

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/355205699>

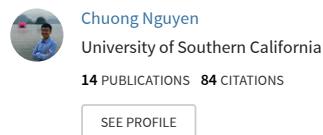
Contact-timing and Trajectory Optimization for 3D Jumping on Quadruped Robots

Preprint · October 2021

CITATIONS
0

READS
155

2 authors, including:



Some of the authors of this publication are also working on these related projects:



Contact-timing and Trajectory Optimization for 3D Jumping on Quadruped Robots

Chuong Nguyen and Quan Nguyen

Abstract— Performing highly agile acrobatic motions with a long flight phase requires perfect timing, high accuracy, and coordination of the full-body motion. To address these challenges, this paper presents a novel timings and trajectory optimization framework for legged robots performing aggressive 3D jumping. In our approach, we firstly utilize an effective optimization framework using simplified rigid body dynamics to solve for contact timings and a reference trajectory of the robot body. The solution of this module is then used to formulate a full-body trajectory optimization based on the full nonlinear dynamics of the robot. This combination allows us to effectively optimize for contact timings while ensuring that the jumping trajectory that can be effectively realized in the robot hardware. We first validate the efficiency of the proposed framework on the A1 robot model for various 3D jumping tasks such as double-backflips off the high altitude of 2m. Experimental validation was then successfully conducted for a variety of aggressive 3D jumping motions such as diagonal jumps, barrel roll and double barrel roll from a box of height 0.4m and 0.9m respectively.

I. INTRODUCTION

In the last decade, there has been a rapid development of legged robots to traversing rough terrain [1],[2],[3],[4],[5]. Among them, the research into jumping behaviors on legged robots has greatly drawn research attentions because of its remarkable advantages to navigate high obstacles [6],[7],[8],[9].

There exists a number of dynamic models of legged robots, which are commonly utilized by optimization and control-based frameworks to perform agile locomotion tasks. One widely used model is the single rigid body (SRB) dynamics that considers the effect of ground reaction forces (GRF) to the robot body while ignoring the leg dynamics ([7],[10],[11]). This simplification enables can be used for real-time feedback control. This model can also be utilized to achieve 3D jumping via Model Predictive Control (MPC) [12]. However, to guarantee real-time computation, the prediction horizon of the MPC controller needs to be small enough, limiting the approach from addressing motions with a long aerial time. Moreover, since the legs' dynamics is neglected, this approach affects the accuracy of jumping motions. While the SRB dynamics model is a suitable choice for online execution, the full-body dynamics (FBD) at one end of the spectrum is utilized when model accuracy comes to crucial. The FBD is utilized in trajectory optimization (TO) framework to accurately plan periodic running gait constrained in 2D planar plane for a 13 degree of freedom

This work is supported by USC Viterbi School of Engineering startup funds.

The authors are with the Department of Aerospace and Mechanical Engineering, University of Southern California, USA: vanchuong.nguyen@usc.edu, quann@usc.edu,

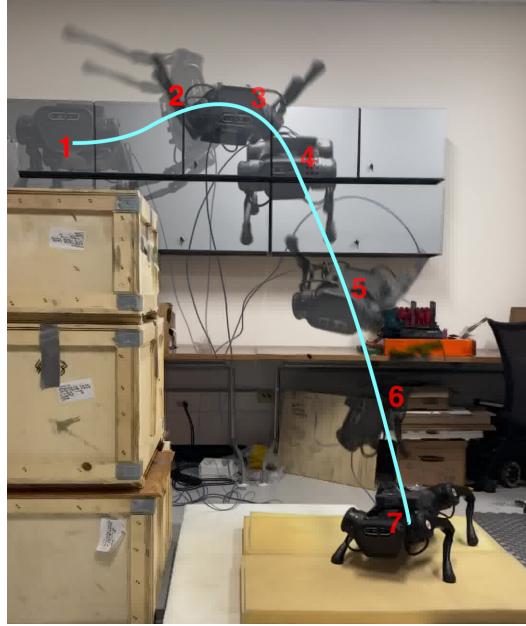


Fig. 1: A combined motion snapshot of the Unitree A1 robot performing a double barrel roll from a 0.9m platform. Experiment and simulation video: <https://youtu.be/d7RcWEXTbqc>.

(DoF) bipedal robot in [13], and for a rearing and posture recovery in place in [14]. The jumping motions usually require long aerial phase and high accuracy of full-body coordination for all contact phases, thus utilizing full-body dynamics is essential to achieve high accuracy motions. Our previous work [15] employs the FBD of 2D quadruped robots with 7 DoF to implement the TO for 2D jumping motions. The work enables jumping up and down from a desk with high accuracy. In this work, we are interested in complex 3D jumping motions and implementing the TO for FBD to accurately plan and execute these highly agile jumps for high DoF robots (e.g. quadruped robots).

Another simplified model, namely centroidal dynamics and full-body kinematic, is also commonly used in TO when planning agile motions for legged robots ([9],[16]). The advantage of this model is that it reduces the complexity of dynamics compared to full-body dynamics, while guaranteeing the feasibility of kinematic motions. Thus, motion planners are still able to achieve some 3D motions while keeping the computational tractability. Regarding the quadruped jumping, a recent work [9] proposes an effective framework that implements TO for this centroidal dynamics, then uses a whole-body controller (WBC) for tracking the reference motions, to perform various jumping tasks in 3D. Difference from [9], we optimize for contact timings and leverage the full nonlinear dynamics of the robot in the

trajectory optimization. This effective combination allows us to achieve highly accuracy jumping motions such that multiple barrel rolls and backflips.

Prior to computing optimal contact forces at a specific time, it is essential to know if the foot is in contact with the environment. This naturally raises the necessary of determining the contact timing of a foot when planning the legged robots motion. While there is an increasing number of work attempt to find optimal contact timings for agile motions ([17],[18],[19],[20]), the contact timings optimization for highly dynamic jumps in 3D space has not been well explored. The recent work [19] proposes a TO approach based on SRB dynamics to solve for optimal contact timings and a sequence of foothold location. However, since this work uses a linearized rigid body dynamics using Euler angle representation, it offers a limited range of achievable motions due to the singularity [21]. In our work, we are interested in optimizing the contact timings for highly aggressive and complex 3D jumps with long aerial time. We utilize rotation matrix to represent SRB orientation to prevent the singularity issue and unwinding issue that are associated with Euler angle and quaternion representation respectively. This allows us to plan general complex jumping motions in 3D while optimizing the contact timings. In addition, difference than [19], the solution of the proposed TO of contact timings is then utilized to formulate the full-body TO that considers torque constraints and the whole-body coordination. Therefore, our framework enables aggressive jumps to be realized in the robot hardware.

The contribution of the work is summarized as follows:

- We propose a TO approach to simultaneously solve for optimal contact timings and reference trajectory of robot body in a fast manner, which will be used for the full-body TO.
- In the TO of contact timings, the rotation matrix is directly utilized to represent the orientation of robot body to avoid singularity and unwinding issues. This allows us plan a wide range of complex 3D jumping motions.
- We present a full-body TO to perform a diverse set of aggressive 3D jumping tasks that require high accuracy and long aerial time.
- Our proposed framework is validated on both experiment and the A1 robot model for various gymnastic 3D jumping tasks, e.g., double-backflips and double barrel roll off boxes of height 2m and 0.9m correspondingly.

The rest of our paper is organized as follows. The TO approaches are presented in Section II, and jumping and landing controller are briefly described in Section III. Results from selected hardware experiments are shown in Section IV. Finally, Section V provides concluding remarks.

II. TRAJECTORY OPTIMIZATION WITH CONTACT TIMING OPTIMIZATION

A. Motivation

When performing highly dynamic and complex jumping in 3D, it is critical to utilize the full rigid-body dynamics

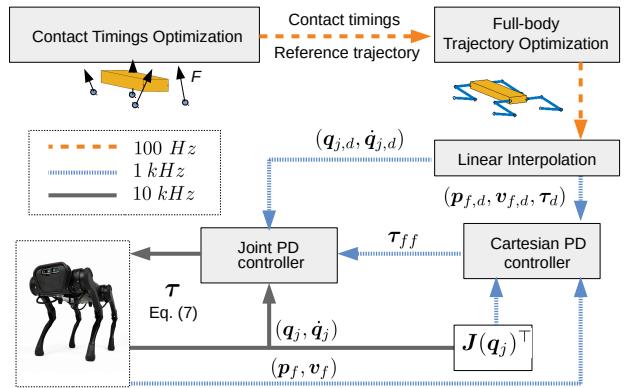


Fig. 2: **Block Diagram for the propose framework.** The timings and TO of SRB dynamics produces the contact timings and reference trajectory at 100 Hz for the TO of whole-body dynamics. Linear interpolation is then used to get the reference profile at 1 kHz for the Cartesian controller. The joint PD controller executes at 10 kHz.

of the robot to maximize the jumping performance as well as guarantee the accuracy of the jumping trajectory while realizing in the real robot hardware. To the best our knowledge, it does not have any work on implementing the full-body TO to achieve 3D quadruped jumping motions.

In the full-body TO set up, contact timings is predefined for each jumping phase. Since jumping motions usually have long aerial phase and there is no control inputs to change the CoM trajectory of the robot during that phase, optimizing the contact timings is important for the entire motion. In addition, the manual selection of the timings is time consuming, and not optimal, and even it is not feasible for the full-body TO to obtain solutions for many complex 3D jumping motions. Therefore, it is crucially important to implement an approach to automatically compute optimal timing. In addition, timings in highly dynamic acrobatic motion with long flight phase plays a crucial role in minimizing the effort or energy, guaranteeing the feasibility of the motion within the limit of actuator powers.

In the following, we introduce a framework this includes optimal contact timings TO and full-body TO to generate a general and complex 3D jumping motions at high accuracy.

B. Contact timings optimization

Directly implementing the contact timings optimization on the full-body optimization framework takes greatly considerable time to solve due to the highly complexity of the problem. In our implementation, it does not even produce a feasible solution for many complex 3D jumps. Therefore, these issues motivates us to take advantage of the SRB to optimize for contact timing. Section II-B will present our contact-timing optimization framework to obtain the optimal contact timings and body reference trajectory, which then be used to formulate the full-body TO to achieve tractable solving time in Section II-C. The overview of our approach and control diagram is illustrated in Fig. 2.

Unlike prior works [9],[19] that use Euler angle representation, we utilize rotation matrix instead to represent

TABLE I: A1 Robot Parameters

Parameter	Symbol	Value	Units
Gear Ratio	gr	9	
Max Torque	τ_{max}	33.5	Nm
Max Joint Speed	\dot{q}_{max}	21	rad/s
Total leg mass	m_l	4.71	kg
Total robot mass	m	12	kg
Trunk dimension	l, w, h	0.361, 0.194, 0.114	m
Trunk Inertia	I_{xx}, I_{yy}, I_{zz}	0.017, 0.056, 0.065	kg.m^2
Hip Link Lengths	l_1	0.083	m
Leg Link Lengths	l_2, l_3	0.2	m

the body orientation when planning 3D jumping motions. Our utilization prevents the singularity issue associated with Euler angles, and also avoids the unwinding issue related with quaternions representation [21]. Thus, our framework allows to generate any 3D jumping motions.

Given the sequence of contacts, we will optimize their duration (i.e. contact timings). Here we choose Cartesian space to derive the SRB's equation of motion:

$$\ddot{\mathbf{p}} = \sum_{s=1}^{n_s} \mathbf{f}^s / m - \mathbf{g}, \quad (1a)$$

$$\mathbf{I}_b \dot{\Omega} + \Omega \times \mathbf{I}_b \Omega = \mathbf{R}^{-1} \sum_{s=1}^{n_s} \mathbf{f}^s \times (\mathbf{p} - \mathbf{p}_f^s), \quad (1b)$$

$$\dot{\mathbf{R}} = \mathbf{R} \hat{\Omega} \quad (1c)$$

where n_s is number of feet, m is the robot's mass, $\Omega \in \mathbb{R}^3$ is angular velocity expressed in the body frame, \mathbf{R} is rotation matrix of the body frame, \mathbf{g} is the gravity acceleration; $\mathbf{p}, \dot{\mathbf{p}}, \ddot{\mathbf{p}} \in \mathbb{R}^3$ is the CoM position, velocity, acceleration of the body in the world frame; $\mathbf{f}^s \in \mathbb{R}^3$ is GRF on foot s^{th} ; $\mathbf{p}_f^s \in \mathbb{R}^3$ is s^{th} foot position in the world frame. The hat mat $(\cdot) : \mathbb{R}^3 \rightarrow so(3)$ converts any vector in \mathbb{R}^3 to the space of skew-symmetric matrices. For the sake of notation, we define the robot's state as $\mathbf{x} := [\mathbf{p}; \dot{\mathbf{p}}; \Omega; \dot{\Omega}; \mathbf{R}]$. The contact timing and trajectory optimization are then formulated as follows:

$$\text{minimize } \sum_{k=1}^{N^c} \epsilon_\Omega \Omega_k^T \Omega_k + \epsilon_f f_k^T f_k + \epsilon_R e_{R_k}^T e_{R_k}$$

$$\text{s.t. } [\mathbf{R}, \mathbf{p}, \mathbf{p}_f^s](k=1) = [\mathbf{R}_0, \mathbf{p}_0, \mathbf{p}_{f,0}^s], \text{ initial states} \quad (2a)$$

$$[\mathbf{R}, \mathbf{p}, \mathbf{p}_f^s](k=N^c) = [\mathbf{R}_g, \mathbf{p}_g, \mathbf{p}_{f,g}^s], \text{ final states} \quad (2b)$$

$$|\mathbf{R}(k)[\mathbf{p}_f^s(k) - \mathbf{p}(k)] - \bar{\mathbf{p}}_f^s(k)| \leq \mathbf{r}, \text{ foot position} \quad (2c)$$

$$\beta(\mathbf{x}_k, \mathbf{f}_k) \leq 0, \quad (2d)$$

$$\gamma(\mathbf{x}_k, \mathbf{x}_{k+1}, \mathbf{f}(k), \mathbf{p}_f^s(k)) = 0, \text{ discrete dynamic} \quad (2e)$$

$$\mathbf{R}_{k+1} = \mathbf{R}_k \exp(\widehat{\Omega_k T_i / N_i}), \quad (2f)$$

$$\sum_{i=1}^n T_i \in [T_{min}, T_{max}], \quad N^c = \sum_{i=1}^n N_i \quad (2g)$$

$$\text{for } k = 1, 2, \dots, N^c$$

where $\Omega_k \in \mathbb{R}^3$, $f_k \in \mathbb{R}^{12}$ is angular velocity of SRB w.r.t the body frame, and GRF on four legs at the iteration k^{th} ; $\epsilon_\Omega, \epsilon_f, \epsilon_R$ are cost function weights of corresponding

elements, and $\epsilon_R, \epsilon_\Omega \gg \epsilon_f$. We use error term of rotation matrix as $e_{R_k} = \log(\mathbf{R}_{ref,k}^T \mathbf{R}_k)^\vee$, where $\log(\cdot) : SO(3) \rightarrow so(3)$ is the logarithm map, and the vee map $(\cdot)^\vee : so(3) \rightarrow \mathbb{R}^3$ is the inverse of hat map [21],[22]. With given final and initial rotation matrix, \mathbf{R}_g and \mathbf{R}_0 respectively, we utilize a linear interpolation to obtain $\mathbf{R}_{ref,k}$ at k^{th} step. The equation (2c) implies that the s^{th} foot position is constrained inside a sphere \mathcal{S}_s of radius \mathbf{r} so that the joint angle are within limits. The center of the sphere $\bar{\mathbf{p}}_f^s(k)$ is relative to the CoM position.

The function $\beta(\cdot)$ captures various constraints on CoM, friction cone limits, GRF, and geometric constraints related to the ground and obstacle clearance. The function $\gamma(\cdot)$ captures the dynamic constraints discretized from (1a)-(1b).

In (2g), n is a number of contact phases. For example, if the pre-selected contact schedule is four-leg contact, rear-leg contact, and flight phase, then $n = 3$. N_i is the predefined number of time steps for the i^{th} contact phase, $i \in \{1, 2, \dots, n\}$. Note that we optimize the timing T_i for each period and the total time $T_{opt} = \sum_{i=1}^n T_i$ given predefined interval $[T_{min}, T_{max}]$. The equation (2f) is derived from (1c) to ensure \mathbf{R}_k evolves in the $SO(3)$ manifold. Here, $\exp(\cdot) : so(3) \rightarrow SO(3)$ is the matrix exponential map. The algorithm will check a k^{th} iteration belongs to which i^{th} predefined contact phase to choose T_i/N_i accordingly.

The term $f_k^T f_k$, which is intuitively represented for generated power ([9],[12]), is added to the cost function of (2). By utilizing that term, we are able to minimize necessary effort or energy of robot when perform jumping tasks.

Remark 1: Despite the advantage of using rotation matrix \mathbf{R} to achieve the most 3D jumping motions, utilizing it in the TO set up introduces more optimization variables and constraints. This significantly increases the problem size and solving time. To achieve a feasible solution, \mathbf{R} in (2d) is required to be approximated at high accuracy enough to satisfy the $SO(3)$ property at every time step, and Taylor series approximation at a high degree is utilized for that purpose. However, choosing a really high degrees of Taylor's approximation is prohibitively costly in term of computational time. Therefore, in order to balance between accuracy and solving time, we utilize the 4th degree of Taylor's series:

$$\exp(\mathbf{A}) = \sum_{k=0}^4 \frac{\mathbf{A}^k}{k!}. \quad (3)$$

Moreover, for complex jumping motions in 3D, adding and minimizing the term $e_{R_k}^T e_{R_k}$ in the cost function in (2) helps to guide the TO toward a feasible solution in a fast manner.

Remark 2: Our proposed TO for contact timing also gives advantage that it is general to any legged robot with arbitrary numbers of legs. This can efficiently generate any 3D jump motion regarding obstacles with long aerial time.

C. Full-body trajectory optimization for 3D jumping

When performing highly dynamic jumping, it is important to consider the full nonlinear dynamics of the robot in the

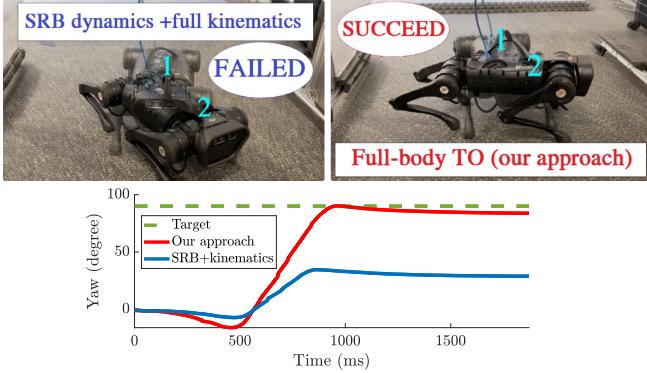


Fig. 3: Comparison between TO of SRB dynamics+ full-body kinematics constraints approach and full-body TO (our approach) for 90° spinning jump target in experiment.

TABLE II: Optimal contact timings, solving time via IPOPT for the contact timings and TO of SRB in Section II-B and full-body TO in Section II-C

Jumping tasks	Optimal contact timings	Solving time contact timing	Solving time full-body TO
lateral jump 30cm	50, 28	4.14 [s]	31 [s]
lateral jump down	52, 35	5.02 [s]	32 [s]
90° spinning jump	56, 31	5.7 [s]	39 [s]
diagonal jump	54, 30, 33	5.98 [s]	226 [s]
barrel roll	51, 32, 35	6.61 [s]	35 [s]
double barrel roll	52, 34, 55	11.7 [s]	46 [s]
double backflip	50, 33, 69	7.23 [s]	89 [s]

optimization framework. This will guarantee the accuracy of the jumping trajectory while transferring to the hardware.

1) *Full-body dynamics model*: The robot is modeled as a rigid-body system consisting of n bodies, and spatial vector algebra [23] is used to construct the robot's equations of motion:

$$\begin{bmatrix} \mathbf{H} & -\mathbf{J}_s^T \\ -\mathbf{J}_s^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \mathbf{F}_s \end{bmatrix} = \begin{bmatrix} -\mathbf{C}\dot{\mathbf{q}} - \mathbf{g} + \mathbf{B}\boldsymbol{\tau} + \mathbf{B}_f\boldsymbol{\tau}_f \\ \mathbf{J}_s(\mathbf{q})\dot{\mathbf{q}} \end{bmatrix} \quad (4)$$

where $\mathbf{q} := [x; y; z; q_{roll}; q_{pitch}; q_{yaw}; \mathbf{q}_j]$ is a vector of generalized coordinates, in which $x, y, z, q_{roll}, q_{pitch}, q_{yaw}$ are the COM position and body's rotational angles respectively, and $\mathbf{q}_j \in \mathbb{R}^{12}$ is a vector of joint angles. The mass matrix is denoted by \mathbf{H} ; the matrix \mathbf{C} is represented for Coriolis and centrifugal terms; \mathbf{g} is the gravity vector; \mathbf{J}_s is the spatial Jacobian of the body containing the s^{th} contact foot, expressed at the foot and in the world coordinate system; \mathbf{B} and \mathbf{B}_{fric} are distribution matrices of actuator torques $\boldsymbol{\tau}$ and the joint friction torques $\boldsymbol{\tau}_{fric}$; \mathbf{F}_s is the spatial force at the s^{th} contact foot.

2) *Cost function and constraints*: The ultimate goal of our framework is to find a feasible jumping motion for each 3D jumping task with the full rigid-body dynamics consideration. Due to the high complexity of this problem, the feasibility domain is limited. Therefore, the purpose of

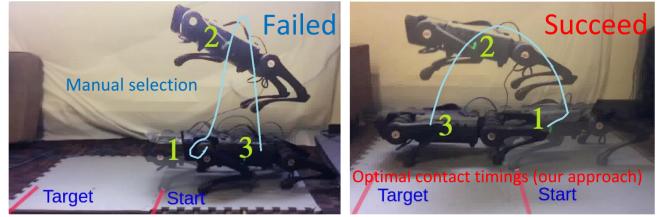


Fig. 4: Motion snapshots from jumping forward with manual selection of contact timings in the left figure and optimal contact timings (our approach) in the right figure in experiment.

this cost function is to guide the optimization to converge to a feasible solution where the robot's coordinates stays close to the reference configuration \mathbf{q}_{ref} if possible. The CoM position and body orientation obtained from Section II-B are also linearly interpolated to get their profiles sampling at $dt = 10 ms$, which is then used as reference for the full-body TO here. The cost function of the TO is defined as follows:

$$J = \sum_{k=1}^{N-1} \epsilon_q (\mathbf{q}_k - \mathbf{q}_{ref,k})^T (\mathbf{q}_k - \mathbf{q}_{ref,k}) + \epsilon_\tau \boldsymbol{\tau}_k^T \boldsymbol{\tau}_k + \epsilon_N (\mathbf{q}_N - \mathbf{q}_N^d)^T (\mathbf{q}_N - \mathbf{q}_N^d), \quad (5)$$

where N denotes the total of the time steps (i.e. $N = T_{opt}/dt$ with T_{opt} is optimal total time obtained from Section II-B; $\mathbf{q}_k, \boldsymbol{\tau}_k$ are the generalized coordinates and joint torque at the iteration k^{th} ; \mathbf{q}_N is the the generalized coordinates at the end of the trajectory; the first six elements of $\mathbf{q}_{ref,k}$ is the reference CoM position and body orientation obtained from contact timing optimization in Section II-B. The last 12 elements of $\mathbf{q}_{ref,k}$ is set to be the final joint configuration. We also use $\mathbf{q}_{ref,k}$ as initial guess for the TO to reduce the solving time. $\epsilon_q, \epsilon_\tau, \epsilon_N$ are cost function weights of corresponding elements. The following constraints are enforced in our trajectory optimization:

- Full-body dynamics constraints (4)
- Initial configuration: $\mathbf{q}(k=0) = \mathbf{q}_0, \dot{\mathbf{q}}(k=0) = \mathbf{0}$.
- Pre-landing configuration: $\mathbf{q}_{j,k} = \mathbf{q}_{j,N}^d, \dot{\mathbf{q}}_{j,k} = \mathbf{0}$ ($k \in [(N-a) : N]$), where a is a jumping task-based selection ($a < N$).
- Final configuration: $\mathbf{q}_k(k=N) = \mathbf{q}_N^d$.
- Joint angle constraints: $\mathbf{q}_{j,min} \leq \mathbf{q}_{j,k} \leq \mathbf{q}_{j,max}$
- Joint velocity constraints: $|\dot{\mathbf{q}}_{j,k}| \leq \dot{\mathbf{q}}_{j,max}$
- Joint torque constraints: $|\boldsymbol{\tau}_k| \leq \boldsymbol{\tau}_{max}$
- Friction cone limits: $|\mathbf{F}_k^x/\mathbf{F}_k^z| \leq \mu, |\mathbf{F}_k^y/\mathbf{F}_k^z| \leq \mu$
- Minimum GRF: $\mathbf{F}_k^z \geq \mathbf{F}_{min}^z$
- Geometric constraints to guarantee: (i) each robot part does not collide with others, (ii) the whole robot body and legs have a good clearance with obstacle and ground.

III. JUMPING AND LANDING CONTROLLER

Having introduced the full-body TO approach, in this section we present a jumping and landing controller to help robot tracking the reference trajectory and effectively

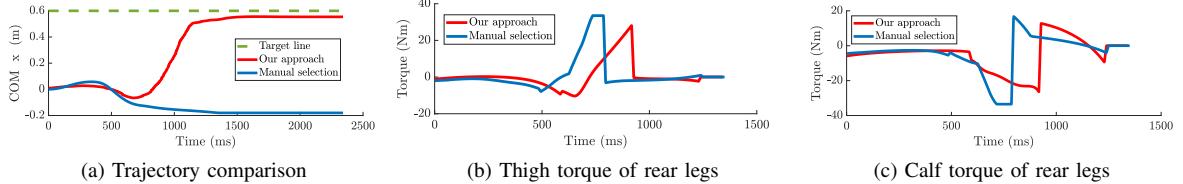


Fig. 5: **Comparison between manual selection and optimal contact timings (our approach)** The comparison is considered based on target achievement from experiments in Fig. 5a, and the feed forward torque of rear legs generated from the full-body TO for jumping forward 0.6 m under manual selection of contact timings and optimal contact timing (our approach) in Fig. 5b & 5c. The Fig. 5 is accompanied with the Fig. 4.

handle the high impact with the ground when performing 3D aggressive jumps off from high altitude (see Fig.2). The desired joint angle q_d , joint velocity \dot{q}_d , foot position $p_{f,d}$ and foot velocity $v_{f,d}$ w.r.t their hips, and feed-forward joint torque τ_d are obtained from the full-body TO. They are then linearly interpolated to get new reference profiles at 1 kHz. To track the reference trajectories, we use the feedback Cartesian PD controller that executes at 1 kHz:

$$\tau_{ff} = J(q_j)^\top [K_p(p_{f,d} - p_d) + K_d(v_{f,d} - v_f)] + \tau_d$$

where $J(q_j)$ is the foot Jacobian at the configuration q_j ; K_p and K_d are diagonal matrices of proportional and derivative gains. The joint PD controller running at 10 kHz in the low-level motor control is integrated to improve the tracking performance. The full controller for tracking desired trajectories is:

$$\tau = \tau_{ff} + K_{p,joint}(q_{j,d} - q) + K_{d,joint}(\dot{q}_{j,d} - \dot{q}). \quad (6)$$

Since there always exists a model mismatch between the optimization and hardware, it normally has orientation angle errors upon landing. Therefore, we utilize a real-time landing controller to handle impact, control GRF, balance the whole body motion during the landing phase, and recover the robot from unexpected landing configurations. For that controller, we extend the proposed QP controller presented in [15] for the 3D jumping motions.

```

if (any  $c_s >= \delta$ )&(t  $\geq T_{posing}$ )
then switch to landing controller,

```

where c_s is the value obtained from the contact sensor of the foot s^{th} , and δ is the force threshold to determine if the ground impact happened. T_{posing} is the instant at the beginning of the pre-landing configuration (after that instant the desired joint velocity $\dot{q}_j^d = 0$). When the impact is detected in any foot, we switch from jumping controller to landing controller, then based on which foots are in contact, different robot models are used for the landing controller. Our experimental results validate that it is effective to use the QP landing controller for SRB to handle impact with the ground and balance the robot when landing.

Remark 3: Normally, all legs are not in ground simultaneously due to the mismatch between the optimization model and hardware. Hence, utilizing a controller for swing legs also plays a crucial role here. Based on the contact model

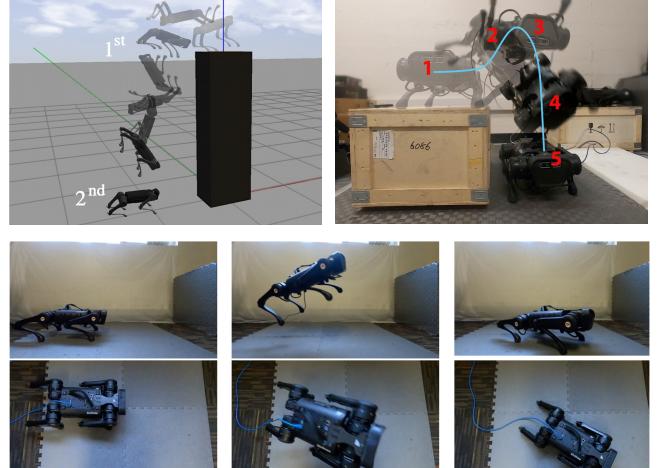


Fig. 6: **Other example of successful 3D jumping motions achieved via our proposed framework.** The top left figure is a 3D double backflip from a box of height 2m. The top right is a 3D barrel roll from a box of height 0.4m. The six figures at the bottom are motion snapshots from a successful 3D diagonal jump $[x, y, yaw] = [0.4 \text{ m}, -0.3 \text{ m}, 45^\circ]$ from a side and a top view in experiments.

derived from the contact detection, the swing legs during landing phase are set at zero normal force and kept at the pre-landing configuration using PD controller until the ground contact is detected on these legs to prevent excessive impact force and unnecessarily extended movement.

Remark 4: For highly aggressive jumping motions, e.g. double barrel roll and double backflips, due to the high linear and angular velocity of robot body upon the ground impact, the robot continues to move and rotate in the current directions. This may cause unfavorable landing posture, which is observed in our experiments. To tackle this issue, we utilize PD controller during the pre-landing configuration to extend the legs in that directions based on the body velocity reference obtained from TO at the impact.

IV. RESULTS

This section presents results from experimental testing and simulation with the A1 robot. A video of the results is included as supplementary material.

A. Numerical Simulation

We use MATLAB and CasADI (see [24]) to construct and solve all presented TO approaches for all 3D jumping

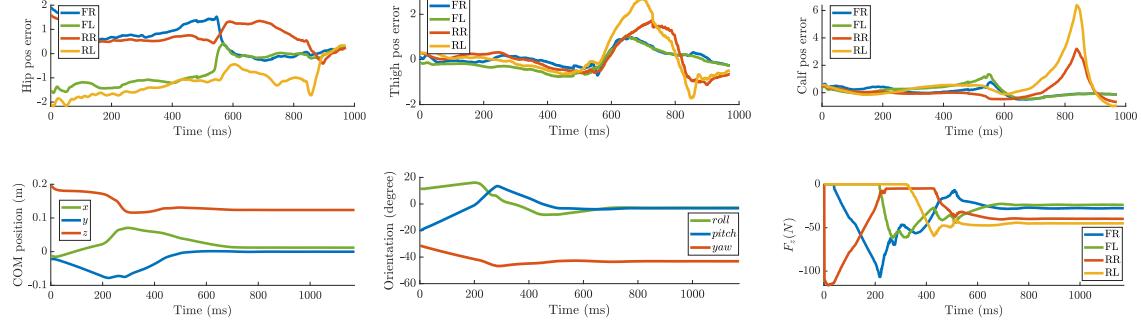


Fig. 7: **3D diagonal jump in experiments.** The upper figures are the tracking error of joint positions (degree) with jumping controller, and the lower figures are the COM position, body orientation, and force command from landing controller during the landing phase, when the robot performs a diagonal jump (see Fig. 6). FR, FL, RR and RL are used to denote front right, front leg, rear right and rear left leg respectively.

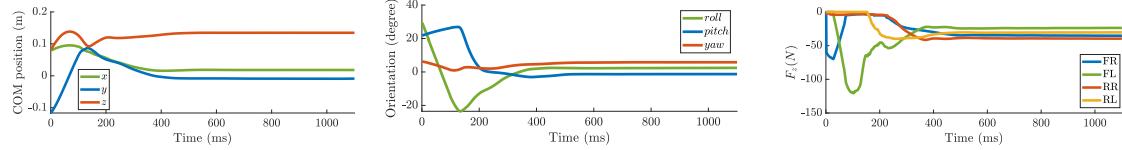


Fig. 8: **Landing controller for a barrel jump in experiments** This plots show the COM position, body orientation, and force command during the landing phase when the robot performs a barrel roll from a box with the height of 0.4 m (see Fig. 6).

behaviors. It is flexible to predefine the contact sequence according to the user’s preference. For example, for the lateral and spinning jumps, we use four-leg contact and flight schedule, while we utilize the four-leg contact, two-leg contact, and flight schedule for the other jumps. As shown in Table II, the contact timing optimization is solved in a fast manner for all 3D jumps, and the full-body TO is solved at tractable time given the high complexity of the full-body dynamics model. In our framework, all 3D jumping tasks are successfully conducted with A1 robot model in ROS before transferred to hardware.

B. Experimental Validation

We validate our proposed framework on the commercial Unitree’s A1 robot [25]. It has high torque density electric motors with single-stage 9:1 planetary gear reduction, and uses these actuators for all the hip, thigh, and knee joints to enable full 3D control of GRF. A pressure-based contact sensor is equipped on each foot. The A1 legs feature a large range of motion: the hip joints have a range of motion of $\pm 46^\circ$, the thigh joints have a range of motion from -60° to 240° and the knee joints have a range from -154.5° to -52.5° . The hip and knee designs allow the robot to operate identically forward, backward and flipped upside-down. The lower link is driven by a bar linkage which passes through the upper link. The robot parameters and its actuation capabilities are summarized in Table I. In the next sections, we present the development of the framework.

Firstly, we implement the TO of SRB dynamics+full-body kinematic constraints to make comparisons with our proposed full-body TO for various 3D jumps. The TO of SRB dynamics+full-body kinematics constraints is constructed based on (2), but we use joint space instead of Cartesian

space to enforce related kinematic constraints and the joint torque is not an optimization variable that finds similar to the TO approach presented in [9]. For instance, we pick a spinning jump 90° to discuss here. As we can see in Fig. 3, while our approach using the full-body TO guarantees a highly accurate jump, the TO framework with SRB and kinematics constraints fails to achieve the target angle. The reason for the failure is that it neglects the legs’ dynamics that is crucial to be considered for high accuracy jumping.

Secondly, we make a comparison between optimal contact timings and manually selected contact timings. We pick the experiment with jumping forward 0.6 m with the contact sequence of four-leg contact, rear-leg contact and flight to discuss here. Their results are illustrated in Fig. 4 and Fig. 5. If contact timings are manually selected with unnecessary long flight time (e.g., 550 ms), this makes the motor’s torque saturated in 100 ms that is up to 1/3 of rear-leg contact phase. This seriously affects the motor working condition, causing failed joint tracking performance. Thus, the robot is unable to reach the target. On the other hand, selecting too small flight time makes the optimization unsolvable since the robot does not have enough power to jump to the desired configuration. By using optimal contact timings, we are able to minimize the effort or energy, prevent the torque saturation issue and guarantees the successful jump on the robot hardware.

Finally, we present the results of different aggressive 3D jumping experiments on the A1 robot that combine optimal contact timings and full-body TO. To the best of our knowledge, the herein results is the first implementation of TO for full-dynamics of quadruped robots for the jumping motions in 3D. We pick some 3D jumping tasks to discuss

here. In a successful diagonal jump (see Fig. 6), the jumping controller guarantees the high tracking performance and the landing controller is able to handle the impact with the ground as evident in Fig. 7. Using our approach, the robot can successfully performs a barrel roll from a box of height 0.4 m in experiments (see Fig. 6). Especially, with our proposed framework, the robot is able to complete a highly aggressive gymnastics double barrel roll jump, which is the first jump ever achieved by the quadrupedal robots (see Fig.1). These results illustrates the efficiency of our approach in optimizing the contact timings for the jumping task as well as the accuracy of the optimization framework while transferring the result to the real hardware. In addition, this also validates the effectiveness of the landing controller. In the barrel roll experiment, since the robot rotates with high angular acceleration from the high altitude, it has a considerably hard impact with the ground. As illustrated in Fig. 8, the landing controller computes reasonable force command to robustly recover the whole body position and orientation under hard impact and error of landing configuration. Fig. 8 shows trajectories of the body position and orientation along with force command during the landing phase.

V. CONCLUSIONS

This paper has introduced the framework for performing highly dynamics 3D jumps with long aerial time on quadruped robots that require model accuracy, perfect timings and coordination of the whole body motion. This framework combines contact timings of SRB dynamics, full-body TO, the jumping controller and robust landing controller. The efficiency of the framework is validated via both A1 robot model and experiments on performing these aggressive tasks. The vision to autonomously detect obstacles will be integrated with the proposed framework in our future work.

ACKNOWLEDGMENTS

The authors would like to thank Yiyu Chen and Hiep Hoang at Dynamic Robotics and Control lab for their assistance in simulation and hardware experimentation. We also thank Dr. Roy Featherstone for the insightful discussion about Spatial v2 for rigid body dynamics.

REFERENCES

- [1] Q. Nguyen, A. Hereid, J. W. Grizzle, A. D. Ames, and K. Sreenath, “3d dynamic walking on stepping stones with control barrier functions,” in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 827–834, IEEE, 2016.
- [2] Q. Nguyen, A. Agrawal, X. Da, W. C. Martin, H. Geyer, J. W. Grizzle, and K. Sreenath, “Dynamic walking on randomly-varying discrete terrain with one-step preview,” *Robotics: Science and Systems*, vol. 2, no. 3, pp. 384–399, 2017.
- [3] Q. Nguyen, X. Da, J. W. Grizzle, and K. Sreenath, “Dynamic walking on stepping stones with gait library and control barrier functions,” *Algorithmic Foundations of Robotics XII*, pp. 384–399, 2020.
- [4] Y. Gong, R. Hartley, X. Da, A. Hereid, O. Harib, J.-K. Huang, and J. Grizzle, “Feedback control of a cassie bipedal robot: Walking, standing, and riding a segway,” in *2019 American Control Conference (ACC)*, pp. 4559–4566, July 2019.
- [5] M. Raibert, K. Blankespoor, G. Nelson, R. Playter, and the Big-Dog Team, “Bigdog, the rough-terrain quadruped robot,” in *Proceedings of the 17th World Congress*, pp. 10822–10825, 2008.
- [6] Y. Ding and H.-W. 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), Vancouver, BC, Canada*, IEEE, 2017.
- [7] H.-W. Park, P. M. Wensing, and S. Kim, “Jumping over obstacles with mit cheetah 2,” *Robotics and Autonomous Systems*, vol. 136, p. 103703, 2021.
- [8] 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.
- [9] M. Chignoli, “Trajectory optimization for dynamic aerial motions of legged robots,” *MIT Libraries*, 2021.
- [10] J. D. Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, “Dynamic locomotion in the mit cheetah 3 through convex model-predictive control,” in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 1–9, Oct 2018.
- [11] D. Kim, J. D. Carlo, B. Katz, G. Bledt, and S. Kim, “Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control,” in *arXiv preprint arXiv:1909.06586*, Sep 2019.
- [12] Y. Ding, A. Pandala, and H.-W. Park, “Real-time model predictive control for versatile dynamic motions in quadrupedal robots,” in *2019 International Conference on Robotics and Automation (ICRA), Montreal, Canada*, pp. 8484–8490, May 2019.
- [13] M. Posa, C. Cantu, , and R. Tedrake, “A direct method for trajectory optimization of rigid bodies through contact,” *The International Journal of Robotics Research*, vol. 33, pp. 69–81, 2014.
- [14] A. Radulescu, I. Havoutis, D. G. Caldwell, and C. Semini, “Whole-body trajectory optimization for non-periodic dynamic motions on quadrupedal systems,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5302–5307, IEEE, 2017.
- [15] 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), Canada*, pp. 7448–7454, IEEE, 2019.
- [16] H. Dai, A. Valenzuela, and R. Tedrake, “Whole-body motion planning with centroidal dynamics and full kinematics,” in *IEEE-RAS International Conference on Humanoid Robots, (Madrid, Spain)*, pp. 295–302, IEEE, 2014.
- [17] I. Mordatch, E. Todorov, and Z. Popović, “Discovery of complex behaviors through contact-invariant optimization,” *ACM Transactions on Graphics*, vol. 31, pp. 1–8, 2012.
- [18] M. Neunert, F. Farshidian, A. W. Winkler, and J. Buchli, “Trajectory optimization through contacts and automatic gait discovery for quadrupeds,” *IEEE Robotics and Automation Letters (RA-L)*, vol. 2, p. 1502–1509, 2017.
- [19] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, “Gait and trajectory optimization for legged systems through phase-based end-effector parameterization,” *IEEE Robotics and Automation Letters*, vol. 3, pp. 1560–1567, 2018.
- [20] B. Ponton, A. Herzog, S. Schaal, and L. Righetti, “A convex model of momentum dynamics for multi-contact motion generation,” in *IEEE-RAS International Conference on Humanoid Robots*, p. 842–849, IEEE, 2016.
- [21] T. Lee, M. Leok, and N. H. McClamroch, “Geometric tracking control of a quadrotor uav on $se(3)$,” in *49th IEEE Conference on Decision and Control, Atlanta, GA, USA*, pp. 5420–5425, Dec 2010.
- [22] F. Bullo and A. D. Lewis, “Geometric control of mechanical systems: modeling, analysis, and design for simple mechanical control systems,” *Springer Science & Business Media*, vol. 49, no. 1, 2004.
- [23] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.
- [24] 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.
- [25] “Unitree a1 robot,” <https://www.unitree.com/products/a1/>.