2020 17th International Conference on Ubiquitous Robots (UR)
June 22-26, 2020. Kyoto, Japan

Optimized Jumping of an Articulated Robotic Leg

Junjie Shen[1], Yeting Liu[1], Xiaoguang Zhang[1], and Dennis Hong[1]

Abstract— **This paper proposes a nonlinear programming (NLP) formulation intended for the trajectory optimization of legged robot jumping applications during the stance phase, taking into consideration the detailed robot model, actuator capability, terrain condition, etc. The method is applicable to a wide class of jumping robots and was successfully implemented on an articulated robotic leg for jumping in terms of maximum reachable height, minimum energy consumption, as well as optimum energy efficiency. The simulation and experimental results demonstrate that this approach is capable of not only planning one single jumping trajectory, but also designing a periodic jumping gait for legged robots.**

I. Introduction

Legged robots, despite of increased complexity and power consumption compared with other types of mobile robot, have the potential to exert a much larger influence to human environments in the future. The articulated limbs provide them with the inimitable possibility of going anywhere a human can go and doing anything a human can do. While progress has been made, legged robots are only beginning to fulfill this great potential.

Traditionally, legged robots are realized with hydraulic actuators for producing tremendous magnitude of force [1], [2]. However, their energy efficiency is limited and the actuation system is often large and difficult to install [3], [4]. Later, legged robots are equipped with electromagnetic (EM) actuators with large gear ratio in order to achieve high torque density [5]. Nevertheless, these heavily geared motors, specifically designed to perform accurate position-controlled tasks in fairly structured environments, are quite vulnerable when faced with significant ground impact for legged locomotion, due to increased reflected inertia and gear friction from the gearbox [6]. Recently, more advanced legged robots have been developed which are capable of dynamic locomotion over irregular terrain with the help of series elastic actuators (SEA) [7], [8]. SEAs are utilized to mitigate the ground impact by intentionally adding controlled variable mechanical impedance in series with an actuator [9]. However, their force bandwidth is limited [10], [11]. In addition, these legged systems with added mechanical impedance usually result in complex dynamics, making them difficult to control at best and restricted in their capabilities at worst. Lately, impressive advances in EM technology, i.e., direct-drive [6] and quasi-direct-drive motors [12], [13], [14], have demonstrated that they are capable of producing sufficient torque and speed for legged robot dynamic locomotion without high gearing [15], [16]. This leads to the benefit of high transparency and mechanical performance [6], which enables accurate modeling and control of legged systems with straightforward torque inputs. In this paper, we will only focus on legged robots with this type of actuator.

The capability of jumping motions is one of the main characters distinguishing legged robots from other types of mobile robot, which has been extensively studied for several decades. Raibert and Brown developed a one-legged hopping machine with springy leg of telescopic type and realized a hopping gait with an empirical controller [2]. Using a similar controller, Hyon and Mita designed another one-legged hopping robot that had an articulated leg composed of three links [17]. A leg spring was further utilized not only to enhance energy efficiency but also to absorb large impulse at touch-down. Arikawa and Mita proposed a practical motion planning method for jumping motions of multi-degree-of-freedom jumping robot on the basis of the boundary state at take-off and verified it with simulation of normal jump as well as somersault [18]. Hutter et al. employed an operational space controller to impose the behavior of the spring loaded inverted pendulum (SLIP) model on an articulated robotic leg, which was then capable of continuous hopping on uneven ground [19].

Though many legged systems have been shown to be capable of jumping motions, not until recent the mathematical optimization technique is applied to optimize the control strategy for some specific jumping tasks. Lim et al. reduced the jumping trajectory optimization of biarticular legged robots to a parameter optimization problem by parameterizing the joint trajectory in terms of B-splines [20]. However, joint torque cannot be constrained directly but using a penalty function and ground friction was not considered. Hiasa et al. used a similar parameterization for each joint as reference and were able to constrain joint torque directly with their nonlinear optimization simulation approach [21]. However, the optimized joint torque was computed from a PD controller by following the parameterized joint reference, which was far from the ideal values. Ding and Park proposed a sequential nonlinear optimization process that simultaneously solved for optimal control input as well as chose mechanical design parameters specifically for single robotic leg jumping task [22]. Nonetheless, the leg was modeled as a simple point mass at the base with ground reaction force (GRF) as input and thus the experimental results deviated from the optimized results largely. Without considering ground friction, slippage happened during the

[1]Junjie Shen, Yeting Liu, Xiaoguang Zhang, and Dennis Hong are with the Robotics and Mechanisms Laboratory, the Department of Mechanical and Aerospace Engineering, University of California, Los Angeles, CA 90095, USA {junjieshen, liu1995, hawkblizzard, dennishong}@ucla.edu

978-1-7281-5714-6/20/$31.00 ©2020 IEEE

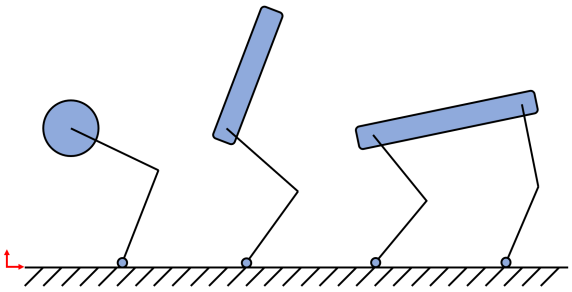

Fig. 1. Examples of 2D legged robot. On the left is the single leg model; in the middle is the simplified biped model; and on the right is the simplified quadruped model. The world frame is in red.

experiment. Besides, the optimized results were conservative by enforcing initial GRF to zero. Nguyen et al. presented a different nonlinear optimization method for quadruped robots and successfully implemented it on the MIT Cheetah 3 [23]. Nevertheless, the purpose of their cost function is more like to find a feasible solution than to optimize the jumping performance. Besides, only one single jump is considered.

Inspired by the previous work, this paper proposes a nonlinear programming (NLP) formulation intended for the trajectory optimization of legged robot jumping applications during the stance phase, taking into account the detailed robot model, actuator capability, terrain condition, etc. The method was successfully tested on an articulated robotic leg not only for optimizing one single jumping trajectory but also for designing a periodic jumping gait. The rest of this paper is organized as follows. Section II describes the legged robot model of interest. Section III details the trajectory optimization algorithm for legged robot jumping applications via an NLP formulation. Section IV illustrates the proposed method with a single degree-of-freedom (DOF) robotic leg for vertical jumping. Section V illustrates the proposed method with a two DOF robotic leg for jumping gait design. Section VI concludes the paper.

II. Robot Model

In this paper, we focus on optimizing jumping trajectory (during the stance phase right before take-off) for an articulated robotic leg on the sagittal plane. However, the proposed method can be generalized to a wide class of legged robots, as shown in Fig. 1. Define the vector of generalized coordinates $\boldsymbol{q} = \left[x, z, \alpha, \boldsymbol{\theta}^T\right]^T$, where x, z, α are the body position and angle, and $\boldsymbol{\theta}$ is the joint position vector including both actuated joints $\boldsymbol{\theta}_a$ and passive joints $\boldsymbol{\theta}_p$. The equations of motion take the form:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{B}\boldsymbol{\tau} + \boldsymbol{J}(\boldsymbol{q})^T \boldsymbol{d}, \tag{1}$$

where $\boldsymbol{M}(\boldsymbol{q})$ stands for the inertia matrix, the vector $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ captures the Coriolis, centrifugal, and gravitational forces, $\boldsymbol{B}$ defines how the actuation torques $\boldsymbol{\tau}$ enter the model, and the Jocabian matrix $\boldsymbol{J}(\boldsymbol{q})$ transforms external forces $\boldsymbol{d}$ into generalized forces. We can convert (1) into its state-space form for the sake of NLP formulation:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\tau}, \boldsymbol{d}), \tag{2}$$

where the state $\boldsymbol{x} = \left[\boldsymbol{q}^T, \dot{\boldsymbol{q}}^T\right]^T$ and

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\tau}, \boldsymbol{d}) = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \boldsymbol{M}(\boldsymbol{q})^{-1}\left(\boldsymbol{B}\boldsymbol{\tau} + \boldsymbol{J}(\boldsymbol{q})^T \boldsymbol{d} - \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\right) \end{bmatrix}. \tag{3}$$

Additionally, a kinematics constraint is further imposed to fix each stance foot on the ground before take-off:

$$\boldsymbol{h}(\boldsymbol{q}) = \boldsymbol{p}, \tag{4}$$

where $\boldsymbol{p}$ describes the position of stance feet and

$$\frac{\partial \boldsymbol{h}(\boldsymbol{q})}{\partial \boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q}). \tag{5}$$

III. Problem Formulation

This section illustrates how we formulate the optimization of jumping trajectory to an NLP for legged robots. First of all, a typical formulation for a mathematical optimization problem can be written as follows:

$$\begin{aligned} \underset{\boldsymbol{z}}{\text{minimize}} \quad & c(\boldsymbol{z}) \\ \text{subject to} \quad & \boldsymbol{\phi}(\boldsymbol{z}) = \boldsymbol{0}, \\ & \boldsymbol{\psi}(\boldsymbol{z}) \leq \boldsymbol{0}, \end{aligned} \tag{6}$$

where $\boldsymbol{z} \in \mathbb{R}^n$ is the vector of decision variables, $c : \mathbb{R}^n \to \mathbb{R}$ is the scalar objective function, $\boldsymbol{\phi} : \mathbb{R}^n \to \mathbb{R}^m$ is the equality constraint function, and $\boldsymbol{\psi} : \mathbb{R}^n \to \mathbb{R}^r$ is the inequality constraint function [24]. At least one of c, $\boldsymbol{\phi}$, and $\boldsymbol{\psi}$ needs to be nonlinear to make (6) an NLP.

A. Decision Variables

The optimal jumping problem for legged robots is initially a continuous-time trajectory optimization problem. To simplify the integration calculations involved, the direct collocation method is used to discretize the trajectories at N collocation points with even time intervals Δt. We find that trapezoidal collocation works well here since the duration of stance T is really short for jumping. The set of decision variables χ can first be defined as

$$\chi := \{\boldsymbol{q}[k], \dot{\boldsymbol{q}}[k], \boldsymbol{\tau}[k], \boldsymbol{d}[k] \,|\, k = 1, \ldots, N\}, \tag{7}$$

where $\boldsymbol{q}[k]$, $\dot{\boldsymbol{q}}[k]$, $\boldsymbol{\tau}[k]$, and $\boldsymbol{d}[k]$ are known as the collocation points at time $t[k] = (k-1)\Delta t$. To further ensure a smooth and physically feasible profile, polynomials of order S are used to parameterize $\boldsymbol{\tau}$ and $\boldsymbol{d}$, which gives

$$\boldsymbol{\tau}[k] = \sum_{i=0}^{S} \boldsymbol{\lambda}_i t[k]^i, \quad \boldsymbol{d}[k] = \sum_{i=0}^{S} \boldsymbol{\nu}_i t[k]^i, \quad k = 1, \ldots, N, \tag{8}$$

where $\boldsymbol{\lambda}_i$ and $\boldsymbol{\nu}_i$ are the vectors of coefficients for the monomial of order i. Parameterization is not further applied to the states $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ because they are already subject to the dynamics (1), which will end up with a smooth and feasible trajectory if $\boldsymbol{\tau}$ and $\boldsymbol{d}$ are nice functions. The set of decision variables χ thus becomes

$$\chi := \{\boldsymbol{q}[k], \dot{\boldsymbol{q}}[k] \,|\, k = 1, \ldots, N\} \cup \{\boldsymbol{\Lambda}, \boldsymbol{V}\}, \tag{9}$$

where $\boldsymbol{\Lambda} = [\boldsymbol{\lambda}_0, \ldots, \boldsymbol{\lambda}_S]$ and $\boldsymbol{V} = [\boldsymbol{\nu}_0, \ldots, \boldsymbol{\nu}_S]$. Note that the total number of decision variables decreases if $S < N - 1$.

For different applications and interests, χ can also involve decision variables such as the generalized acceleration $\ddot{\boldsymbol{q}}[k]$, the duration of stance T, the position of stance feet $\boldsymbol{p}$, the motor gear ratio γ, the link length l, etc.

B. Objective Function

The jumping performance can be evaluated in many different ways. For example, to maximize the maximum reachable height $h_{\max}$ of the center of mass (CoM) after take-off, the objective function $c(\chi)$ will be

$$c(\chi) = -h_{\max} = -z_{CoM}[N] - \frac{\dot{z}_{CoM}[N]^2}{2g}, \tag{10}$$

where g is the gravitational acceleration, $z_{CoM}[N]$ is the height, and $\dot{z}_{CoM}[N]$ is the vertical velocity component of CoM at take-off. Both of them are functions of $\boldsymbol{q}[N]$ and $\dot{\boldsymbol{q}}[N]$. In addition, when the leg dynamics is negligible compared to the body, (10) can be simplified to

$$c(\chi) = -z[N] - \frac{\dot{z}[N]^2}{2g}, \tag{11}$$

where $z[N]$ is the height and $\dot{z}[N]$ is the vertical velocity component of the body at take-off.

Another way to evaluate the jumping performance can be energy consumption E when the goal height h is fixed. It can be defined as the integral of the absolute mechanical power of the actuators over the duration of stance T:

$$E := \int_0^T |\boldsymbol{\tau}|^T \left|\dot{\boldsymbol{\theta}}_a\right| dt, \tag{12}$$

where $\dot{\boldsymbol{\theta}}_a$ is the actuated joint velocity vector. The objective function is further approximated as a summation

$$\begin{aligned} c(\chi) &= \sum_{k=1}^{N-1} \frac{1}{2} \left(|\boldsymbol{\tau}[k]|^T \left|\dot{\boldsymbol{\theta}}_a[k]\right| + |\boldsymbol{\tau}[k+1]|^T \left|\dot{\boldsymbol{\theta}}_a[k+1]\right| \right) \Delta t \\ &= \widetilde{E}. \end{aligned} \tag{13}$$

This approximation is done by applying the trapezoid rule for integration between each adjacent pair of collocation points. We can also combine (10) and (13) together with a tuning weight to take both into consideration.

C. Constraints

1) Dynamics Constraint: It is applied between every adjacent pair of collocation points using trapezoidal collocation following [25]:

$$\boldsymbol{x}[k+1] - \boldsymbol{x}[k] = \frac{\Delta t}{2} \left(\boldsymbol{f}[k+1] + \boldsymbol{f}[k] \right), \quad k = 1, \ldots, N-1, \tag{14}$$

where $\boldsymbol{f}[k] = \boldsymbol{f}\left(\boldsymbol{x}[k], \boldsymbol{\tau}[k], \boldsymbol{d}[k]\right)$ is the result of evaluating the system dynamics at each collocation point. Additional decision variable $\ddot{\boldsymbol{q}}[k]$ can be introduced to avoid calculating the inverse of the inertia matrix in $\boldsymbol{f}$.

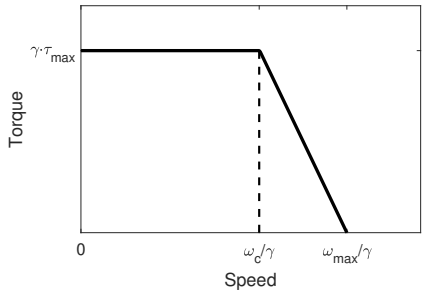

Fig. 2. A typical BLDC motor speed-torque curve scaled by gear ratio γ.

2) Kinematics Constraint: It is applied at each collocation point according to (4) and (5):

$$\boldsymbol{h}(\boldsymbol{q}[k]) = \boldsymbol{p}, \quad k = 1, \ldots, N, \tag{15}$$

which is the position kinematics constraint and

$$\dot{\boldsymbol{h}}(\boldsymbol{q}[k]) = \boldsymbol{J}(\boldsymbol{q}[k])\dot{\boldsymbol{q}}[k] = \boldsymbol{0}, \quad k = 1, \ldots, N, \tag{16}$$

which is the velocity kinematics constraint. If $\ddot{\boldsymbol{q}}[k]$ is also involved as decision variable, a third acceleration kinematics constraint needs to be imposed as well by taking further time derivative of (16).

3) Motor Constraint: A typical brushless DC electric motor (BLDC motor) speed-torque curve scaled by gear ratio γ is shown in Fig. 2. Considering the combination of both positive and negative values, the constraint on actuated joint velocity and actuation torque can be formulated as the following polyhedron at each collocation point:

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{I} & \gamma b \boldsymbol{I} \\ -\boldsymbol{I} & \gamma b \boldsymbol{I} \\ \boldsymbol{I} & -\gamma b \boldsymbol{I} \\ -\boldsymbol{I} & -\gamma b \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\tau}[k] \\ \dot{\boldsymbol{\theta}}_a[k] \end{bmatrix} \leq \begin{bmatrix} \gamma \tau_{\max} \boldsymbol{1} \\ \gamma \tau_{\max} \boldsymbol{1} \\ b \omega_{\max} \boldsymbol{1} \\ b \omega_{\max} \boldsymbol{1} \\ b \omega_{\max} \boldsymbol{1} \\ b \omega_{\max} \boldsymbol{1} \end{bmatrix}, \quad k = 1, \ldots, N, \tag{17}$$

where $\boldsymbol{I}$ is the identity matrix, $\boldsymbol{1} = [1, \ldots, 1]^T$ is of the same length as $\boldsymbol{\tau}[k]$ or $\dot{\boldsymbol{\theta}}_a[k]$, and $b = \gamma \tau_{\max} / (\omega_c - \omega_{\max})$.

4) Friction Cone Constraint: It is applied for each stance foot at each collocation point to prevent slippage:

$$\left| d_x^{(j)}[k] \right| \leq \mu \left| d_z^{(j)}[k] \right|, \quad j = 1, \ldots, M, \quad k = 1, \ldots, N, \tag{18}$$

where $\boldsymbol{d}^{(j)}[k] = \left[d_x^{(j)}[k], d_z^{(j)}[k] \right]^T$ is the GRF acting on the jth stance foot at the kth collocation point and μ is the coefficient of friction between the stance foot and ground. Since a stance foot cannot pull the ground, i.e., $d_z^{(j)}[k] \geq 0$ for all j and k, (18) is reduced to

$$\begin{bmatrix} 0 & -1 \\ 1 & -\mu \\ -1 & -\mu \end{bmatrix} \begin{bmatrix} d_x^{(j)}[k] \\ d_z^{(j)}[k] \end{bmatrix} \leq \boldsymbol{0}, \quad j = 1, \ldots, M, \quad k = 1, \ldots, N. \tag{19}$$

If the ground is inclined, (19) can be further modified to

$$\begin{bmatrix} 0 & -1 \\ 1 & -\mu \\ -1 & -\mu \end{bmatrix} \begin{bmatrix} \cos\beta & \sin\beta \\ -\sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} d_x^{(j)}[k] \\ d_z^{(j)}[k] \end{bmatrix} \leq \mathbf{0}, \\ j = 1,\ldots,M, \quad k = 1,\ldots,N, \quad (20)$$

where β is the inclination angle.

5) Take-off Constraint: Some special constraints need to be taken into consideration, which makes it different from other types of trajectory optimization for legged robots. To make sure the robot will jump (leave the ground) afterwards, the last collocation point is considered as the moment of take-off. That is, the GRF should be zero for each stance foot and the CoM velocity component normal to the ground should be positive at the last collocation point:

$$\boldsymbol{d}^{(j)}[N] = \mathbf{0}, \quad j = 1,\ldots,M, \quad (21)$$

$$-\dot{x}_{CoM}[N]\sin\beta + \dot{z}_{CoM}[N]\cos\beta \geq 0. \quad (22)$$

6) Other Constraints: Some other constraints can be included on a case-by-case basis, such as the initial and final configuration constraint, joint angle range constraint, external force constraint, etc. In general, these constraints can be formulated in a linear manner.

D. Complete Formulation

The complete NLP formulation of the jumping trajectory optimization problem for legged robots is thus given by

$$\begin{aligned} \underset{\chi}{\text{minimize}} \quad & c(\chi) \\ \text{subject to} \quad & \textit{Dynamics Constraint } (14), \\ & \textit{Kinematics Constraint } (15)(16), \\ & \textit{Motor Constraint } (17), \\ & \textit{Friction Cone Constraint } (20), \\ & \textit{Take-off Constraint } (21)(22), \\ & \textit{Other Constraints}, \end{aligned} \quad (23)$$

where the nonlinearity usually comes from the robot model, i.e., (14), (15), and (16).

IV. Example of a Single DOF Robotic Leg

This section explores the optimal jumping problem for a single DOF robotic leg via the NLP formulation.

A. Single DOF Robotic Leg Model

Fig. 3 shows the single DOF robotic leg model in detail. The body with mass M is mounted on a vertical linear guide with horizontal position $x = x_o$. Its vertical position is denoted as z. The femur and tibia links share the same mass m, length l, and inertia J. The point foot is mounted on another vertical linear guide with horizontal position p. Therefore, the friction cone constraint is not considered here. The only BLDC motor with gear ratio γ is mounted on the body and the femur joint θ_1 is actuated while the tibia joint θ_2 is passive. The equations of motion take the form as (1), where $\boldsymbol{q} = [x, z, \theta_1, \theta_2]^T$, $\boldsymbol{B\tau} = [0, 0, \tau, 0]^T$, $\boldsymbol{d} = [d_x, d_z, d_r]^T$. $[d_x, d_z]^T$ is the GRF acted on the foot and d_r is the external force exerted on the body by the linear guide in the x direction. An extra constant friction term d_f from the linear guide in the z directional is also considered in the model. The kinematics constraint (4) gives

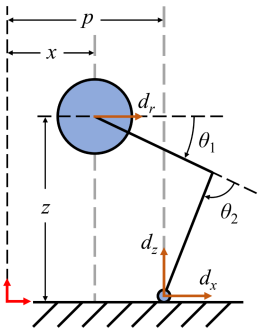

Fig. 3. Single DOF robotic leg model. The world frame is in red. The two linear giudes are shown by the dashed gray lines.

$$\boldsymbol{h}(\boldsymbol{q}) = \begin{bmatrix} x + l\cos\theta_1 + l\cos(\theta_1+\theta_2) \\ z - l\sin\theta_1 - l\sin(\theta_1+\theta_2) \\ x \end{bmatrix} = \begin{bmatrix} p \\ 0 \\ x_o \end{bmatrix}, \quad (24)$$

and the Jacobian matrix can be computed from (5) as

$$\boldsymbol{J}(\boldsymbol{q}) = \begin{bmatrix} 1 & 0 & -l(\sin\theta_1 + \sin(\theta_1+\theta_2)) & -l\sin(\theta_1+\theta_2) \\ 0 & 1 & -l(\cos\theta_1 + \cos(\theta_1+\theta_2)) & -l\cos(\theta_1+\theta_2) \\ 1 & 0 & 0 & 0 \end{bmatrix}. \quad (25)$$

The robot parameters are listed in TABLE I.

TABLE I
Robot Parameters

Symbol	Parameter	Value & Unit
M	Body mass	0.88 kg
m	Link mass	0.085 kg
l	Link length	0.2 m
J	Link inertia	4.3×10^{-4} kg·m^2
$\tau_{\max}$	Maximum torque	0.42 N·m
ω_c	Cutoff speed	1755 rad/s
$\omega_{\max}$	Maximum speed	1910 rad/s
γ	Gear ratio	24

B. Maximum Reachable Height

To maximize the maximum reachable height $h_{\max}$, the NLP is formulated as follows

$$\begin{aligned} \underset{\chi}{\text{minimize}} \quad & \textit{Objective Function } (11) \\ \text{subject to} \quad & \textit{Dynamics Constraint } (14), \\ & \textit{Kinematics Constraint } (15)(16), \\ & \textit{Motor Constraint } (17), \\ & \textit{Take-off Constraint } (21)(22), \\ & \textit{Other Constraints}, \end{aligned} \quad (26)$$

where the set of decision variables χ is defined as

$$\chi := \{\boldsymbol{q}[k], \dot{\boldsymbol{q}}[k] | k = 1,\ldots,N\} \cup \{\boldsymbol{\Lambda}, \boldsymbol{V}, p, T\}. \quad (27)$$

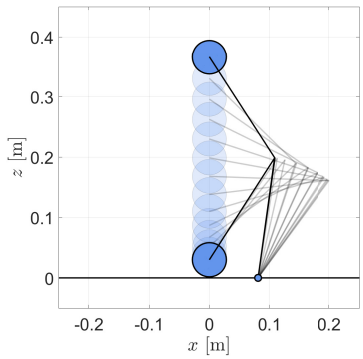

Fig. 4. Visualization for NLP results of maximum reachable height. The robotic leg starts at the bottom, pushes the ground, moves upward, and eventually takes off for a vertical jump.

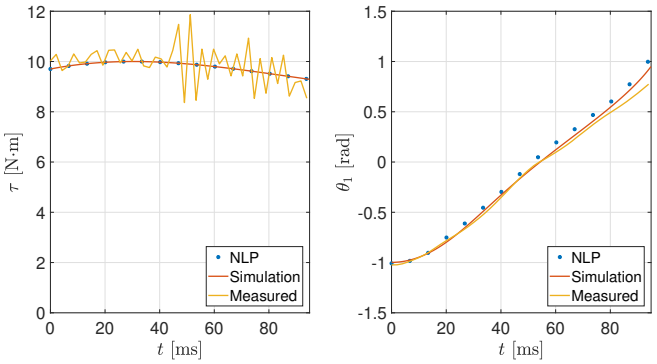

Fig. 5. Comparison between NLP, simulation, and experimental results of maximum reachable height. The optimized torque profile is implemented. It is clear that the robot model information is well captured in the NLP.

The foot position p and stance duration T are critical decision variables for this problem; x_o is really arbitrary and is set to 0 for convenience; with constant friction from the two linear guides, the gravitational acceleration g in (11) is modified to $g + d_f/(M+2m)$; the friction cone constraint is not considered but d_z is still enforced to be nonnegative all the time; the robotic leg is assumed to be static at the beginning, which is formulated as an initial configuration constraint; and the initial body height $z[1]$ is imposed to be greater than 3 cm to avoid physical interference with the ground.

MATLAB's *fmincon* function is used to solve (26) with $N = 15$ and $S = 5$ for local optima. Fig. 4 visualizes the NLP results. The NLP optimized value $h_{\max}$ is determined to be 1.066 m with optimized solution $p = 8.09$ cm, $z[1] = 3.00$ cm, and $T = 93.7$ ms. Fig. 5 compares the NLP results with the simulation and experimental results for one vertical jump, which are implemented with the same optimized joint torque profile. The sequential screenshots of the experimental results are shown in Fig. 6. The optimized torque curve implies that the maximum reachable height is achieved when the robotic leg pushes the ground as hard as possible, which also makes sense in reality. Therefore, we can actually verify the optimality using a simulation-based search with the optimal control strategy (maximum torque) over all feasible combinations of p and $z[1]$, which yields Fig. 7. The global optima $h_{\max} = 1.109$ m, which is achieved when $p = 7.75$ cm, $z[1] = 3.00$ cm, and $T = 94.1$ ms. The simulation results indicate that the NLP optimized solution is very close to the global optima of the continuous problem, which implies the robot model information is well captured in the NLP. Note that MATLAB's *ode45* function is used to simulate (1) combined with double time derivative of (4), which gives

$$\begin{bmatrix} \boldsymbol{M} & -\boldsymbol{J}(\boldsymbol{q})^T \\ \boldsymbol{J}(\boldsymbol{q}) & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ \boldsymbol{d} \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}\boldsymbol{\tau} - \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \\ -\dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} \end{bmatrix} \tag{28}$$

with initial condition satisfying (4) and $\boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{0}$. That is, given the system state and input, the acceleration and external force can be computed from (28).

C. Minimum Energy Consumption

To minimize the energy consumption E when the goal height h is fixed (thus h is involved as an additional constraint), the NLP is formulated similar to (26) while the objective function is changed to (13); the initial configuration is fixed to be able to achieve the maximum reachable height from Section IV-B, i.e., $p = 8.09$ cm (thus removed from χ) and $z[1] = 3.00$ cm are enforced as additional constraints; and the robotic leg still starts with zero state velocity. Fig. 8 shows the NLP optimized results for different goal heights.

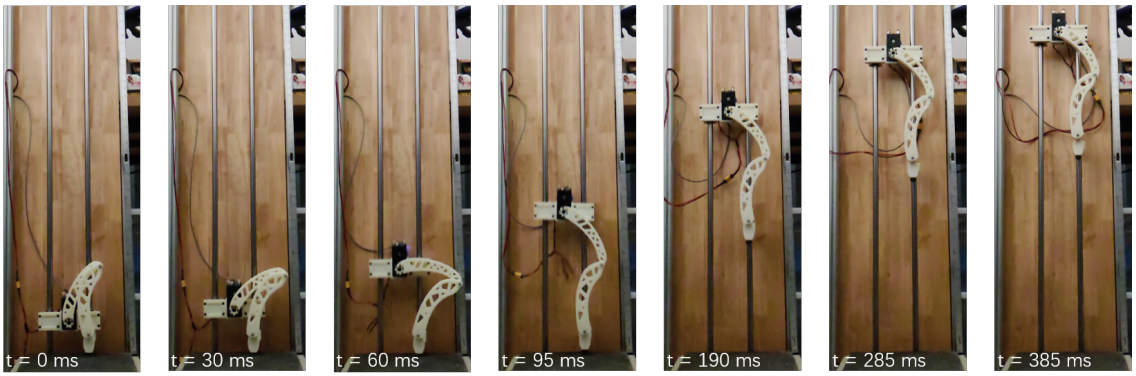

Fig. 6. Screenshots of the experimental results for maximum reachable height with measured $h_{\max} = 1.08$ m. (video link: https://youtu.be/U0KQwYinubk)

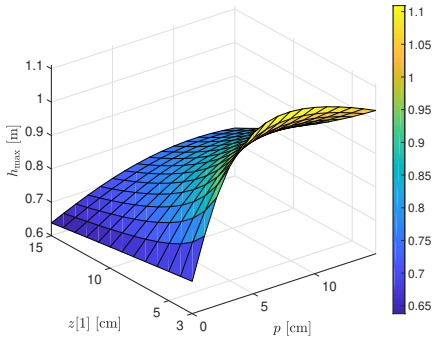

Fig. 7. Simulation results of maximum reachable height under different combinations of foot position p and initial body height $z[1]$. The global optima $h_{\max} = 1.109$ m with $p = 7.75$ cm and $z[1] = 3.00$ cm.

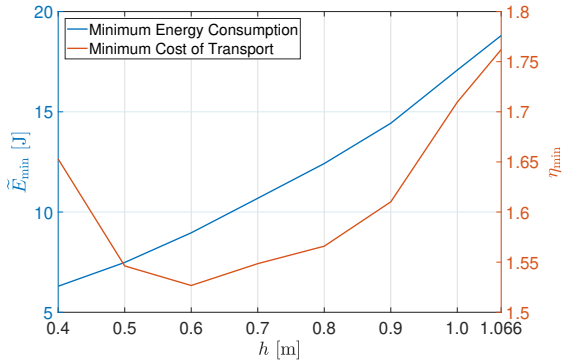

Fig. 8. NLP results of minimum energy consumption for different goal heights. The red line indicates the vertical jumping is most energy-efficient when the goal height h is around 0.6 m.

The blue line shows the optimized minimum energy consumption at each goal height. It is a monotonically increasing function over the given feasible domain, which makes physical sense. Furthermore, the idea of the dimensionless specific mechanical cost of transport [26] is used as a measure of the energy efficiency of the system over one vertical jump, which is defined as

$$\eta = \frac{\widetilde{E}}{(M+2m)g(h-z[1])}, \tag{29}$$

where $h - z[1]$ is the overall vertical distance traveled. The red line in Fig. 8 tells that η is minimized when h is around 0.6 m, i.e., the most energy-efficient way of vertical jumping is when the goal height is set to 0.6 m. We will later apply the idea of η when designing optimum jumping gait for legged robots in terms of energy efficiency.

V. Example of a Two DOF Robotic Leg

This section investigates the optimum jumping gait for a two DOF robotic leg via the NLP formulation. A concept diagram is shown in Fig. 9.

A. Two DOF Robotic Leg Model

The two DOF robotic leg model is almost the same as the single one in Section IV-A but without the two linear guides. In addition, both the femur and tibia joints are now actuated. The equations of motion take the form as (1), where $\boldsymbol{q} = [x, z, \theta_1, \theta_2]^T$, $\boldsymbol{B\tau} = [0, 0, \tau_1, \tau_2]^T$, $\boldsymbol{d} = [d_x, d_z]^T$. The kinematics constraint (4) gives

$$\boldsymbol{h}(\boldsymbol{q}) = \begin{bmatrix} x + l\cos\theta_1 + l\cos(\theta_1+\theta_2) \\ z - l\sin\theta_1 - l\sin(\theta_1+\theta_2) \end{bmatrix} = \begin{bmatrix} p \\ 0 \end{bmatrix}, \tag{30}$$

where the foot position p is now actually arbitrary and is thus set to 0 for convenience. The Jacobian matrix can be computed from (5) as

$$\boldsymbol{J}(\boldsymbol{q}) = \begin{bmatrix} 1 & 0 & -l(\sin\theta_1 + \sin(\theta_1+\theta_2)) & -l\sin(\theta_1+\theta_2) \\ 0 & 1 & -l(\cos\theta_1 + \cos(\theta_1+\theta_2)) & -l\cos(\theta_1+\theta_2) \end{bmatrix}. \tag{31}$$

B. Optimum Jumping Gait

We aim to design an optimum periodic jumping gait for the two DOF robotic leg in terms of energy efficiency via the NLP formulation. The set of decision variables χ is first defined as

$$\chi := \{\boldsymbol{q}[k], \dot{\boldsymbol{q}}[k] \,|\, k = 1, \ldots, N\} \cup \{\boldsymbol{\Lambda}, \boldsymbol{V}, T\}. \tag{32}$$

In spite of the constraints illustrated in Section III-C, some other constraints need to be imposed to ensure periodicity of the jumping gait. For simplicity, after take-off, the body of the robotic leg is assumed to be a projectile without considering the leg dynamics during the flight phase. Therefore, the relationship between the final configuration and the configuration just before touch-down ($\boldsymbol{q}^- = \left[x^-, z^-, \theta_1^-, \theta_2^-\right]^T$ and $\dot{\boldsymbol{q}}^- = \left[\dot{x}^-, \dot{z}^-, \dot{\theta}_1^-, \dot{\theta}_2^-\right]^T$) is given by

$$\dot{x}[N] = \dot{x}^- = v_d, \tag{33}$$

$$\left(\dot{z}^-\right)^2 - \dot{z}[N]^2 = 2g\left(z[N] - z^-\right). \tag{34}$$

(33) indicates a constant desired horizontal velocity $v_d < 0$ throughout the flight phase while (34) determines the change in vertical velocity component based on the height change. By assuming an impulsive and perfectly plastic collision, the touch-down impact model is formulated according to [27] as

$$\boldsymbol{M}(\boldsymbol{q}[1])\left(\dot{\boldsymbol{q}}[1] - \dot{\boldsymbol{q}}^-\right) = \boldsymbol{J}(\boldsymbol{q}[1])^T\boldsymbol{\xi}, \tag{35}$$

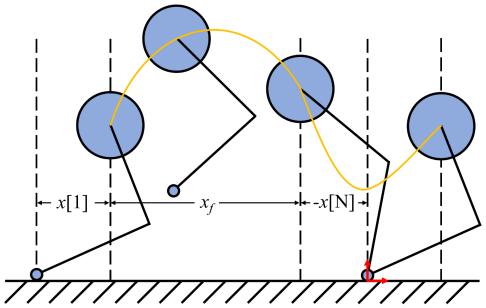

Fig. 9. One complete cycle of the 2 DOF robotic leg jumping gait. It starts from the right after the touch-down impact, pushes the ground, takes off, travels in the air, eventually lands, and completes the cycle. The world frame is in red. The amber line describes the body trajectory.

where the additional decision variable $\boldsymbol{\xi}$ is the impact force from the ground to the foot, the initial generalized velocity $\dot{\boldsymbol{q}}[1]$ represents the velocity right after the impact and the initial position $\boldsymbol{q}[1]$ is assumed to be invariant through the impact, i.e., $\boldsymbol{q}[1] = \boldsymbol{q}^-$. The two links are also assumed to be able to already return to their initial configuration by the end of the flight phase, i.e., $\dot{\theta}_1^- = \dot{\theta}_2^- = 0$.

To optimize the jumping gait in terms of energy efficiency, the idea of cost of transport η is applied yet with modifications to (29), since the problem is now in two dimensions. The total horizontal distance traveled Δx for one jump can be simply measured by the change in foothold on the ground, which is given by

$$\Delta x = x[1] + x_f - x[N], \tag{36}$$

where $x[1] \geq 0$, $x[N] \leq 0$ from Fig. 9, and

$$x_f = -v_d \cdot \frac{\dot{z}[N] - \dot{z}^-}{g}, \tag{37}$$

which is computed based on the projectile assumption of the body during the flight phase. Therefore, η can be defined as

$$\eta = \frac{\widetilde{E}}{(M+2m)g\Delta x}. \tag{38}$$

The jumping height is not further considered because x_f already captures that information, i.e., given v_d, Δx will increase if the height increases and vice versa. The NLP is finally formulated as follows

$$\begin{aligned} \underset{\chi,\,\xi}{\text{minimize}} \quad & \textit{Objective Function} \text{ (38)} \\ \text{subject to} \quad & \textit{Dynamics Constraint} \text{ (14)}, \\ & \textit{Kinematics Constraint} \text{ (15)(16)}, \\ & \textit{Motor Constraint} \text{ (17)}, \\ & \textit{Friction Cone Constraint} \text{ (19)}, \\ & \textit{Take-off Constraint} \text{ (21)(22)}, \\ & \textit{Gait Constraint} \text{ (33)(34)(35)}, \\ & \textit{Other Constraints}. \end{aligned} \tag{39}$$

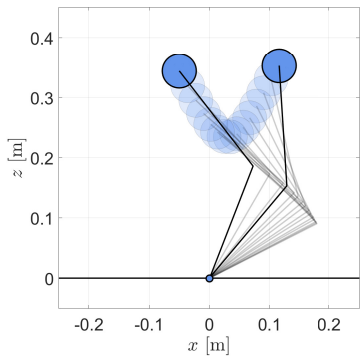

Fig. 10. Visualization for NLP results of optimum jumping gait. The robotic leg starts from the right after the touch-down impact, pushes the ground, travels to the left, and eventually takes off for the next jump.

MATLAB's *fmincon* function is used to solve (39) with $v_d = -1$ m/s, $\mu = 1$, $\beta = 0$, $N = 15$, and $S = 5$, yielding the optimized jumping gait with $\eta = 1.26$, as shown in Fig. 10. The robotic leg starts from the right-hand side right after the touch-down impact, pushes the ground, travels to the left, and eventually takes off for the next jump. MATLAB's *ode45* function is used to verify the NLP results and the simulation results are shown in Fig. 11. During the flight phase, a simple

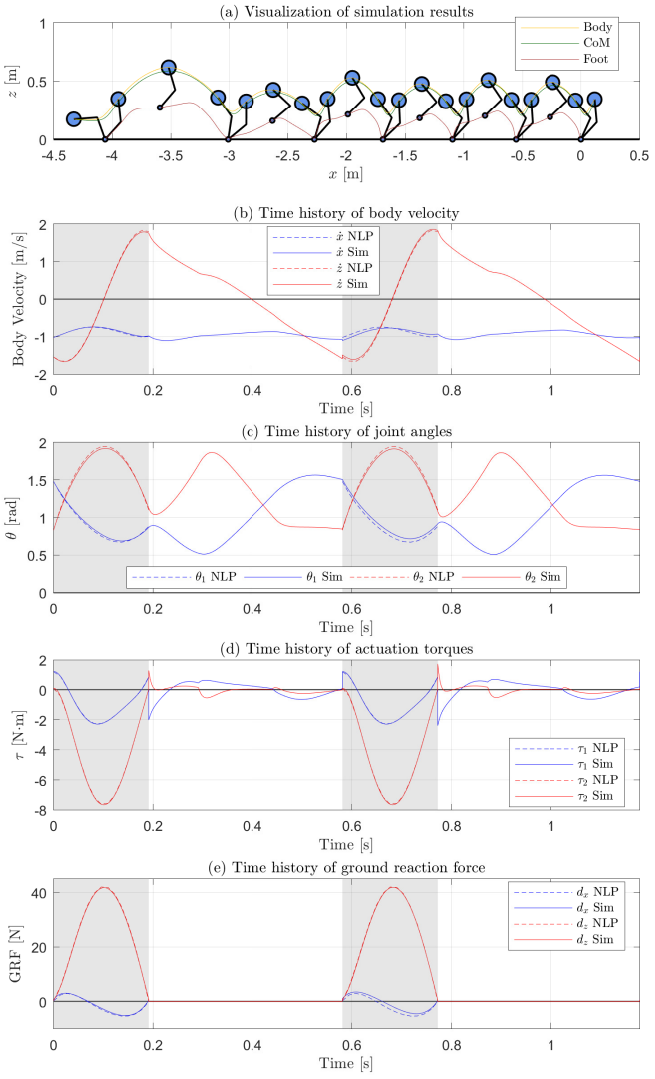

Fig. 11. Simulation results of optimum jumping gait. Figure (a) visualizes the critical moments in the simulation, e.g., touch-down, take-off, midpoint during the flight phase. The amber, green, and brown lines describe the trajectories of the robot body, CoM, and foot, respectively. Starting at the origin, the robotic leg is able to complete four planned jumps to the left before divergence. The rest figures compares the simulation results with the NLP results for the first two jumps. The shaded area represents the stance phase while the white area represents the flight phase. Figure (b) shows the body velocity. The desired horizontal velocity $v_d = -1$ m/s. The simulation results validate the projectile assumption of the body during the flight phase. Figure (c) shows the two joint angles. They are forced to follow a predefined trajectory during the flight phase via a PD controller. Figure (d) shows the actuation torques. The same torque profile is implemented in the simulation as suggested by the NLP results. Figure (e) shows the ground reaction force. The NLP results ensure that no slippage happens with $\mu = 1$.

joint level PD controller is implemented to drive the two links to its desired configuration for the upcoming stance phase. With only open-loop control during the stance phase, the robotic leg is able to complete more or less four planned jumps before divergence, which indicates the proposed NLP formulation is good enough for designing periodic jumping gait for legged robots.

VI. Conclusion

In this paper, a trajectory optimization algorithm specifically served for legged robot jumping applications during the stance phase was presented in detail via a nonlinear programming (NLP) formulation, in consideration of robot full-body dynamics and kinematics, actuator capability, terrain condition, etc. The method is applicable to a wide class of jumping robots and was successfully implemented on an articulated robotic leg as an example. Optimized jumping trajectories were investigated in terms of maximum reachable height, minimum energy consumption, as well as optimum energy efficiency. The simulation and experimental demonstrations verify that this approach is capable of not only optimizing one single jumping trajectory, but also designing a periodic jumping gait for legged robots. In spite of initial guess and local optima issues, the detailed robot model information, i.e., dynamics and kinematics where the nonlinearity usually comes from, is well captured in the NLP. For that reason, the NLP results can be almost directly tested in the simulation and experimental environments with desired outcomes. Our future plan is to study optimum landing strategy for legged robots after take-off.

Acknowledgment

This work is partially supported by ONR through grant N00014-15-1-2064. The authors thank the Robotics and Mechanisms Laboratory (RoMeLa) at UCLA.

References

[1] M. H. Raibert, *Legged Robots That Balance*. MIT Press, 1986.

[2] M. H. Raibert and H. B. Brown, Jr., "Experiments in Balance With a 2D One-Legged Hopping Machine," *Journal of Dynamic Systems, Measurement, and Control*, vol. 106, pp. 75–81, 03 1984.

[3] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, "Bigdog, the rough-terrain quadruped robot," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 10822 – 10825, 2008. 17th IFAC World Congress.

[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*, vol. 40, pp. 429–455, Mar 2016.

[5] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, "The development of honda humanoid robot," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, vol. 2, pp. 1321–1326 vol.2, May 1998.

[6] G. Kenneally, A. De, and D. E. Koditschek, "Design principles for a family of direct-drive legged robots," *IEEE Robotics and Automation Letters*, vol. 1, pp. 900–907, July 2016.

[7] A. Ramezani, J. W. Hurst, K. Akbari Hamed, and J. W. Grizzle, "Performance Analysis and Feedback Control of ATRIAS, A Three-Dimensional Bipedal Robot," *Journal of Dynamic Systems, Measurement, and Control*, vol. 136, 12 2013. 021012.

[8] 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 *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 38–44, Oct 2016.

[9] M. Hutter, C. Gehring, M. A. Höpflinger, M. Blösch, and R. Siegwart, "Toward combining speed, efficiency, versatility, and robustness in an autonomous quadruped," *IEEE Transactions on Robotics*, vol. 30, pp. 1427–1440, Dec 2014.

[10] G. A. Pratt and M. M. Williamson, "Series elastic actuators," in *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, vol. 1, pp. 399–406 vol.1, Aug 1995.

[11] D. W. Robinson, J. E. Pratt, D. J. Paluska, and G. A. Pratt, "Series elastic actuator development for a biomimetic walking robot," in *1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (Cat. No.99TH8399)*, pp. 561–568, Sep. 1999.

[12] 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, pp. 509–522, June 2017.

[13] S. Kalouche, "Design for 3d agility and virtual compliance using proprioceptive force control in dynamic legged robots," Master's thesis, Carnegie Mellon University, Pittsburgh, PA, August 2016.

[14] T. Zhu, J. Hooks, and D. Hong, "Design, modeling, and analysis of a liquid cooled proprioceptive actuator for legged robots," in *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 36–43, July 2019.

[15] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2245–2252, Oct 2018.

[16] J. Yu, J. Hooks, X. Zhang, M. Sung Ahn, and D. Hong, "A proprioceptive, force-controlled, non-anthropomorphic biped for dynamic locomotion," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pp. 1–9, Nov 2018.

[17] S. H. Hyon and T. Mita, "Development of a biologically inspired hopping robot-"kenken"," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 4, pp. 3984–3991 vol.4, May 2002.

[18] K. Arikawa and T. Mita, "Design of multi-dof jumping robot," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 4, pp. 3992–3997 vol.4, May 2002.

[19] M. Hutter, C. D. Remy, M. A. Höpflinger, and R. Siegwart, "Slip running with an articulated robotic leg," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4934–4939, Oct 2010.

[20] B. Lim, J. Babic, and F. C. Park, "Optimal jumps for biarticular legged robots," in *2008 IEEE International Conference on Robotics and Automation*, pp. 226–231, May 2008.

[21] S. Hiasa, R. Sato, A. Ming, F. Meng, H. Liu, X. Fan, X. Chen, Z. Yu, and Q. Huang, "Development of a bipedal robot with bi-articular muscle-tendon complex between hip and knee joint," in *2018 IEEE International Conference on Cyborg and Bionic Systems (CBS)*, pp. 391–396, Oct 2018.

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

[23] Q. Nguyen, M. J. Powell, B. Katz, J. D. Carlo, and S. Kim, "Optimized jumping on the mit cheetah 3 robot," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 7448–7454, May 2019.

[24] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[25] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *SIAM Review*, vol. 59, no. 4, pp. 849–904, 2017.

[26] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.

[27] Y. Hurmuzlu and D. B. Marghitu, "Rigid body collisions of planar kinematic chains with multiple contact points," *The International Journal of Robotics Research*, vol. 13, no. 1, pp. 82–92, 1994.