2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Optimization Based Dynamic Skateboarding of Quadrupedal Robot

Zhe Xu[1], Mohamed Al-Khulaqui[1], Hanxin Ma[1], Jiajun Wang[1], Quanbin Xin[1], Yangwei You[1], Mingliang Zhou[1], Diyun Xiang[1*], Shiwu Zhang[2]

***Abstract*— Robot skateboarding is a novel and challenging task for legged robots. Accurately modeling the dynamics of dual floating bases and developing effective planning and control methods present significant complexities in accomplishing skateboarding behavior. This paper focuses on enabling the quadrupedal platform *CyberDog2* to achieve dynamic balancing and acceleration on a skateboard. An optimization-based control pipeline is developed through careful derivation of the system's equations of motion, considering both the robot and skateboard dynamics. By accounting for system physical constraints, an advanced offline trajectory optimization method is employed to generate various acceleration trajectories, creating a motion library for the system. An online linear model predictive control with whole body control framework is used to track the generated trajectories and stablize the system in real-time. To validate its effectiveness , we conducted experiments in various scenarios. The quadrupedal robot successfully performed acceleration from a static state to various velocities and demonstrated the ability to balance and steer the skateboard.**

Fig. 1: The quadrupedal robot *CyberDog2* skateboarding with four legs.

I. Introduction

Legged robots have made significant advancements in working in real-world environments, with notable examples including Atlas [1] and Spot [2] from Boston Dynamics, ANYmal from ETH [3], and MIT Cheetah [4]. However, these legged robots still have limitations when it comes to versatile locomotion skills. Human and animal locomotion behaviors with sports equipment, such as skiing, skateboarding, and roller-skating, are challenging for legged robots due to hardware and software limitations. Controlling the body and limbs within physical constraints, as well as interacting with dynamic equipment, present additional challenges. The lack of versatile and agile mobility in commercial legged robot products restricts their applications. Therefore, there is a demand for improved control algorithms to enable legged robots to effectively utilize sports equipment.

A. Related work

Recently, there have been numerous remarkable advancements in the field of legged robots, particularly in roller-skating. These developments encompass both quadruped and humanoid robots. The Marco Hutter group has conducted influential research on roller-skating using the renowned ANYmal robot [3]. They have achieved roller-skating and skiing motions using the same controller based on a virtual model [5]. Furthermore, Marko Bjelonic et al., also from the Marco Hutter group, equipped a new version of the ANYmal robot [6] with active wheels, resulting in impressive locomotion skills such as ducking under a table, fast spinning [7], and standing on its rear legs [8]. Thanks to online optimization [9] and offline motion libraries [10], the ANYmal robot with wheels has exhibited exceptional performance in the DAPRA subterranean challenge [11]. In the humanoid robot community, there have been notable studies as well [12], [13]. Seong-Ho Jo et al. conducted pioneering research on humanoid roller-skating, focusing on the application of a low-dimensional approximated model to achieve static motion [14]. Additionally, Koji Ishihara et al. developed a hierarchical model predictive control (MPC) and successfully demonstrated roller-skating using a humanoid[15]. Although these works have achieved outstanding performances in roller-skating, their controllers may not be suitable for riding skateboards due to the unfixed nature of the feet and skateboard.

Studies on robot skateboarding have been conducted by Noriaki Takasugi et al. using humanoid robots JAXON [16] and HRP2 [17]. They utilized contact wrench cone methodology to incorporate complex contact states and developed a stabilizer [18], [19]. However, the skateboard used in the experiments had no steering ability, which simplified the dynamics of robot skateboarding.

Compared to previous works, this research addresses the following three key challenges in the context of a quadruped robot skateboarding with steerable wheels:(1) The dynamics with dual floating bases have not been extensively studied, necessitating the development of a new model from scratch;(2) Pushing skateboard with one leg and maintaining

[1] Xiaomi Robotics Lab, Xiaomi Science Park, Anningzhuang Road, Haidian District, Beijing, 100085, P.R.China.
[2] University of Science and Technology of China, No.96, JinZhai Road Baohe District, Hefei, Anhui, 230026, P.R.China.
[*] Corresponding author. Email:xiangdiyun@gmail.com

balance with the remaining three legs involves complex behavior, the compute-intensive trajectory generation is not feasible for robot onboard processor;(3) To maintain balance on a flexible skateboard, the controller must consider the system dynamics while operating within limited resources.

B. Contribution

This paper aims to explore the task of riding a skateboard with the quadrupedal robot *CyberDog2*. In contrast to roller-skating or skiing, the connection between the robot's feet and the skateboard has to be ensured by the controller, rather than through mechanical attachments. Both the robot and the skateboard are floating-base systems, where the control of the underactuated system solely relies on the contact forces between them. Achieving balance, steering, and forward acceleration on the skateboard necessitates not only a motion library that generates feasible trajectories for the controller to follow, but also a robust controller that can effectively handle physical limitations and uncertainties.

The contributions of this paper are as follows:

1) We proposed a model of the system, which considers both robot and skateboard dynamics. Single rigid body and full kinematics model (SRBKM) [20] for robot is applied, since joint limitations are considered. The entire process is divided into three modes: four-leg balancing, three-leg balancing and acceleration.
2) To overcome the computational limitations of on-board processing, we developed an offline trajectory optimization (TO) method. This method generated a motion library with constraints of system dynamics and kinematics. By utilizing this library, the robot can propel itself from a static state and achieve different velocities by following various trajectories.
3) We introduce a online controller with linearized MPC and whole body control (WBC) to maintain the robot balance. The system dynamics are linearized based on reasonable assumptions. Despite the limited computational resources on the commercial robot, the controller operates at a fast rate, which are 30Hz for MPC and 500Hz for WBC, to guarantee system stability.

This paper is organized as follows. Section II introduces the robot platform and the skateboard. Section III elaborates on the system models. Section IV describes the control pipeline and details the TO during acceleration. Section V provides detailed system linearization and controller development process. Section VI presents experiment results on real robot. We conclude this research and give suggestion of future work in Section VII.

II. Hardware & System

A. Robot Platform

The robotic platform shown in Fig. 1 is the commercial quadrupedal robot *CyberDog2* developed by Xiaomi Robotics Lab. It weights about 8.9kg and is similar in size to MIT Mini Cheetah [21]. The robot is equipped with twelve torque-controllable quasi-direct drive actuators, which have outstanding transparency due to low gear ratio and friction.

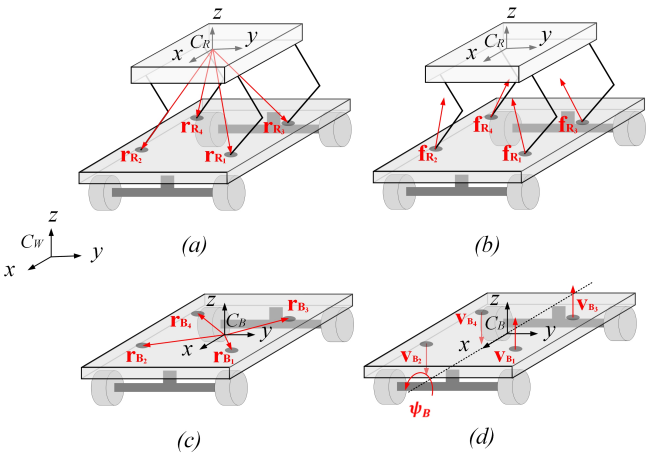

Fig. 2: The world, robot and skateboard frames, as well as the foot numbers are shown in this figure. (a) $\mathbf{r}_{R_i}$ in world frame are vectors from robot frame origin to each foot; (b) contact forces between feet and skateboard are described in world frame;(c)$\mathbf{r}_{B_i}$ in world frame are vectors from skateboard frame origin to contact points; (d) The skateboard angular velocity $\boldsymbol{\psi}_B$ determines contact point velocities w.r.t C_B.

Based on the inertia measurement unit (IMU) located at the center of mass (CoM) of body and encoders mounted as each joint, MR813 (Allwinner Technology, Quad A53 1.5GHz) processes state estimator, runs motion controller and generates targets for actuators to follow. Nvidia TX2 is used for mapping and localization based on the robot's states and environment information.

B. Skateboard

We take a commercially available handmade skateboard designed for medium-size dogs for *CyberDog2* to ride. The skateboard has a size of 0.265 * 0.74 * 0.11m and equipped with two steerable trunks. Length between front and rear trunk is 0.49m and width of each trunk is 0.24m.

III. System Modeling

In this section, we outline the derivation of the dynamic system for robot skateboarding. The skateboarding task is divided into three modes: four-leg balancing, three-leg balancing, and acceleration. Frames used to describe the system and some variables definition are shown in Fig. 2.

A. Four-Leg Balancing

1) Robot Dynamics: Using the SRBKM model formulation, the robot's dynamic model can described using:

$$\mathbf{x}_R = \begin{bmatrix} \mathbf{p}_R, & \boldsymbol{\theta}_R, & \mathbf{v}_R, & \boldsymbol{\omega}_R, & \mathbf{q}_j \end{bmatrix}^T \tag{1}$$

where $\mathbf{p}_R = [p_{Rx}, p_{Ry}, p_{Rz}]^T$ and $\mathbf{v}_R = [v_{Rx}, v_{Ry}, v_{Rz}]^T$ are the position and linear velocity of robot in world coordinate, respectively. $\boldsymbol{\theta}_R = [\theta_{roll}, \theta_{pitch}, \theta_{yaw}]^T$ represents the orientation of robot expressed using Euler angles with Z-Y-X rotation sequence, $\boldsymbol{\omega}_R = [\omega_x, \omega_y, \omega_z]^T$ is the angular velocity of the robot expressed in it's own coordinate and $\mathbf{q}_j \in \mathbb{R}^{12}$ denotes the vector of the robot's leg joint angles.

The control variables are:

$$\mathbf{u}_R = [\mathbf{f}_i, \dot{\mathbf{q}}_j]^T \quad (i = 1, ..., n_R \quad j = 1, ..., n_j) \tag{2}$$

where n_R is the number of support legs, which varies in different modes. n_j is the number of joints, which is twelve for this platform. Thus, the dynamics of robot system in the four-leg balancing mode can be expressed as:

$$\begin{bmatrix} \dot{\mathbf{p}}_R \\ \dot{\boldsymbol{\theta}}_R \\ \dot{\mathbf{v}}_R \\ \dot{\boldsymbol{\omega}}_R \\ \dot{\mathbf{q}}_j \end{bmatrix} = \begin{bmatrix} \mathbf{v}_R \\ \boldsymbol{T}(\boldsymbol{\theta}_R)\boldsymbol{\omega}_R \\ \frac{1}{m_R}\sum_{i=1}^{n_R} \mathbf{f}_i + \mathbf{g} \\ \boldsymbol{I}_R^{-1}(\boldsymbol{R}_R^T \sum_{i=1}^{n_R}([\mathbf{r}_{R_i}]_\times \mathbf{f}_i) - [\boldsymbol{\omega}_R]_\times(\boldsymbol{I}_R\boldsymbol{\omega}_R)) \\ \dot{\mathbf{q}}_j \end{bmatrix} \tag{3}$$

where $\boldsymbol{T}(\theta_R)$ represents the mapping between the angular velocity in the robot's frame and the differentiation of Euler angles[22], m_R is the robot's total mass, $\boldsymbol{I}_R$ denotes the robot's moment of inertia in body frame, $\boldsymbol{R}_R$ stands for the rotation between the robot's body frame and the world frame. $[\ \ ]_\times$ refers to the cross product matrix of a 3×1 vector.

2) Skateboard's Dynamics: While in the four-leg balancing mode, our sole focus is on the skateboard's velocity and posture. By manipulating the skateboard's roll angle, the robot is able to control skateboard's heading direction. Since we constraint the robot to not rotate relative to the skateboard in z_B, the skateboard's yaw and corresponding angular velocity are equal to those of the robot. Hence, the skateboard's state consists of the 2-dimensional planar velocity and roll rotation in world frame, as well as angular velocity in skateboard frame. They are expressed as follows:

$$\mathbf{x}_B = [\mathbf{v}_B, \boldsymbol{\varphi}_B, \boldsymbol{\psi}_B]^T = [v_{Bx}, v_{By}, \varphi_{roll}, \psi_{roll}]^T \tag{4}$$

We limit the robot only to exhibit lateral translation w.r.t. the skateboard in this mode. Thus, skateboard acceleration equals to the projection of the robot's acceleration along the skateboard's x^B. This can be expressed by :

$$\dot{\mathbf{v}}_B = \boldsymbol{T}_{v_B}\boldsymbol{R}_R^T\dot{\mathbf{v}}_R \tag{5a}$$

$$\boldsymbol{T}_{v_B} = \begin{bmatrix} cos(\theta_{yaw}) & -sin(\theta_{yaw}) \\ sin(\theta_{yaw}) & cos(\theta_{yaw}) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5b}$$

where $\boldsymbol{T}_{v_B}$ is a matrix that transfers local velocity to world frame. Since the skateboard cannot rotate in pitch direction, the differentiation of the skateboard's roll is simplified to $\dot{\boldsymbol{\varphi}}_B = \boldsymbol{\psi}_B$ based on the $\boldsymbol{T}()$ mapping matrix in Eq. 3.

$\boldsymbol{I}_B$ only considers inertia of the wooden board instead of all skateboard components. For term $\dot{\boldsymbol{\psi}}_B$, the $\begin{bmatrix}\boldsymbol{\psi}_B & 0 & 0\end{bmatrix}_\times^T(\boldsymbol{I}_B\begin{bmatrix}\boldsymbol{\psi}_B & 0 & 0\end{bmatrix}^T)$ can be trimmed because the skateboard is very close to a homogeneous wooden cuboid, whose inertia elements are decoupled in three directions . Additionally, the skateboard's steering mechanism is somewhat complicated, we adopt the approximate representation described in [23] where a Spring-Damper system is used to formulate the trunk torque in roll direction of skateboard as follow:

$$\boldsymbol{l}(\boldsymbol{\varphi}_B, \boldsymbol{\psi}_B) = -K_B\boldsymbol{\varphi}_B - B_B\boldsymbol{\psi}_B \tag{6}$$

In this way, dynamics of skateboard are presented as:

$$\begin{bmatrix} \dot{\mathbf{v}}_B \\ \dot{\boldsymbol{\varphi}}_B \\ \dot{\boldsymbol{\psi}}_B \end{bmatrix} = \begin{bmatrix} \boldsymbol{T}_{v_B}\boldsymbol{R}_R^T\dot{\mathbf{v}}_R \\ \boldsymbol{\psi}_B \\ \frac{\boldsymbol{S}_{\psi_B}}{I_{B_x}}(-\boldsymbol{R}_B^T\sum_{i=1}^{n_B}([\mathbf{r}_{B_i}]_\times\mathbf{f}_i) + \boldsymbol{l}(\boldsymbol{\varphi}_B, \boldsymbol{\psi}_B) \end{bmatrix} \tag{7}$$

where $\boldsymbol{S}_{\psi_B} = [1 \quad 0 \quad 0]^T$ is a selection vector for term $\boldsymbol{\psi}_B$. $\boldsymbol{R}_B$ is the rotation matrix to world frame with roll of φ_{roll} and yaw of θ_{yaw}. I_{B_x} represents x^B axis element of $\boldsymbol{I}_B$. Feet positions in the skateboard's frame are ${}^C\mathbf{r}_{B_i} = [{}^Cr_{B_{i_x}} \quad {}^Cr_{B_{i_y}} \quad 0]^T$, which are predefined as constants. When considering the skateboard's orientation, the position vector of the robot's feet in the world frame can be expressed as:

$$\mathbf{r}_{B_i} = \boldsymbol{R}_B{}^C\mathbf{r}_{B_i} \tag{8}$$

Therefore, the dynamics of system in this mode is Eq. 3 for first twenty-four rows and Eq. 7 for last four rows. The number of contact for robot n_R and number of contact for skateboard n_B are four.

B. Three-Leg Balancing

In this state the robot balances on the skateboard using only three legs while swinging the rear-left leg from ground to skateboard based on a predefined Bézier trajectory. This state serves as the transitional state between four-leg balancing and acceleration. During this state the robot does not move relative to the skateboard without any posture variations. The remaining state variables are the same as those of Eq. 7, except for $n_R = n_B = 3$.

C. Accelerating

During acceleration, one leg periodically pushes off the ground to propel the system forward, while the other three adjust contact forces to balance the system. Simultaneously ensuring propelling force, robot balance, and skateboard posture for straight movement increases task complexity compared to previous sections, as the velocity of the robot torso may vary from that of skateboard in $x - y$ plane.

1) Dynamics during acceleration: Different from balancing with four or three legs, robot torso may swing along x^B and y^B axes, so $\mathbf{v}_B$ cannot be derived from $\mathbf{v}_R$. The differentiation of skateboard linear velocity is computed by projecting forces in x^B.

$$\dot{\mathbf{v}}_B = \boldsymbol{S}_{xy}(\boldsymbol{R}_B\dot{\mathbf{v}}_B^B) \tag{9a}$$

$$\dot{\mathbf{v}}_B^B = \frac{1}{m_B}\boldsymbol{S}_x(-\boldsymbol{R}_B^T\sum_{i=1}^{n_B}\mathbf{f}_i - sign(v_{B_x}^B)\mathbf{f}_{fri}) \tag{9b}$$

where $\boldsymbol{S}_{xy}$ is a 2×3 matrix that selects the first two rows of a given matrix, while 3×3 matrix $\boldsymbol{S}_x$ is specifically used for selecting element in the x^B direction. $\mathbf{v}_B^B$ represents the skateboard velocity projection in skateboard frame. $v_{B_x}^B$ denotes its x^B element. Friction in x^B are considered as a constant value during acceleration, which means $\mathbf{f}_{fri} = [f_{x^B} \quad 0 \quad 0]^T$. The rest of the state variable derivatives remain same as Eq. 7 except for $\mathbf{v_B}$. Note that in this mode $n_R = 4, n_B = 3$. Positions of the three feet in contact with skateboard can be computed by Eq.8.

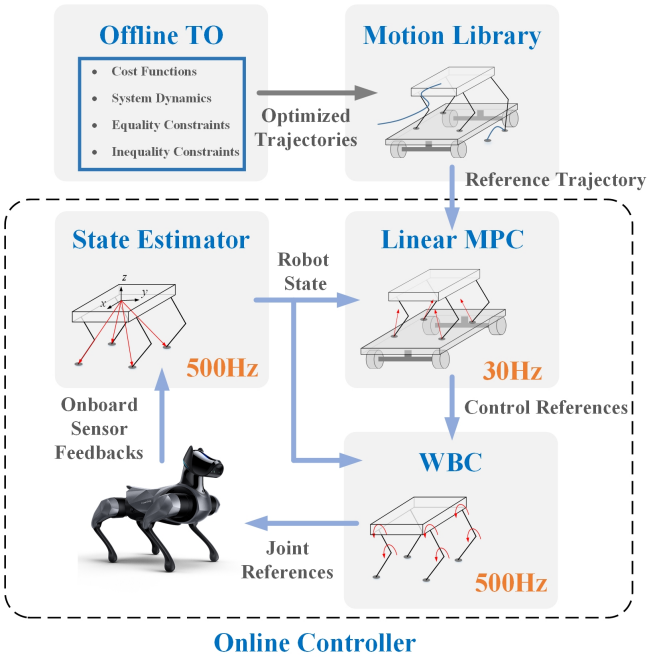

Fig. 3: The control pipeline consists of five modules: Offline TO, Motion Library, Linear MPC, WBC, and State Estimator. Offline TO generates optimized trajectories for different velocities, which are then stored in the Motion Library for the online controller to select from. Linear MPC for modifying TO trajectories to keep balance runs at 30Hz, while WBC for generating joint references and State Estimator for computing robot state operate at 500Hz.

IV. Motion Planning

The control framework and inter-module relationships are illustrated in Figure 3. The state estimator uses IMU and joint sensors to compute robot states and skateboard posture. A motion library holds multiple trajectories for the robot to accelerate to different velocities, generated via offline TO. After completing the acceleration, the robot switches to the balancing mode and steers the skateboard by commands. The QP-style controller considers different dynamics and ensures robot balance throughout the entire process.

A. Contact scheduler

The contact scheduler in this study uses periodic sequences of swing and stance phases. The acceleration trajectories have consistent period of 1.2 seconds but vary in the number of strides. During acceleration, only the leg responsible for propulsion switches between phases, while the remaining legs maintain contact with the skateboard.

B. Offline Trajectory Optimization

The acceleration process requires robot to generate a propel force and maintain system balance simultaneously by adjusting its contact forces and torso position. Achieving this solely through heuristics is extremely hard. Hence, offline TO is applied to construct a the motion library. To solve the problem, the offline TO is converted to a nonlinear programming(NLP) problem by multiple shooting [24]. We rely on IPOPT [25] to solve the NLP problem and store the optimized offline trajectories in onboard computer for the robot to execute.

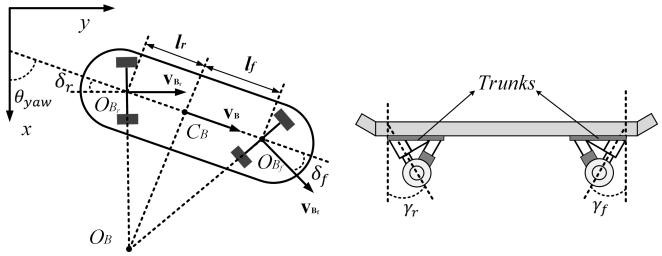

Fig. 4: Robot and skateboard rotate along O_B. Subscript f and r denote front and rear trunk respectively. $\boldsymbol{\gamma}_{f/r}$ are rake angles.

1) Dynamics Constraint: The dynamics constraint mentioned in Section III should be revised to adapt to optimization. In Eq. 9, the $sign$ is discontinuous and non-differentiable, which is incompatible for optimization. So we approximate it with the function $tanh$.

$$\dot{\mathbf{v}}_B^B = \frac{1}{m_B}\boldsymbol{S}_x(-\boldsymbol{R}_B^T\sum_{i=1}^{n_B}\mathbf{f}_i - tanh(kv_{B_x}^B)\mathbf{f}_{fri}) \tag{10}$$

where k denotes a scaling factor that reflects the degree of approximation.

2) Equality Constraints: Besides dynamics equality constraints, the system has to satisfy a dubins car style kinematics constraint, in which forward linear velocity and angular velocity are coupled due to skateboard mechanism. Based on previous research about skateboard kinematics[23], we derived the relationship. The steering angles $\delta_{f/r}$ of front and rear trunks are determined by φ_{roll} and so-called rake angles $\gamma_{f/r}$ shown in Fig. 4.

$$sin(\varphi_{roll})cot(\gamma_{f/r}) = tan(\delta_{f/r}) \tag{11}$$

The point O_B locates at the intersection of two wheel axes, serving as the center of yaw rotation of skateboard. The line O_BC_B which is vertical to skateboard x^B axis determines the origin point of the skateboard local frame C_B. The velocity at C_B has no component in y_B direction, resulting in $v_{B_y}^B = 0$. The location of C_B can be computed using simple geometry as Eq.12, where $l_{f/r}$ are distances between C_B and $O_{B_{f/r}}$.

$$\frac{l_f}{l_r} = \frac{tan(\delta_f)}{tan(\delta_r)} = \frac{cot(\gamma_f)}{cot(\gamma_r)} \tag{12}$$

We define $K_{board} = l_f/cot(\gamma_f) = l_r/cot(\gamma_r)$ to simplify following equations.

Velocity projections along x^B direction are equal at C_B, O_{B_f} and O_{B_r}. So we can derive the velocity projections at O_{B_f} and O_{B_r} in y_B direction. Additionally, they also satisfy equations about angular velocity in z_W direction.

$$v_{B_{f_y}}^B = -v_{B_x}^B tan(\delta_f) = v_{B_y}^B + \omega_z l_f \tag{13a}$$

$$v_{B_{r_y}}^B = v_{B_x}^B tan(\delta_r) = v_{B_y}^B - \omega_z l_r \tag{13b}$$

So, from Eq. 11 to Eq. 13, the relationship between skateboard linear velocity and angular velocity can be established.

$$sin(\varphi_{roll})\boldsymbol{S}_x\boldsymbol{R}_B^T\mathbf{v}_B^B + K_{board}\boldsymbol{S}_z\boldsymbol{\omega}_R = 0 \tag{14}$$

where $\boldsymbol{S}_x$ is same as that in Eq. 9, while $\boldsymbol{S}_z$ selects z^B.

Another equality constraint we focus on is the feet constraints, which guarantees continuous contact between feet and skateboard or ground. When standing on skateboard, foot must move with the skateboard. The velocity of contact point projected onto skateboard frame can be computed based on the feet and the skateboard velocities in world. Moreover, the contact points velocities also depend on $\boldsymbol{\psi}_B$ as Fig. 2(d) depicted. The equation is described as:

$$\mathbf{v}_i^B = \boldsymbol{R}_B^T(\mathbf{v}_i - [v_{B_x}, v_{B_y}, 0]^T) = [\boldsymbol{\psi}_B, 0, 0]_\times^T \mathbf{r}_{B_i} \quad (15)$$

3) Inequality Constraints: The inequality constraints encompass several standard limitations. These include ensuring that the joints operate within their physical bounds, that the contact forces adhere to the friction cone constraints, and that the initial state matches the specified state. Additionally, φ_B is restricted to prevent board collisions with the wheels.

4) Cost Function: The quadratic cost function aims to minimize the disparities between the optimized states and desired states with minimal effort. Contact force and joint velocity variations are considered to get smooth trajectories.

$$\min_{\mathbf{X}_R, \mathbf{X}_B, \mathbf{U}_R} \int_{t=t_0}^{T} (\left\|\mathbf{x}_B(t) - \mathbf{x}_{B_{des}}(t)\right\|^2_{\mathbf{W}_{\mathbf{x}_B}} + \quad (16a)$$

$$\left\|\mathbf{x}_R(t) - \mathbf{x}_{R_{des}}(t)\right\|^2_{\mathbf{W}_{\mathbf{x}_R}} + \quad (16b)$$

$$\left\|\mathbf{u}_R(t)\right\|^2_{\mathbf{W}_\mathbf{u}} + \quad (16c)$$

$$\left\|\mathbf{u}_R(t) - \mathbf{u}_R(t-dt)\right\|^2_{\mathbf{W}_{\triangle \mathbf{u}}}) dt \quad (16d)$$

$\mathbf{X}_R = [\mathbf{x}_R(0), ...\mathbf{x}_R(T)]^T$, $\mathbf{X}_B = [\mathbf{x}_B(0), ...\mathbf{x}_B(T)]^T$, $\mathbf{u}_R = [\mathbf{u}_R(0), ...\mathbf{u}_R(T)]^T$ are optimized variables in TO.

V. Motion Control

Online TO is applied in many excellent works on legged locomotion. However, due to the limited computation resource of onboard processor, online TO is infeasible on this commercial product. Hence, we adopt a linearization method used in MIT Cheetah [26] to develop the online controller.

A. State Estimator

The state estimator is used to compute both robot state and skateboard state. Robot posture $\boldsymbol{\theta}_R$ and angular velocity $\boldsymbol{\omega}_R$ are sensed directly by onboard IMU. The estimation of the robot's position and velocity is based on the method described in [27], with some minor modifications. The posture of the skateboard is estimated by leveraging the positions of the contact feet, following the approach described in [28]. The velocity of the skateboard is computed by averaging velocities of the stance feet.

B. Linear MPC

Since desired roll and pitch angles are zero, the derivative of robot Euler angles are simplified to $\dot{\boldsymbol{\theta}}_R = \boldsymbol{\omega}_R$. By assuming that state of robot are close to desire state, $\mathbf{r}_{R_i}$, $\mathbf{r}_{B_i}$, $\boldsymbol{\theta}_R$ and $\boldsymbol{\theta}_B$ are set to be feedback values when MPC updates and stay constantly during one MPC loop. Therefore, corresponding algebraic terms are constants. Another assumption is that off-diagonal terms of inertia tensor are small, so that $[\boldsymbol{\omega}_R]_\times(\boldsymbol{I}_R\boldsymbol{\omega}_R)$ is zero. The linearized continuous-time dynamics equation is

$$\begin{bmatrix}\dot{\mathbf{x}}_R \\ \dot{\mathbf{x}}_B\end{bmatrix} = \boldsymbol{A}\begin{bmatrix}\mathbf{x}_R \\ \mathbf{x}_B\end{bmatrix} + \boldsymbol{B}\mathbf{u}_R + \boldsymbol{C} \quad (17)$$

where $\boldsymbol{A} \in \mathbb{R}^{28\times28}$, $\boldsymbol{B} \in \mathbb{R}^{28\times24}$ and $\boldsymbol{C} \in \mathbb{R}^{28}$.

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{0}_{3\times6} & \boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times14} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times6} & \mathbf{0}_{3\times3} & \boldsymbol{T}(\boldsymbol{\theta_R}) & \mathbf{0}_{3\times14} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{20\times6} & \mathbf{0}_{20\times3} & \mathbf{0}_{20\times3} & \mathbf{0}_{20\times14} & \mathbf{0}_{20\times1} & \mathbf{0}_{20\times1} \\ \mathbf{0}_{1\times6} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times14} & 0 & 1 \\ \mathbf{0}_{1\times6} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times14} & -K_B & -B_B \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} \mathbf{0}_{6\times3} & ... & \mathbf{0}_{6\times3} & \mathbf{0}_{6\times12} \\ \boldsymbol{I}_{3\times3}/m_R & ... & \boldsymbol{I}_{3\times3}/m_R & \mathbf{0}_{3\times12} \\ \boldsymbol{I}_R^{-1}\boldsymbol{R}_R^T[\mathbf{r}_{R_1}]_\times & ... & \boldsymbol{I}_R^{-1}\boldsymbol{R}_R^T[\mathbf{r}_{R_4}]_\times & \mathbf{0}_{3\times12} \\ \mathbf{0}_{12\times3} & ... & \mathbf{0}_{12\times3} & \boldsymbol{I}_{12\times12} \\ \boldsymbol{B}_{\mathbf{v}_B} & ... & \boldsymbol{B}_{\mathbf{v}_B} & \mathbf{0}_{2\times12} \\ \mathbf{0}_{1\times3} & ... & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times12} \\ -\frac{\boldsymbol{S}_{\psi_B}}{I_{B_x}}\boldsymbol{R}_B^T[\mathbf{r}_{B_1}]_\times & ... & -\frac{\boldsymbol{S}_{\psi_B}}{I_{B_x}}\boldsymbol{R}_B^T[\mathbf{r}_{B_4}]_\times & \mathbf{0}_{1\times12} \end{bmatrix}$$

$$\boldsymbol{C} = \begin{bmatrix}\mathbf{0}_{6\times1} & \mathbf{g} & \mathbf{0}_{15\times1} & \boldsymbol{C}_{\mathbf{v}_B} & \mathbf{0}_{2\times1}\end{bmatrix}^T$$

$\boldsymbol{B}_{\mathbf{v}_B} = 1/m_R\boldsymbol{T}_{v_B}(\boldsymbol{\theta}_R)\boldsymbol{R}_R^T$, $\boldsymbol{C}_{\mathbf{v}_B} = \boldsymbol{T}_{v_B}\boldsymbol{R}_R^T\mathbf{g}$ during four or three legs balancing; $\boldsymbol{B}_{\mathbf{v}_B} = -1/m_B\boldsymbol{S}_{xy}\boldsymbol{R}_B(\boldsymbol{S}_x\boldsymbol{R}_B^T)$, $\boldsymbol{C}_{\mathbf{v}_B} = -1/m_B\boldsymbol{S}_{xy}\boldsymbol{R}_B(\boldsymbol{S}_x sign(v_{B_x}^B)\boldsymbol{f}_{fri})$ in acceleration mode. $sign()$ is set to ±1 or 0 in $\boldsymbol{C}_{\mathbf{v}_B}$ based on feedback velocity when MPC updates.

With above simplifications, the discrete dynamics is

$$\mathbf{X}(k+1) = \boldsymbol{A}_k\mathbf{X}(k) + \boldsymbol{B}_k\mathbf{U}(k) + \boldsymbol{C}_k \quad (18)$$

where, $\mathbf{X}(k) = [\mathbf{x}_R(k), \mathbf{x}_B(k)]^T$ and $\mathbf{U}(k) = \mathbf{u}_R(k)$.

Then, we construct a QP problem that minimizes

$$\min_{\mathbf{X},\mathbf{U}} \sum_{k=0}^{N} \left\|\mathbf{X}(k+1) - \mathbf{X}_{ref}(k+1)\right\|_{\boldsymbol{Q}} + \left\|\mathbf{U}(k)\right\|_{\boldsymbol{R}} \quad (19)$$

which subject to initial condition, dynamics and contact forces constraints. At the end of control pipeline, a whole body controller [26] computes references for actuators.

VI. Experiment

Experiments are separated into two sections, which are balancing test to prove proposed controller and propelling test to validate motion library.

A. Balancing Tests

The first experiment verifies the controller during four-leg balance mode. Robot steers skateboard by adjusting its torso lateral position and contact forces.

To avoid overshooting of skateboard posture control, the period of reference is set to three seconds. High frequency steering is not common used on skateboard due to the stiffness and damping of mechanism. Experiment results showcase skateboard roll angle, along with corresponding robot lateral position and contact forces. The maximum angle of $\boldsymbol{\varphi}_B$ is about 0.12rad, so that amplitude of reference is set to 0.1rad. The proposed controller turns out to be performing well in balancing the system while steering the skateboard.

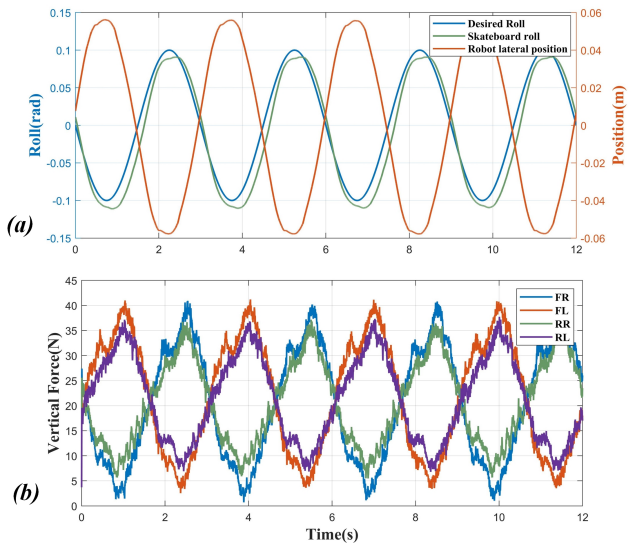

Fig. 5: The results of skateboard posture control. Reference roll, feedback skateboard roll and corresponding torso lateral position are shown in (a). Robot adjusts its torso position, so that the contact forces control skateboard posture to track reference. Vertical contact forces of four legs are presented in (b). Since forces approaching zero during experiment, robot almost has no potential to reach further lateral position.

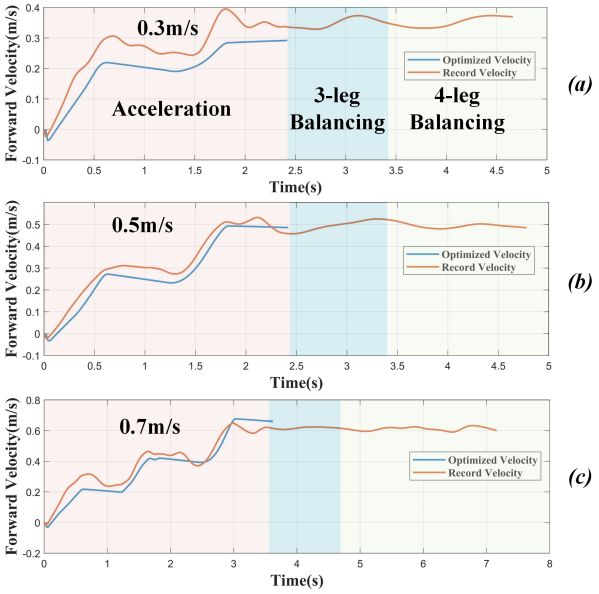

Fig. 6: The results of skateboard accelerating to different velocities are shown. Three skateboarding modes are emphasized by different colors. TO only optimizes trajectories in acceleration mode, while online controller operates throughout whole procedure. The results demonstrate that the motion library is effective in enabling the system to achieve various velocities.

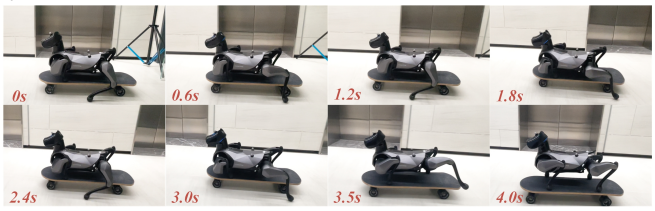

Fig. 7: The snapshots of 0.7m/s acceleration. Robot pushes off ground from 0s to 3s and then transfers to four-leg balancing at 4s.

B. Acceleration Tests

Three optimized trajectories, which accelerate the system to various velocities, are examined in this section. We apply motion capture to get ground truth speed of the system. Four ultra-red cameras collect position data of three markers which are fixed on the back of robot. Therefore, velocities can be roughly obtained by taking average of marker position derivatives. Since the update of motion capture is around 20Hz, the data shown in Fig. 6 is heavily filtered. However, three acceleration procedures are evidenced clearly. Experiment snapshots of 0.7m/s is presented in Fig. 7.The successful experiments serve as conclusive evidences of the motion library's effectiveness. Since parameter variations, such as stiffness of skateboard steering mechanisms and terrain frictions, are not included in modelling, the experiments in supplementary video proved the robustness of the discussed control pipeline.

VII. Conclusion

This paper presents a control pipeline for achieving quadrupedal locomotion control with the *CyberDog2* robot riding a skateboard. The lack of previous research on robot skateboarding can be attributed to several challenges. Firstly, the two floating base system, with only reaction forces to maintain contact, makes the modeling process complex. Secondly, the intricate acceleration process requires a planner to coordinate all state and control trajectories effectively. Lastly, a controller must operate within limited resources and stabilize the system in real time. These challenges have hindered previous research efforts in robot skateboarding.

The system is modeled as three modes: four-leg balancing, three-leg balancing, and acceleration, where the dynamics varies depending on the relationship between the robot and skateboard velocities. An offline TO method generated acceleration trajectories that optimize both legs propelling force to achieve the desired velocity and torso position to keep system balance, creating a motion library. Finally, a linear MPC was derived based on the simplified dynamics model to ensure balance on the skateboard. To simplify the complex system dynamics, reasonable assumptions about the system were utilized to enable linearization and satisfy the required update rate. The QP problem is then solved to obtain references for the WBC module, which computes the desired torque at a higher update rate.

In real experiments, we demonstrated successful balancing on the skateboard using the proposed online controller. Throughout acceleration experiments, we achieved three distinct acceleration trajectories. Different terrain tests are provided in supplementary video. These results provide evidence for the effectiveness of the control pipeline proposed.

In the future, we aim to further enhance forward velocity tracking accuracy by incorporating visual odometry. Additionally, we plan to explore more sophisticated controllers, such as the nonlinear MPC framework like OCS2 [29], to improve the robot's performance on the skateboard.

REFERENCES

[1] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson, "Optimization based full body control for the atlas robot," in *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 120–127.

[2] E. Guizzo, "By leaps and bounds: An exclusive look at how boston dynamics is redefining robot agility," *IEEE Spectrum*, vol. 56, no. 12, pp. 34–39, 2019.

[3] M. Hutter, C. Gehring, A. Lauber, F. Gunther, C. D. Bellicoso, V. Tsounis, P. Fankhauser, R. Diethelm, S. Bachmann, M. Blösch *et al.*, "Anymal-toward legged robots for harsh environments," *Advanced Robotics*, vol. 31, no. 17, pp. 918–931, 2017.

[4] 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)*. IEEE, 2018, pp. 2245–2252.

[5] M. Bjelonic, C. D. Bellicoso, M. E. Tiryaki, and M. Hutter, "Skating with a force controlled quadrupedal robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7555–7561.

[6] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch *et al.*, "Anymal-a highly mobile and dynamic quadrupedal robot," in *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2016, pp. 38–44.

[7] M. Bjelonic, P. K. Sankar, C. D. Bellicoso, H. Vallery, and M. Hutter, "Rolling in the deep–hybrid locomotion for wheeled-legged robots using online trajectory optimization," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3626–3633, 2020.

[8] E. Vollenweider, M. Bjelonic, V. Klemm, N. Rudin, J. Lee, and M. Hutter, "Advanced skills through multiple adversarial motion priors in reinforcement learning," *arXiv preprint arXiv:2203.14912*, 2022.

[9] M. Bjelonic, R. Grandia, O. Harley, C. Galliard, S. Zimmermann, and M. Hutter, "Whole-body mpc and online gait sequence generation for wheeled-legged robots," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8388–8395.

[10] M. Bjelonic, R. Grandia, M. Geilinger, O. Harley, V. S. Medeiros, V. Pajovic, E. Jelavic, S. Coros, and M. Hutter, "Offline motion libraries and online mpc for advanced mobility skills," *The International Journal of Robotics Research*, vol. 41, no. 9-10, pp. 903–924, 2022.

[11] M. Tranzatto, F. Mascarich, L. Bernreiter, C. Godinho, M. Camurri, S. Khattak, T. Dang, V. Reijgwart, J. Loeje, D. Wisth *et al.*, "Cerberus: Autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the darpa subterranean challenge," *arXiv preprint arXiv:2201.07067*, 2022.

[12] K. Kim, P. Spieler, E.-S. Lupu, A. Ramezani, and S.-J. Chung, "A bipedal walking robot that can fly, slackline, and skateboard," *Science Robotics*, vol. 6, no. 59, p. eabf8136, 2021.

[13] Disney. Storytelling Through Characters at Disney Parks I SXSW 2023. 2023, Mar 11. [Online]. Available: https://www.youtube.com/watch?v=lPqzLE4KjhI

[14] S.-H. Jo, J.-U. Chu, and Y.-J. Lee, "Motion planning for biped robot with quad roller skates," in *2008 International Conference on Control, Automation and Systems*. IEEE, 2008, pp. 1173–1177.

[15] K. Ishihara, T. D. Itoh, and J. Morimoto, "Full-body optimal control toward versatile and agile behaviors in a humanoid robot," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 119–126, 2019.

[16] K. Kojima, T. Karasawa, T. Kozuki, E. Kuroiwa, S. Yukizaki, S. Iwaishi, T. Ishikawa, R. Koyama, S. Noda, F. Sugai *et al.*, "Development of life-sized high-power humanoid robot jaxon for real-world use," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 838–843.

[17] J. Koenemann, A. Del Prete, Y. Tassa, E. Todorov, O. Stasse, M. Bennewitz, and N. Mansard, "Whole-body model-predictive control applied to the hrp-2 humanoid," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3346–3351.

[18] N. Takasugi, K. Kojima, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "Real-time skating motion control of humanoid robots for acceleration and balancing," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1356–1363.

[19] N. Takasugi, K. Kojima, S. Nozawa, F. Sugai, K. Yohei, K. Okada, and M. Inaba, "Extended three-dimensional walking and skating motion generation for multiple noncoplanar contacts with anisotropic friction: Application to walk and skateboard and roller skate," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 9–16, 2018.

[20] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 295–302.

[21] B. Katz, J. Di Carlo, and S. Kim, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6295–6301.

[22] B. Siciliano, O. Khatib, and T. Kröger, *Springer handbook of robotics*. Springer, 2008, vol. 200.

[23] B. Varszegi, D. Takacs, G. Stepan, and S. J. Hogan, "Stabilizing skateboard speed-wobble with reflex delay," *Journal of The Royal Society Interface*, vol. 13, no. 121, p. 20160345, 2016.

[24] H. G. Bock and K.-J. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," *IFAC Proceedings Volumes*, vol. 17, no. 2, pp. 1603–1608, 1984.

[25] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 2006.

[26] J. Di 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)*. IEEE, 2018, pp. 1–9.

[27] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State estimation for legged robots-consistent fusion of leg kinematics and imu," *Robotics*, vol. 17, pp. 17–24, 2013.

[28] C. Gehring, C. D. Bellicoso, S. Coros, M. Bloesch, P. Fankhauser, M. Hutter, and R. Siegwart, "Dynamic trotting on slopes for quadrupedal robots," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 5129–5135.

[29] F. Farshidian *et al.*, "OCS2: An open source library for optimal control of switched systems," [Online]. Available: https://github.com/leggedrobotics/ocs2.