. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 225, 831–849.
- [53] Seok, S., Wang, A., Chuah, M.Y., Otten, D., Lang, J., Kim, S., 2013. Design principles for highly efficient quadrupeds and implementation on the MIT Cheetah robot, in: IEEE Int. Conf. on Robotics and Automation, pp. 3307–3312.
- [54] Tan, J., Zhang, T., Coumans, E., Iscen, A., Bai, Y., Hafner, D., Bohez, S., Vanhoucke, V., 2018. Sim-to-Real: Learning Agile Locomotion For Quadruped Robots. arXiv e-prints , arXiv:1804.10332arXiv:1804.10332.
- [55] Tonneau, S., Song, D., Fernbach, P., Mansard, N., Taix, M., Del Prete, A., 2020. Sl1m: Sparse l1-norm minimization for contact planning on uneven terrain, in: 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE. pp. 6604–6610.
- [56] Villarreal, O., Barasuol, V., Wensing, P.M., Caldwell, D.G., Semini, C., 2020. Mpc-based controller with terrain insight for dynamic legged locomotion, in: 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE. pp. 2436–2442.
- [57] Wang, Y., Boyd, S., 2010. Fast Model Predictive Control Using Online Optimization. IEEE Transactions on Control Systems Technology 18, 267–278.
- [58] Wensing, P., Orin, D., 2014. Development of high-span running long jumps for humanoids, in: IEEE Int. Conf. on Robotics and Automation, pp. 222–227.
- [59] Wensing, P.M., Wang, A., Seok, S., Otten, D., Lang, J., Kim, S., 2017. Proprioceptive actuator design in the mit cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots. Ieee transactions on robotics 33, 509–522.
- [60] Winkler, A.W., Bellicoso, C.D., Hutter, M., Buchli, J., 2018. Gait and trajectory optimization for legged systems through phase-based end-effector parameterization. IEEE Robotics and Automation Letters 3, 1560–1567. doi:10.1109/LRA.2018.2798285.
- [61] Wong, H.C., Orin, D.E., 1995. Control of a quadruped standing jump over irregular terrain obstacles. Autonomous Robots 1, 111–129.

A. Bezier Trajectory Details

As shown in (9) and (10), $x(s)$ and $z(s)$ can be obtained by integrating $u_x(s)$ and $u_z(s)$ twice. Since $u_x(s)$ and $u_z(s)$ are n^{th} -order Bézier polynomials, they are given by,

$$u_x(s) = \sum_{i=0}^n \beta_{i,x} b_{i,n}(s) \quad (26)$$

$$u_z(s) = \sum_{i=0}^n \beta_{i,z} b_{i,n}(s),$$

where $\beta_{i,x}$ and $\beta_{i,z}$ are the Bézier coefficients as optimized in Section 4.2, $s := \frac{t-t_0}{t_f-t_0} \in [0, 1]$ is the normalized time, and $b_{i,n}(s)$ is the i -th Bernstein Polynomial of degree n

$$b_{i,n}(s) = \binom{n}{i} s^i (1-s)^{n-i}. \quad (27)$$

Also, $x(s)$ and $z(s)$ are $(n+2)^{\text{th}}$ -order Bézier polynomials

$$x(s) = \sum_{i=0}^{n+2} c_{i,x} b_{i,n+2}(s) \quad (28)$$

$$z(s) = \sum_{i=0}^{n+2} c_{i,z} b_{i,n+2}(s) \quad (29)$$

for some set of $c_{i,x}$ and $c_{i,z}$.

By taking the second derivative of (28) using common formula [16] for Bernstein polynomials, the dynamics (9) can be used to form $n + 1$ linear equations that relate the Bezier coefficients for u_x to those for x . An additional two linear equations relating the Bezier coefficients to initial conditions x_0 , \dot{x}_0 can be formed to provide

$$\mathbf{C}_x \mathbf{c}_x = \mathbf{b}_x(\beta_x, x_0, \dot{x}_0) \quad (30)$$

where, $\mathbf{c}_x \in \mathbb{R}^{n+3}$ is the vector consisting of the coefficients $c_{i,x}$, and \mathbf{C}_x is a constant matrix. Thus, given a selection of a horizontal ground force profile, along with initial conditions, (30) can be quickly solved to obtain the Bezier coefficients \mathbf{c}_x for x . A similar approach can be used to find the Bezier coefficients \mathbf{c}_z for z

$$\mathbf{C}_z \mathbf{c}_z = \mathbf{b}_z(\beta_z, z_0, \dot{z}_0). \quad (31)$$

Similarly, since x and z are $(n + 2)^{\text{th}}$ -order Bézier polynomials and $u_x(s)$ and $u_z(s)$ are n^{th} -order Bézier polynomials, we can easily deduce that the right hand side of (11), $-\alpha x u_z + \alpha z u_x$, is an $(2n + 2)^{\text{th}}$ -order polynomial. Thus θ is a $(2n + 4)^{\text{th}}$ -order Bézier polynomial

$$\theta(s) = \sum_{i=0}^{2n+4} c_{i,\theta} b_{i,2n+4}(s) \quad (32)$$

for some set of $c_{i,\theta}$. Common formula [16] for the products of Bernstein polynomials can be applied to express the right hand side of (11) using Bézier coefficients. Equating these coefficients to those with the second derivative of (32), and using the initial conditions θ_0 and $\dot{\theta}_0$ provides

$$\mathbf{C}_\theta \mathbf{c}_\theta = \mathbf{b}_\theta(\beta_x, \beta_y, \mathbf{c}_x, \mathbf{c}_z, \theta_0, \dot{\theta}_0), \quad (33)$$

where, $\mathbf{c}_\theta \in \mathbb{R}^{2n+5}$ is the vector consisting of elements with $c_{i,\theta}$. Hence, Bézier coefficients of $x(s)$, $z(s)$, $\theta(s)$ are easily obtained by solving (30), (31), and (33). Following this procedure, the analytical formula for x , z , and θ from (28), (29), and (32) can be used to quickly query the state of the system at any time $t \in [t_0, t_f]$. To handle the multiple phases of motion, this procedure can be cascaded, enforcing continuous differentiability across transition boundaries.