2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Optimized Jumping on the MIT Cheetah 3 Robot

Quan Nguyen[1], Matthew J. Powell[1], Benjamin Katz[1],
Jared Di Carlo[2], and Sangbae Kim[1]

Abstract— **This paper presents a novel methodology for implementing optimized jumping behavior on quadruped robots. Our method includes efficient trajectory optimization, precise high-frequency tracking controller and robust landing controller for stabilizing the robot body position and orientation after impact. Experimental validation was successfully conducted on the MIT Cheetah 3, enabling the robot to repeatably jump onto and jump down from a desk with the height of $30''$ (0.76 m). The result demonstrates the advantages of the approach as well as the capability of the robot hardware itself.**

I. Introduction

Legged robots have great potential for application in challenging man-made and natural environments. In contrast to wheeled or tracked vehicles, legs provide remarkable advantages for navigating uneven terrains, especially high obstacles. Designing and controlling machines to realize these potentials has long motivated work across the legged robotics community. Especially, the recent DARPA Robotics Challenge (DRC) [1] motivates the development of highly-capable quadrupeds (e.g., [2], [3]) and humanoids (e.g., [4], [5]).

Quadruped robots have recently shown impressive advancements in dynamic locomotion capabilities using various actuation methods and control strategies. The use of hydraulic actuators has proven to be successful in legged locomotion with Big Dog by Boston Dynamics [6] and IIT's HyQ quadruped [7]. These robots take advantage of the hydraulic actuation system's ability to output large forces at the joints. ANYmal at ETH [2] features an extensive range of motion for completing autonomous tasks in real-world situations with the use of Series Elastic Actuators (SEAs). The MIT Cheetah 2 made use of a custom proprioceptive actuator design [8] that possesses high impact mitigation, force control, and position control capabilities. This design enabled it to autonomously jump over obstacles [3] and bound at speeds of 6m/s [9]. While this prior work uses a simple model for real-time planning, in this paper, we maximize the performance of the robot to make it jump as high as possible using trajectory optimization based on the full dynamical model of the system.

A recent work [10] presented a design and control framework for optimized jumping of a single-legged robot. The system is in a small scale with a link length of 12 cm and attached to a boom, restricting its motion in the lateral direction. In this paper, using different methods of optimization and control, we are able to successfully implement optimized jumping on a standard-size desk with a height of $30''$ with an untethered, large-scale quadrupedal robot, MIT Cheetah 3 (see Figure 1). Miniature robot jumpers (e.g., Salto [11], EPFL Jumper [12]) can achieve impressive high jumping thanks to theirs small weights as well as specific design for jumping tasks usually by storing energy using springs.

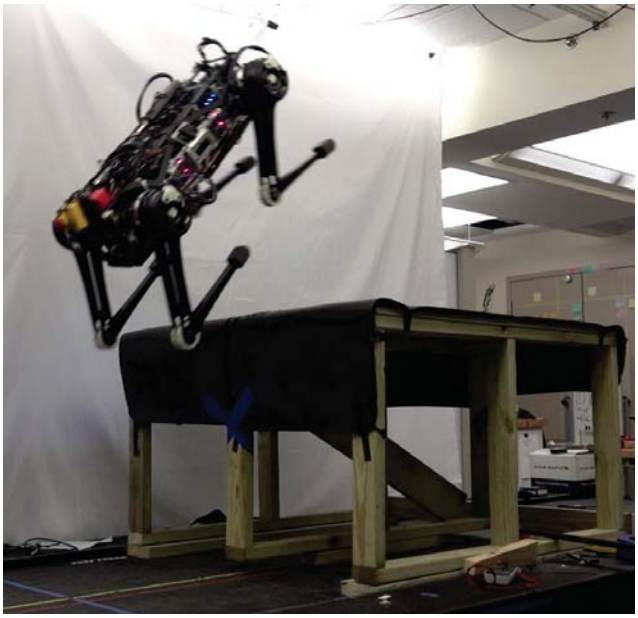


Fig. 1: **MIT Cheetah 3 jumps on a desk.** With our method, the robot was able to jump on a 30-inch desk.

The remainder of the paper is organized as follows. Section IV describes the trajectory optimization and the jumping controller. The force-based landing controller is presented in Section V. Results from selected hardware experiments are shown in Section VI. Finally, Section VII provides concluding remarks.

II. Hardware Platform

The hardware design of Cheetah 3 builds on the actuation paradigm of the MIT Cheetah 1 and 2 robots [13]. By using high torque density electric motors with back-drivable single-stage planetary gear reductions, and low-inertia legs, the Cheetah 3 robot can control ground reaction forces through proprioception, without the use of any force sensors, torque sensors, or series compliance at the joints or feet. The Cheetah 2 robot was designed primarily for fast locomotion in the sagittal plane, and used high performance actuators at the hip and knee joints, but not for abduction/adduction (ab/ad).

[1]Department of Mechanical Engineering, MIT, Cambridge, MA 02139, USA: qtn@mit.edu,mjpowell@mit.edu, benkatz@mit.edu, sangbae@mit.edu
[2]Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA 02139, USA: dicarloj@mit.edu

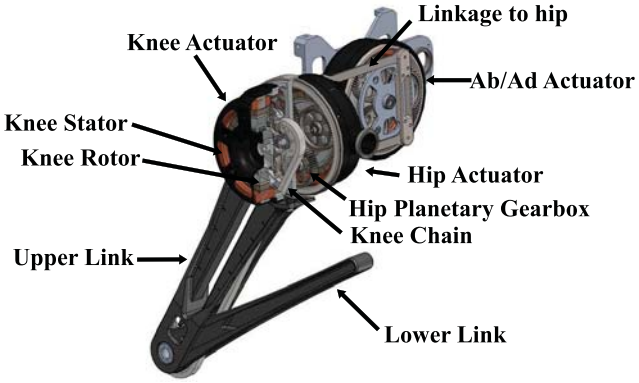


Fig. 2: **Cheetah 3 Leg Design.** Each leg consists of three actuators.

TABLE I: Actuator Parameters

Parameter	Value	Units
Gear Ratio	7.67	
Max Torque	230	Nm
Max Joint Speed	21	Rad/s

TABLE II: Physical Robot Parameters

Parameter	Symbol	Value	Units
Mass	m	45	kg
Body Inertia	I_{xx}	0.35	$\text{kg} \cdot \text{m}^2$
	I_{yy}	2.1	$\text{kg} \cdot \text{m}^2$
	I_{zz}	2.1	$\text{kg} \cdot \text{m}^2$
Body Length	l_{body}	0.600	m
Body Width	w_{body}	0.256	m
Body Height	h_{body}	0.200	m
Leg Link Lengths	l_1, l_2	0.34	m

Cheetah 3 has nearly identical actuators on all three degrees of freedom on each leg, enabling fully 3D control of ground reaction forces. The Cheetah 3 legs feature a large range of motion. The ab/ad joints have a range of motion of more than $\pm 45°$, and the hip and knee designs allow the robot operate identically forwards, backwards, and flipped upside-down. Each of Cheetah 3's actuators consists of a custom high torque density electric motor coupled to a single-stage 7.67:1 planetary gear reduction. The lower link is driven by a roller chain which passes through the upper link, providing an additional 1.15x gear reduction. The legs are serially actuated, but to keep leg inertia low, the hip and knee actuators are co-axially located at the hip of each leg as shown in Figure 2. Cheetah 3's actuation capabilities are summarized in Table I.

Having presented the robot hardware, in the next Section, we describe the development of algorithm for the jumping tasks.

III. Trajectory Optimization

As seen in quadrupeds animals, high-power jumping is normally restricted on about a 2D plane due to the complexity of the motion. Therefore, in this paper, we focus on optimizing jumping trajectory on the sagittal plane. In this Section, we discuss the construction of a jumping trajectory optimization which employs a simplified dynamics model.

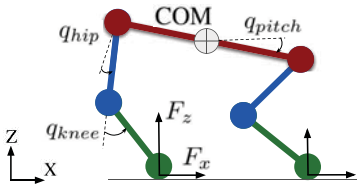


Fig. 3: **2D Quadruped Model.** The jumping trajectory optimization employs a simplified, planar model of Cheetah 3. This figure shows the choice of coordinates and the control inputs of the simplified model.

A. Dynamics

The robot model used for the optimization is simplified as a 5-link robot on a vertical plane. The physical robot parameters are listed in Table II.

The configuration variables are defined as follows:

$$\boldsymbol{Q} := [x;\ z;\ q_{pitch};\ \boldsymbol{q}], \tag{1}$$

where x, z, q_{pitch} are the COM position and body pitch angle in the world frame, and $\boldsymbol{q}$ is the joint angle vector including hip and knee joint angles of front and back legs. Figure 3 illustrates the system states and control inputs of the dynamics.

Spatial vector algebra [14] is used to construct the robot's equations of motion which take the following form:

$$\begin{aligned} &\boldsymbol{H}(\boldsymbol{Q})\ddot{\boldsymbol{Q}} + \boldsymbol{C}(\boldsymbol{Q}, \dot{\boldsymbol{Q}})\dot{\boldsymbol{Q}} + \boldsymbol{g}(\boldsymbol{Q}) \\ &= \boldsymbol{B}\boldsymbol{\tau} + \boldsymbol{B}_{fric}\boldsymbol{\tau}_{fric}(\dot{\boldsymbol{Q}}) + \sum_i \boldsymbol{J}_i^T(\boldsymbol{Q})F_i, \end{aligned} \tag{2}$$

where $\boldsymbol{H}$ is the mass matrix; the matrix $\boldsymbol{C}$ contains Coriolis and centrifugal terms; $\boldsymbol{g}$ is the gravity vector; $\boldsymbol{J}_i$ is the spatial Jacobian of the body containing the i^{th} foot, expressed at the foot and in the world coordinate system; $\boldsymbol{F}_i$ is the spatial force at the i^{th} foot; matrices $\boldsymbol{B}$ and $\boldsymbol{B}_{fric}$ define how the actuator torques $\boldsymbol{\tau}$ and the joint friction torques $\boldsymbol{\tau}_{fric}$ enter the model.

In addition to that, the following constraints are enforced for each stance foot i that is on the ground and actively supporting the robot:

$$\boldsymbol{J}_{i,stance}(\boldsymbol{Q})\ddot{\boldsymbol{Q}} + \dot{\boldsymbol{J}}_{i,stance}(\boldsymbol{Q})\dot{\boldsymbol{Q}} = 0. \tag{3}$$

Having presented the dynamical model of the robot, we now will explain how we setup the contact sequence in the optimization problem.

B. Contact scheduling

The optimization problem is constructed as follow. Initially, the trajectory of the jumping motion is discretized with a fixed sampling time of $T = 10\ ms$. The duration of the motion is therefore determined by the number of time steps N, which is manually tuned in our framework.

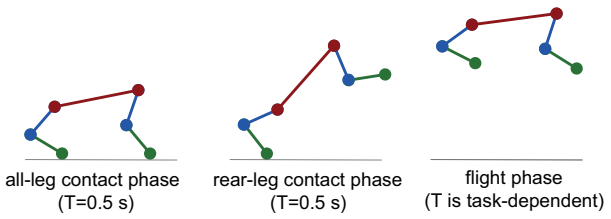


Fig. 4: **Jumping phases.** Each jumping motion includes three phases.

Subsequently, four sequential phases for each jumping motion can be identified: (a) all-leg contact phase, (b) rear-leg contact phase, (c) flight phase and (d) landing phase. While the first three phases are solved via off-line optimization, the last one, namely, the landing phase uses real-time feedback force-based controller to handle impact and balance the whole body motion. The duration of the all-leg contact phase and the rear-leg contact phase are fixed as $0.5\ s$. Based on the jumping task, the duration of the flight phase is adjusted accordingly (see Figure 4 for three different jumping phases and timing.) The dynamics of each phase is then modified based on its contact model.

We then use the nonlinear optimization tool CasADi (see [15]) for constructing and solving our optimal control problem. We now will describe our cost function and constraints used for optimizing the jumping task.

C. Cost function

The following cost function is used for our optimization:

$$J = \sum_{k=1}^{N-1} w_q(\boldsymbol{q}_k - \boldsymbol{q}_{ref})^T(\boldsymbol{q}_k - \boldsymbol{q}_{ref}) + w_\tau \boldsymbol{\tau}_k^T \boldsymbol{\tau}_k + w_N(\boldsymbol{q}_N - \boldsymbol{q}_N^d)^T(\boldsymbol{q}_N - \boldsymbol{q}_N^d), \quad (4)$$

where $\boldsymbol{q}_k, \boldsymbol{\tau}_k$ are the joint angle and joint torque at the iteration k^{th}; $\boldsymbol{q}_N$ is the joint angle at the end of the trajectory; w_q, w_τ, w_n are cost function weights of corresponding elements.

Note that the goal of this optimization framework is to find a feasible jumping motion for the robot to clear a very high obstacle with respect to physical constraints related to the hardware as well as the jumping task. Due to the complexity of this problem, the feasibility domain is very limited. Therefore, the purpose of this cost function is to guide the optimization to converge to a feasible solution where the joint angle profile does not vary arbitrary or in other words, stays close to the reference configuration $\boldsymbol{q}_{ref}$ if possible. In the cost function, we also minimize the energy consumption to ensure that the joint torque profile is not too noisy. However, we just use a very small weight of $w_\tau = 0.0001$ (while $w_q = 1, w_N = 100$) to make sure that we do not over-regulate the use of joint torques because the primary goal of this work is to maximize the performance of the robot.

D. Constraints

The following constraints are enforced in our optimization:

- Dynamics constraints (2)
- Joint angle limits: $\boldsymbol{q}_{min} \leq \boldsymbol{q}_k \leq \boldsymbol{q}_{max}$
- Joint velocity limits: $|\dot{\boldsymbol{q}}_k| \leq \dot{\boldsymbol{q}}_{max}$
- Torque limits: $|\boldsymbol{\tau}_k| \leq \boldsymbol{\tau}_{max}$
- Friction cone limits: $|\boldsymbol{F}_k^x / \boldsymbol{F}_k^z| \leq \mu$
- Minimum ground reaction force (GRF): $\boldsymbol{F}_k^z \geq \boldsymbol{F}_{min}^z$
- Initial joint configuration: $\boldsymbol{q}_0 = \boldsymbol{q}_{0,d}, \dot{\boldsymbol{q}}_0 = \boldsymbol{0}$
- Initial body configuration: $x_0 = 0,\ z_0 = 0,\ q_{pitch,0} = 0,\ \dot{x}_0 = 0,\ \dot{z}_0 = 0,\ \dot{q}_{pitch,0} = 0$ (the origin of the world frame is set at the initial position of the robot's COM)
- Pre-landing configuration: $\boldsymbol{q}_k = \boldsymbol{q}_{N,d}, \dot{\boldsymbol{q}}_k = \boldsymbol{0}$ $(k \in [(N-40) : N])$
- Final body configuration: $x_N \geq d_{jumping},\ z_N = h_{platform},\ q_{pitch,N} = 0$
- Geometric constraints related to ground and obstacle clearance

To be more specific, the dynamics constraints are used to represent the dynamics described in Section III-A, where the constraints (3) are then modified accordingly based on different contact models of different jumping phases during the motion (see Figure 4). While we use the maximum torque $\boldsymbol{\tau}_{max}$ from the real hardware for stance phases, a smaller value of torque limit during swing phases is applied to avoid aggressive swing motion. We also use a conservative friction coefficient $\mu = 0.4$ to compensate the model mismatch between the optimization model and the real hardware (the real surface friction coefficient is estimated about $\hat{\mu} = 0.7$). Furthermore, because implementing the optimized trajectory on the hardware may cause the robot to land either sooner or later than the expected time from the optimization, instead of enforcing the robot to reach the final configuration $\boldsymbol{q}_{N,d}$ at the final iteration N^{th}, we want the robot to reach the desired static posing configuration 40 iterations $(0.4\ s)$ ahead of time. This feature is important to ensure a favorable landing configuration and to avoid bad impact with the platform or ground. The constraint on the final body configuration ensures that the robot lands inside the platform.

Remark 1: In addition to that, the choice of the desired posing configuration $\boldsymbol{q}_{N,d}$ also plays a crucial role in ensuring smooth landing. Having the legs extended or close to singularity upon impact will result in high knee velocities due to the property of the Jacobian matrix. Furthermore, note that exceeding the maximum value of motor velocity may cause over-current, which was observed in our experiment.

Geometric constraints related to ground and obstacle clearance ensure the knees and swing feet have a good clearance with the ground; the whole robot body and legs have a good clearance with the platform.

IV. Jumping Controller

From the optimization, we get the desired joint angle ($\boldsymbol{q}_d$), joint velocity ($\dot{\boldsymbol{q}}_d$) and feed-forward joint torque ($\boldsymbol{\tau}_d$). Those

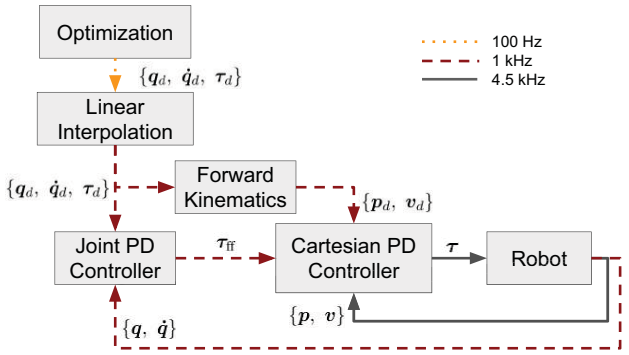


Fig. 5: **Block Diagram for the Jumping Controller.** The optimization produces discrete trajectory at 100 Hz, which is then linearly interpolated to get the desired profile at 1 kHz. The joint PD controller and the Cartesian PD controller execute at 1 kHz and 4.5 kHz respectively.

trajectories have the sampling time of 10 ms. In order to get the reference trajectory for our controller running at 1 kHz or with the sampling time of 1 ms, we do linear interpolation between the two nearby solutions from the optimization. Figure 5 presents the block diagram of the jumping controller, including the explanation of different control levels with different sampling times. The feedback joint PD controller running at 1 kHz is computed as follows:

$$\boldsymbol{\tau}_{\text{ff}} = \boldsymbol{K}_{p,joint}(\boldsymbol{q}_d - \boldsymbol{q}) + \boldsymbol{K}_{d,joint}(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}) + \boldsymbol{\tau}_d \quad (5)$$

where $\boldsymbol{K}_{p,joint}$ and $\boldsymbol{K}_{d,joint}$ are diagonal matrices of proportional and derivative gains for the joint coordinate.

In addition to that, we also use Cartesian PD controller run at 4.5 kHz to improve the tracking performance. From the desired trajectory of joint angles ($\boldsymbol{q}_d$) and joint velocities ($\dot{\boldsymbol{q}}_d$), using forward kinematic, we can extract the foot positions ($\boldsymbol{p}_d$) and foot velocities ($\boldsymbol{v}_d$) of each foot with respect to their hips.

The full controller for tracking desired trajectories is

$$\boldsymbol{\tau} = \boldsymbol{J}(\boldsymbol{q})^\top [\boldsymbol{K}_p(\boldsymbol{p}_d - \boldsymbol{p}) + \boldsymbol{K}_d(\boldsymbol{v}_d - \boldsymbol{v})] + \boldsymbol{\tau}_{\text{ff}} \quad (6)$$

where $\boldsymbol{J}(\boldsymbol{q})$ is the foot Jacobian at the configuration $\boldsymbol{q}$; $\boldsymbol{K}_p$ and $\boldsymbol{K}_d$ are diagonal matrices of proportional and derivative gains for the Cartesian coordinate; $\boldsymbol{\tau}_{\text{ff}}$ is the feed-forward torque from the joint PD controller (5).

V. Landing Controller

Having introduced the jumping controller, in this Section we present a force-based controller to handle impact and balance the whole body motion during the landing phase. Due to model mismatch between simulation and experiment, there is a significant pitch angle error when the robot is about to land either on the table or the ground. Therefore, the landing controller plays a crucial role to recover the robot from unexpected landing configurations. The experimental validation of this controller also proves clearly the ability to handle hard impact and control ground reaction forces of the hardware design in general and the motors in particular.

A. Contact Detection

Firstly, we will present the contact detection method we use to trigger the switch from the jumping controller presented in Section IV to the landing controller.

For walking experiments in the Cheetah 3 robot, we need to use a complex contact fusion algorithm based on gait timing and other criteria (see [16]). For jumping behavior, because the real jumping time is normally far off from the expected value and its ground impact is much harder than that of walking, we simply detect the impact and switch to the landing control using the following condition:

if $(t \geq T_{posing})\&(\text{any } \dot{q}_{knee,i} \geq \dot{q}_{knee,impact})$
then switch to landing control,

where T_{posing} is the instant when the desired trajectory is set to pose the robot in a static configuration to prepare for landing (after that time the desired knee velocity $\dot{q}_{knee,d} = 0$); $\dot{q}_{knee,i}$ is the knee velocity of the i^{th} foot; and $\dot{q}_{knee,impact}$ is the knee velocity threshold to determine if the ground impact happened.

Using this contact detection, we also clarify between three different contact models: front-leg contact; rear-leg contact; and all-leg contact. It will help the landing controller to determine which robot model should be used.

B. Controller Model

While in the trajectory optimization process, we use a detailed dynamical model of the robot described in Section IV, a simplified control model is used to optimize the ground reaction forces for balancing the whole body motion after impact, enabling a formation of real-time optimal controller.

By design, the robot has light limbs with low inertia as compared to the overall body. Therefore, it is reasonable to ignore the effects of the legs into the whole body motion for planning ground reaction forces. This assumption is also commonly used in control of legged robots [17], [18]. The following linear model is then derived:

$$\underbrace{\begin{bmatrix} \mathbf{I}_3 & \dots & \mathbf{I}_3 \\ [\boldsymbol{p}_1 - \boldsymbol{p}_c]\times & \dots & [\boldsymbol{p}_4 - \boldsymbol{p}_c]\times \end{bmatrix}}_{\boldsymbol{A}} \boldsymbol{F} = \underbrace{\begin{bmatrix} m(\ddot{\boldsymbol{p}}_c + \boldsymbol{g}) \\ \boldsymbol{I}_G \dot{\boldsymbol{\omega}}_b \end{bmatrix}}_{\boldsymbol{b}}, \quad (7)$$

where m and $\boldsymbol{I}_G$ are the robot's total mass and centroidal rotational inertia, $\boldsymbol{g}$ is the gravity vector and $\boldsymbol{p}_i, i \in \{1,2,3,4\}$ are the positions of the feet. The term $[\boldsymbol{p}_i - \boldsymbol{p}_c]\times$ is the skew-symmetric matrix representing the cross product $(\boldsymbol{p}_i - \boldsymbol{p}_c) \times \boldsymbol{F}_i$.

C. Force-based Landing Controller

For the landing controller, we adopted the controller described in [17] with slight modification. This controller enforces PD control on the center of mass and body orientation, while also ensuring that foot forces satisfy friction constraints as well as minimum and maximum ground

Fig. 6: Motion snapshots from a successful jump on a desk with the height of 30 inches.

Fig. 7: Motion snapshots from a successful jump down from a desk with the height of 30 inches.

reaction forces. The PD control law is given by

$$\begin{bmatrix} \ddot{\boldsymbol{p}}_{c,d} \\ \dot{\boldsymbol{\omega}}_{b,d} \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}_{p,p}(\boldsymbol{p}_{c,d} - \boldsymbol{p}_c) + \boldsymbol{K}_{d,p}(\dot{\boldsymbol{p}}_{c,d} - \dot{\boldsymbol{p}}_c) \\ \boldsymbol{K}_{p,\omega}\log(\boldsymbol{R}_d\boldsymbol{R}^T) + \boldsymbol{K}_{d,\omega}(\boldsymbol{\omega}_{b,d} - \boldsymbol{\omega}) \end{bmatrix}. \tag{8}$$

The desired angular acceleration reflects PD control on $SO(3)$ wherein the desired and actual body orientations are described using rotation matrices $\boldsymbol{R}_d$ and $\boldsymbol{R}$, respectively, and the orientation error is obtained using the exponential map representation of rotations [19], [20].

The goal of this controller is to resolve an optimal distribution of leg forces $\boldsymbol{F}$ that drive the approximate COM dynamics to the corresponding desired dynamics given by

$$\boldsymbol{b}_d = \begin{bmatrix} m(\ddot{\boldsymbol{p}}_{c,d} + \boldsymbol{g}) \\ \boldsymbol{I}_G\,\dot{\boldsymbol{\omega}}_{b,d} \end{bmatrix}. \tag{9}$$

Since the model (7) is linear, the controller can naturally be expressed as a quadratic program (QP) [21], that can be solved in real-time of 1 kHz:

$$\begin{aligned} \boldsymbol{F}^* = \min_{\boldsymbol{F}\in\mathbb{R}^{12}} \; & (\boldsymbol{A}\boldsymbol{F} - \boldsymbol{b}_d)^T\boldsymbol{S}(\boldsymbol{A}\boldsymbol{F} - \boldsymbol{b}_d) \\ & + \alpha\|\boldsymbol{F}\|^2 + \beta\|\boldsymbol{F} - \boldsymbol{F}^*_{\text{prev}}\|^2 \\ \text{s.t.} \quad & \boldsymbol{C}\boldsymbol{F} \le \boldsymbol{d} \\ & \boldsymbol{F}^z_{swing} = \boldsymbol{0} \end{aligned} \tag{10}$$

The cost function in (10) represents three goals: driving the COM to the desired dynamics, minimizing the force commands, and filtering the change of the solution $\boldsymbol{F}^*$ with respect to the previous time-step, $\boldsymbol{F}^*_{prev}$. The weight parameters $\boldsymbol{S}, \alpha, \beta$ reflect the relative priority in different elements. Constraints $\boldsymbol{C}\boldsymbol{F} \le \boldsymbol{d}$ ensure that the optimized forces stay inside the friction pyramid and that the normal forces lie within feasible bounds.

Based on the contact model coming out from the contact detection, the normal forces of swing legs are constrained to be zeros, $\boldsymbol{F}^z_{swing} = \boldsymbol{0}$. Having this constraint in addition to the friction cone constraints in the QP controller, we will enforce the force vectors of the swing legs to be zeros, $\boldsymbol{F}_{swing} = 0$. The swing legs are then kept at the posing position using PD control described in Section IV until the contact detection in those legs are triggered.

Remark 2: Note that, we initially try to set all legs to be stance legs whenever we detect ground impact at any legs. However, in experiment, the robot normally has a rear-leg contact with the desk or ground first with a significant pitch angle offset at impact (see Figures 6 and 7). Therefore, if we apply force control for all legs in this configuration, the front legs will extend fast, leading to excessive impact force and knee velocity post-impact. Switching from swing phase using position control to stance phase using force control properly helped us to overcome this issue.

VI. Results

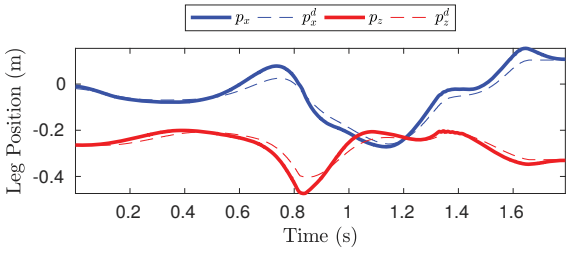
This section presents results from experimental testing with the MIT Cheetah 3 robot. A video of the results is included as supplementary material.

Using our approach presented throughout this paper, the robot can repeatedly jump onto and jump down from different platforms with different height of $\{14'', 20'', 30''\}$. We pick the experiment with the 30-inch desk as illustrated in Figure 6 and 7 to discuss in this paper.

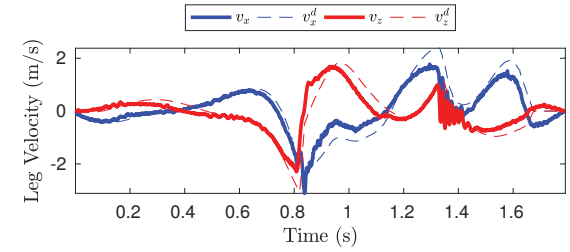
While the experiment of jumping on the desk shows clearly the effectiveness of the proposed method in general, the jumping down experiment emphasizes the performance of the landing controller presented in Section V.

Figure 8 shows the tracking performance of the foot position and velocity for the jumping phase of the first experiment (jumping onto a 30-inch desk, see Figure 6). With the Cartesian PD controller running at a high frequency of $4.5\ kHz$, our robot can achieve good tracking performance both during stance and swing phases.

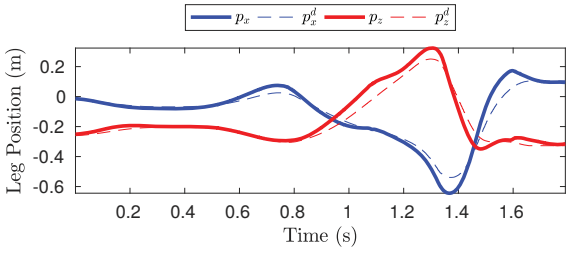
Accordingly, during the jumping phase with rear-leg contact, the system is under-actuated and we don't do feedback control on the body pitch angle. Thus, there is normally a significant pitch angle error with the magnitude of more than 20° upon landing as evident in Figure 9b. Therefore, it is very important to have a landing controller that can robustly recover the whole body position and orientation under a large error of landing configuration. Figure 9 shows the performance of the landing controller when the robot jumped down from a 30-inch desk (see Figure 7). In this experiment, the rear legs had impact with the ground when



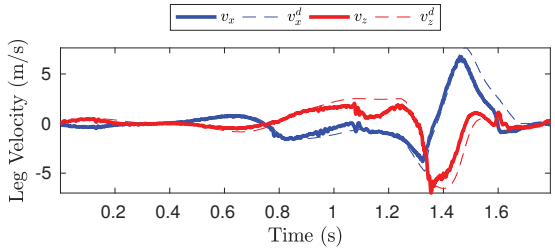
(a) Front-right leg position



(b) Front-right leg velocity

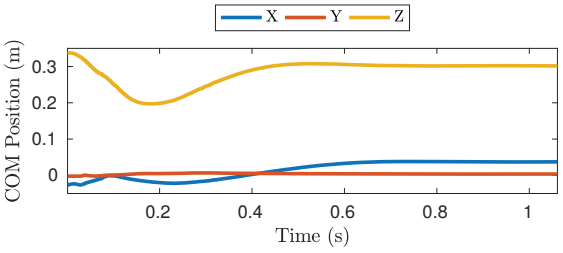


(c) Back-right leg position

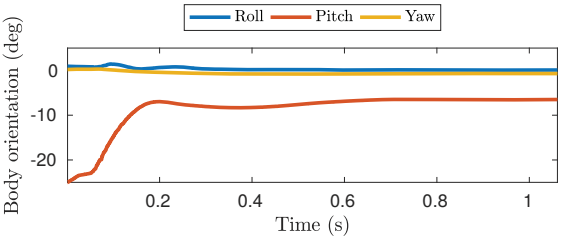


(d) Back-right leg velocity

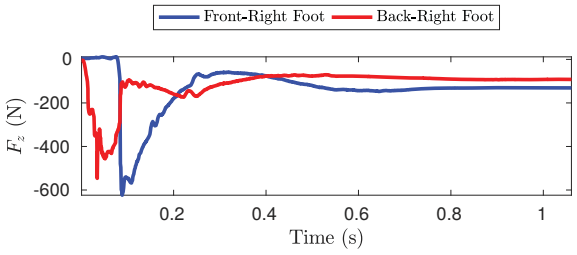
Fig. 8: **Tracking performance of the jumping controller.** This plot shows the tracking of leg position and velocity during the jumping phase when the robot jumped onto a 30-inch desk (see Figure 6.)



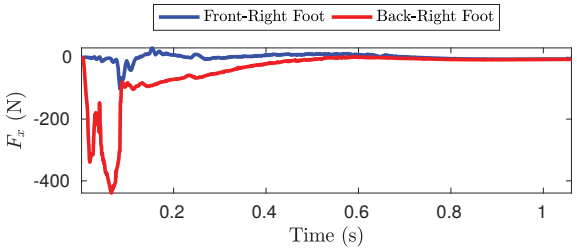
(a) COM Position



(b) Body Orientation



(c) Vertical Force F_z



(d) Horizontal Force F_x

Fig. 9: **Force command of the landing controller.** This plot shows the COM position, body orientation and force command during the landing phase when the robot jumped down from a 30-inch desk (see Figure 7).

the pitch angle was $-25°$. However, the landing controller was able to recover the pitch angle while maintaining roll and yaw angle about zeros using reasonable force commands as illustrated in Figure 9.

VII. Conclusions and Future Work

This paper has introduced a framework for implementing jumping behaviors on quadruped robots. The combination of trajectory optimization, high-frequency tracking controller and robust landing control enabled the robot to repeatedly jump onto and jump down from a 30-inch desk. The experimental results not only validate the advantages of the method but also prove the capability of the hardware on performing aggressive tasks that require generating high power as well as absorbing high impact. Our future plan is to integrate the approach with vision to autonomously detect and jump on obstacles.

Acknowledgments

The authors thank Gerardo Bledt and the other members of the MIT Biomimetic Robotics Lab for the support and assistance during the experimental implementation.

REFERENCES

[1] "DARPA Robotics Challenge." http://www.theroboticschallenge.org/, Nov. 2012.

[2] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger, "ANYmal - a highly mobile and dynamic quadrupedal robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 38–44, Oct 2016.

[3] H.-W. Park, P. Wensing, and S. Kim, "Online planning for autonomous running jumps over obstacles in high-speed quadrupeds," in *Proceedings of Robotics: Science and Systems*, (Rome, Italy), July 2015.

[4] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, pp. 1–27, 2015.

[5] N. A. Radford and et al., "Valkyrie: Nasa's first bipedal humanoid robot," *Journal of Field Robotics*, vol. 32, no. 3, pp. 397–419, 2015.

[6] M. Raibert, K. Blankespoor, G. Nelson, R. Playter, and the BigDog Team, "Bigdog, the rough-terrain quaduped robot," in *Proceedings of the 17th World Congress*, pp. 10822–10825, 2008.

[7] C. Semini, V. Barasuol, J. Goldsmith, M. Frigerio, M. Focchi, Y. Gao, and D. G. Caldwell, "Design of the hydraulically actuated, torque-controlled quadruped robot hyq2max," *IEEE/ASME Transactions on Mechatronics*, vol. 22, pp. 635–646, April 2017.

[8] S. Seok, A. Wang, D. Otten, and S. Kim, "Actuator design for high force proprioceptive control in fast legged locomotion," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1970–1975, IEEE, 2012.

[9] H.-W. Park, P. M. Wensing, and S. Kim, "High-speed bounding with the MIT Cheetah 2: Control design and experiments," *International Journal of Robotics Research*, vol. 36, no. 2, pp. 167–192, 2017.

[10] Y. Ding and H.-W. Park, "Design and experimental implementation of a quasi-direct-drive leg for optimized jumping," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pp. 300–305, IEEE, 2017.

[11] D. W. Haldane, M. Plecnik, J. K. Yim, and R. S. Fearing, "Robotic vertical jumping agility via series-elastic power modulation," *Science Robotics*, vol. 1, no. 1, 2016.

[12] M. Kovac, M. Fuchs, A. Guignard, J.-C. Zufferey, and D. Floreano, "A miniature 7g jumping robot," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 373–378, IEEE, 2008.

[13] P. M. Wensing, A. Wang, S. Seok, D. Otten, J. Lang, and S. Kim, "Proprioceptive actuator design in the MIT Cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 509–522, 2017.

[14] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.

[15] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, In Press, 2018.

[16] G. Bledt, P. M. Wensing, S. Ingersoll, and S. Kim, "Contact model fusion for event-based locomotion in unstructured terrains," in *2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia*, 2018.

[17] M. Focchi, A. del Prete, I. Havoutis, R. Featherstone, D. G. Caldwell, and C. Semini, "High-slope terrain locomotion for torque-controlled quadruped robots," *Autonomous Robots*, vol. 41, pp. 259–272, Jan 2017.

[18] B. Stephens and C. Atkeson, "Push recovery by stepping for humanoid robots with force controlled joints," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 52–59, Dec. 2010.

[19] F. Bullo and R. M. Murray, "Proportional derivative (PD) control on the Euclidean group," in *Proc. of the European Control Conference*, (Rome, Italy), pp. 1091–1097, June 1995.

[20] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[21] C. Gehring, S. Coros, M. Hutter, M. Bloesch, M. Hoepflinger, and R. Siegwart, "Control of dynamic gaits for a quadrupedal robot," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3287–3292, May 2013.