Convex Model Predictive Control of Single Rigid Body Model on SO(3) for Versatile Dynamic Legged Motions

Junjie Shen[1] and Dennis Hong[1]

Abstract— **This paper presents a convex model predictive control framework for versatile dynamic legged motions with negligible leg dynamics. The framework utilizes the single rigid body model linearly approximated around the operating point. With ground reaction forces as direct control inputs to the system, no reference control trajectory needs to be specified in advance. By using the rotation matrix for the evolution of rotational dynamics, issues arising from other representations can be avoided. Moreover, the rotation matrix is parametrized using the history of angular velocity without introducing additional variables. The effect is that we can still take the orientation into consideration efficaciously without directly working on it. The framework tackles the robot reference tracking problem via trajectory optimization, which is formulated into a standard quadratic program and can be solved efficiently in real time with guaranteed optimality. It was verified on various legged robots with different numbers of legs for performing different types of dynamic motions in the simulation environment. We thus envision a promising future of the proposed convex model predictive control framework in legged robots and potentially in other applications as well.**

I. Introduction

Legged robots, despite their increased complexity compared with other types of robot, have the potential to exert a much larger influence on human environments in the future. The articulated limbs provide them with 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.

As one of the major challenges, controlling dynamic motions for legged robots is extremely difficult. First, the robot movement only results from the contact of the feet with the environment. These contact forces are strongly restricted and thus need to be carefully planned to achieve the desired behavior. Second, the system dynamics is highly nonlinear and complex, nominally underactuated and unstable, multi-input and multi-output, as well as time-variant and hybrid [1]. One of the promising approaches that has been proven effective is model predictive control (MPC). It usually considers solving a trajectory optimization (TO) problem in real time, which determines the control sequences over a receding prediction horizon into the future. Successful implementations of whole-body MPC on legged robots have shown its great capability of simultaneously planning and controlling complex dynamic motions [2], [3]. However, directly involving the sophisticated full-body dynamics in TO is still computationally expensive and sometimes even intractable [4]–[8]. As a result, there exist a variety of methods using the empirically simplified models or so-called templates [9], which only focus on the most salient aspect of the system dynamics, such as the linear inverted pendulum model [10]–[12], centroidal momentum model [13]–[15], and single rigid body (SRB) model [16]–[23]. Among all these examples, the SRB model, which assumes negligible leg dynamics following the popular trend in light leg design [15], efficiently captures the effect of the net external wrench on the evolution of both robot body position and orientation.

However, it is still debatable how to properly parametrize the robot body orientation in an MPC framework. The orientation is originally described by the rotation matrix evolving on the manifold of the special orthogonal group $\mathsf{SO}(3)$ [24]. Unfortunately, optimization directly on rotation matrix leads to an overparametrization of the problem [25]. To solve this issue, many researchers use Euler angles instead to represent the body orientation [16]–[20]. Despite their intuitive interpretations, using Euler angles and applying the regular techniques of Euclidean spaces are not properly invariant under the action of rigid transformations [25], [26]. Moreover, Euler angles suffer from the singularity issue, which restricts the robot from performing motions involving large body angle deviation from the nominal horizontal plane. Unlike Euler angles, quaternion is a minimal globally nonsingular representation for orientation [27], but its state space of 3-sphere provides a double covering of $\mathsf{SO}(3)$ where a single orientation may correspond to two unit quaternions. This ambiguity should be carefully resolved, otherwise the unwinding phenomenon would occur where the body unnecessarily rotates through a large angle even if the initial orientation error is small [28]. As a consequence, it is problematic to have the quaternion involved in MPC. Recently, a variation-based control strategy is proposed for systems whose dynamics evolves on $\mathsf{SO}(3)$ [29]. Instead of working on the rotation manifold, it considers the orientation error in the tangent space, the description of which is not only singularity-free but also invariant to the orientation configuration [30]. Latest works have demonstrated its significant power to robustly stabilize various dynamic legged motions involving large orientation deviation, either using the original error dynamics in a nonlinear MPC fashion [21] or its linearly approximated counterpart in a convex MPC framework [22]. Nevertheless, intricate nonlinear optimization complicates the solving process and suffer local minima issue, while the variation-based linearization scheme is rather convoluted and additionally requires prespecifying

[1] Junjie Shen 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, dennishong}@ucla.edu

a reference control trajectory.

Inspired by the previous works, we address the aforementioned issues and make the following major contributions. Unlike the Euclidean space, optimization directly on the rotation manifold leads to an overparametrization of the problem [25]. In this work, we propose to parametrize the rotation matrix using the history of angular velocity. As a result, we are still able to take the orientation into consideration effectively without directly working on it. A novel convex MPC framework of SRB model on $\mathsf{SO}(3)$ is accordingly introduced along with a linearization scheme for the nonlinear rotational dynamics. The linear approximation is valid for a short period over the MPC prediction horizon, which is justified in the simulation environment.

The rest of this paper is organized as follows. Section II describes the SRB robot model, the parametrization of the rotation matrix, as well as the linearization scheme. Section III elaborates the proposed convex MPC framework based on quadratic programming (QP). To evaluate the performance, a series of simulations have been successfully conducted and the results are discussed in Section IV. Section V concludes the paper with potential future directions.

II. Robot Modeling

The robot model of interest is the SRB model subject to the ground reaction force (GRF) exerted on each stance leg, which assumes negligible leg dynamics and point contact. As a result, the entire mass and inertia of the robot can be considered lumped at the body center of mass (CoM).

A. Single Rigid Body Dynamics

The continuous-time dynamics of the SRB model, as illustrated in Fig. 1, can be summarized as follows:

$$\dot{\boldsymbol{p}} = \boldsymbol{v}, \tag{1a}$$

$$\dot{\boldsymbol{v}} = \frac{1}{m}\sum_{i=1}^{n} \boldsymbol{f}_i + \boldsymbol{g}, \tag{1b}$$

$$\dot{\boldsymbol{R}} = \boldsymbol{R}\widehat{\boldsymbol{\omega}}, \tag{1c}$$

$$\dot{\boldsymbol{\omega}} = \boldsymbol{I}^{-1}\left(\boldsymbol{R}^\top\left(\sum_{i=1}^{n} \boldsymbol{r}_i \times \boldsymbol{f}_i\right) - \boldsymbol{\omega}\times\boldsymbol{I}\boldsymbol{\omega}\right), \tag{1d}$$

where $\boldsymbol{p}, \boldsymbol{v} \in \mathbb{R}^3$ are the body CoM position and velocity, respectively, $\boldsymbol{f}_i \in \mathbb{R}^3$ is the GRF exerted on the ith foot, $\boldsymbol{R} \in \mathsf{SO}(3)$ is the rotation matrix representing the orientation of the body frame $\mathcal{B}$, $\boldsymbol{\omega} \in \mathbb{R}^3$ is the body angular velocity, $\boldsymbol{r}_i \in \mathbb{R}^3$ denotes the ith foot location $\boldsymbol{c}_i \in \mathbb{R}^3$ relative to CoM, $m \in \mathbb{R}$ is the robot mass, $\boldsymbol{I} \in \mathbb{R}^{3\times 3}$ is the fixed body moment of inertia tensor, $n \in \mathbb{N}$ is the number of legs, and $\boldsymbol{g} \in \mathbb{R}^3$ is the gravity vector. Note that $\boldsymbol{p}$, $\boldsymbol{v}$, $\boldsymbol{f}_i$, $\boldsymbol{R}$, $\boldsymbol{r}_i$, $\boldsymbol{c}_i$, $\boldsymbol{g}$ are all expressed in the inertia frame $\mathcal{I}$ while $\boldsymbol{\omega}$ and $\boldsymbol{I}$ are described in frame $\mathcal{B}$; $\widehat{\cdot}$ is the hat operator from $\mathbb{R}^3$ to the space of skew-symmetric matrices $\mathfrak{so}(3)$ such that $\widehat{\boldsymbol{a}}\boldsymbol{b} = \boldsymbol{a}\times\boldsymbol{b}$ for all $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^3$, e.g., if $\boldsymbol{a} = \begin{bmatrix} a_1, a_2, a_3\end{bmatrix}^\top \in \mathbb{R}^3$,

$$\widehat{\boldsymbol{a}} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \in \mathfrak{so}(3), \tag{2}$$

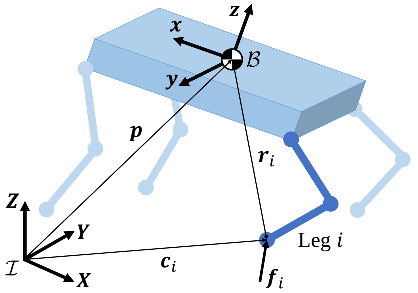

Fig. 1. Single rigid body model. $\mathcal{I}$ is the inertia frame and $\mathcal{B}$ is the robot body frame. $\boldsymbol{p}$, $\boldsymbol{c}_i$, and $\boldsymbol{r}_i$ are the position vectors while $\boldsymbol{f}_i$ is the GRF exerted on the ith foot in frame $\mathcal{I}$. Note that the number of legs may vary.

and $\times$ is the vector cross product. The variables $\boldsymbol{a}$ and $\boldsymbol{b}$ will be used frequently later to illustrate other unusual operators.

The continuous-time dynamics (1) is further discretized so as to make the problem finite-dimensional, using the forward Euler method as follows:

$$\boldsymbol{p}_{k+1} = \boldsymbol{p}_k + \boldsymbol{v}_k\Delta t + \frac{1}{2}\dot{\boldsymbol{v}}_k\Delta t^2, \tag{3a}$$

$$\boldsymbol{v}_{k+1} = \boldsymbol{v}_k + \dot{\boldsymbol{v}}_k\Delta t, \tag{3b}$$

$$\boldsymbol{R}_{k+1} = \boldsymbol{R}_k \exp\left(\widehat{\boldsymbol{\omega}}_k\Delta t\right), \tag{3c}$$

$$\boldsymbol{\omega}_{k+1} = \boldsymbol{\omega}_k + \dot{\boldsymbol{\omega}}_k\Delta t, \tag{3d}$$

where Δt is the sampling period, $k \in \mathbb{N}$ is the index, and $\exp(\cdot)$ is the matrix exponential operator from $\mathfrak{so}(3)$ to $\mathsf{SO}(3)$, also known as Rodrigues' rotation formula:

$$\exp\left(\widehat{\boldsymbol{a}}\right) = \mathbb{I} + \frac{\sin\|\boldsymbol{a}\|}{\|\boldsymbol{a}\|}\widehat{\boldsymbol{a}} + \frac{1-\cos\|\boldsymbol{a}\|}{\|\boldsymbol{a}\|^2}\widehat{\boldsymbol{a}}^2, \tag{4}$$

where $\|\cdot\|$ denotes the Euclidean norm, e.g., $\|\boldsymbol{a}\| = \sqrt{\boldsymbol{a}^\top\boldsymbol{a}}$, and $\mathbb{I} \in \mathbb{R}^{3\times 3}$ is the identity matrix. It also reduces to $\mathbb{I}$ for $\|\boldsymbol{a}\| = 0$. Note that $\boldsymbol{\omega}$ is assumed to remain constant over the sampling period to derive (3c).

B. Parametrization of Rotation Matrix

Contrary to the Euclidean spaces, e.g., $\boldsymbol{p}$, $\boldsymbol{v}$, $\boldsymbol{\omega} \in \mathbb{R}^3$, it is problematic to directly involve the rotation matrix $\boldsymbol{R} \in \mathsf{SO}(3)$ in a mathematical optimization. Working directly on $\boldsymbol{R}$ leads to an overparametrization of the problem, e.g., we parametrize it with nine elements while it is essentially defined by some vector in $\mathbb{R}^3$, which can make the normal equations underdetermined in the end [25]. In our proposed MPC framework, we parametrize the rotation matrix using the history of angular velocity. Iterating (3c) for all the sampling periods until the first time step we obtain

$$\boldsymbol{R}_{k+1} = \underbrace{\boldsymbol{R}_{k-1}\exp\left(\widehat{\boldsymbol{\omega}}_{k-1}\Delta t\right)}_{=\boldsymbol{R}_k}\exp\left(\widehat{\boldsymbol{\omega}}_k\Delta t\right) \tag{5a}$$

$$\vdots$$

$$= \underset{\underset{\boldsymbol{R}_c}{\|}}{\boldsymbol{R}_1}\prod_{j=1}^{k}\exp\left(\widehat{\boldsymbol{\omega}}_j\Delta t\right). \tag{5b}$$

Skew-symmetric matrices generally do not commute with each other and thus their matrix exponentials cannot be combined, e.g., $\exp(\widehat{\boldsymbol{a}})\exp(\widehat{\boldsymbol{b}}) \neq \exp(\widehat{\boldsymbol{a}}+\widehat{\boldsymbol{b}})$. With (5b), the rotation matrix at the $(k+1)$th time step is parametrized with the history of angular velocity from the first time step to the kth time step. Note that the rotation matrix at the first time step $\boldsymbol{R}_1$ in the MPC framework is considered equal to the current body orientation $\boldsymbol{R}_c$, i.e., $\boldsymbol{R}_1 = \boldsymbol{R}_c$ is known, which will be updated for each MPC iteration and can be presumably measured or estimated.

C. Linearization of Rotational Dynamics

While the translational parts, i.e., position and velocity, evolve linearly, the rotational parts, i.e., orientation and angular velocity, are nonlinear in (3). Although recent works have addressed nonlinear MPC, the nonlinearity inevitably complicates the solving process, not to mention the problematic initial guess and local minima issues. As a consequence, we intend to linearize the nonlinear rotational dynamics.

Assuming the variation in body orientation is small over the MPC prediction horizon, thus applying the power series of matrix exponential operator to (5b), and only keeping the terms up to the first order, we obtain

$$\boldsymbol{R}_{k+1} = \boldsymbol{R}_c \prod_{j=1}^{k} \sum_{l=0}^{\infty} \frac{(\widehat{\boldsymbol{\omega}}_j \Delta t)^l}{l!} \approx \boldsymbol{R}_c\Big(\mathbb{I} + \Delta t \sum_{j=1}^{k} \widehat{\boldsymbol{\omega}}_j\Big), \quad (6)$$

which is now linear with respect to the history of $\boldsymbol{\omega}$.

The nonlinearity of the angular velocity dynamics results from the bilinear terms in (1d). Assuming $\boldsymbol{R}$ and $\boldsymbol{r}_i$ will not change substantially under well-controlled motion, we can thus consider them as constants, i.e., $\boldsymbol{R}_k = \boldsymbol{R}_c$ and $\boldsymbol{r}_{i,k} = \boldsymbol{r}_{i,c}$, for a short time over the MPC prediction horizon, which will be updated for each MPC iteration and can be accessed as well. Note that the robot model will always be correct for the first time step if the current measurements are close to the actual values. The MPC can also execute at a sufficiently high frequency, thus preventing it from divergence due to this rough approximation. Finally, the bilinear term representing the change in angular momentum due to inertia tensor rotation is also optimally linearized around the current body angular velocity $\boldsymbol{\omega}_c$ as follows:

$$\boldsymbol{\omega}_k \times \boldsymbol{I}\boldsymbol{\omega}_k \approx \boldsymbol{\omega}_c \times \boldsymbol{I}\boldsymbol{\omega}_c + \boldsymbol{\Lambda}_c^* (\boldsymbol{\omega}_k - \boldsymbol{\omega}_c), \quad (7)$$

using OLQP [31], where $\boldsymbol{\Lambda}_c^*$ is the optimal linear gain matrix. Note that OLQP linear approximation works for a larger region of interest than the conventional Jacobian linearization method. $\boldsymbol{\Lambda}_c^*$ also converges to $\widehat{\boldsymbol{\omega}}_c \boldsymbol{I} - \widehat{\boldsymbol{I}\boldsymbol{\omega}_c}$ as the region of interest gets smaller and smaller around $\boldsymbol{\omega}_c$, which coincides with the Jacobian matrix at $\boldsymbol{\omega}_c$.

D. State-Space Formulation

Let us define the state vector and control input at the kth time step as follows:

$$\boldsymbol{x}_k := \left[\boldsymbol{p}_k^\top, \boldsymbol{v}_k^\top, \boldsymbol{\omega}_k^\top\right]^\top \in \mathbb{R}^9, \quad (8a)$$

$$\boldsymbol{u}_k := \left[\boldsymbol{f}_{1,k}^\top, \boldsymbol{f}_{2,k}^\top, \dots, \boldsymbol{f}_{n,k}^\top\right]^\top \in \mathbb{R}^{3n}, \quad (8b)$$

where the GRFs are considered to be the direct control inputs to the system. Note that the orientation configuration is excluded from the system since we can easily paramtrize the rotation matrix either using (5b) or (6), which would further contribute to the decrease in the number of decision variables in the MPC framework later.

By rearranging the corresponding equations, the linearized discrete-time dynamic equations of motion of the SRB model can be described in the state-space form as follows:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_c \boldsymbol{x}_k + \boldsymbol{B}_c \boldsymbol{u}_k + \boldsymbol{d}_c, \quad (9a)$$

where the constant matrices

$$\boldsymbol{A}_c = \begin{bmatrix} \mathbb{I} & \Delta t\,\mathbb{I} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbb{I} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbb{I} - \Delta t \boldsymbol{I}^{-1}\boldsymbol{\Lambda}_c^* \end{bmatrix} \in \mathbb{R}^{9\times9}, \quad (9b)$$

$$\boldsymbol{B}_c = \begin{bmatrix} (0.5\Delta t^2/m)\cdot \mathbf{1}_{1\times n} \otimes \mathbb{I} \\ (\Delta t/m)\cdot \mathbf{1}_{1\times n} \otimes \mathbb{I} \\ \Delta t \boldsymbol{I}^{-1}\boldsymbol{R}_c^\top \left[\, \widehat{\boldsymbol{r}}_{1,c} \ \cdots \ \widehat{\boldsymbol{r}}_{n,c} \,\right] \end{bmatrix} \in \mathbb{R}^{9\times 3n}, \quad (9c)$$

$$\boldsymbol{d}_c = \begin{bmatrix} 0.5\Delta t^2 \boldsymbol{g} \\ \Delta t \boldsymbol{g} \\ \Delta t \boldsymbol{I}^{-1}\left(\boldsymbol{\Lambda}_c^* - \widehat{\boldsymbol{\omega}}_c \boldsymbol{I}\right)\boldsymbol{\omega}_c \end{bmatrix} \in \mathbb{R}^9, \quad (9d)$$

$\mathbf{0}_{3\times3} \in \mathbb{R}^{3\times3}$ is a matrix of zeros, $\mathbf{1}_{1\times n} \in \mathbb{R}^{1\times n}$ is a row vector of ones, and $\otimes$ is the Kronecker product for simple notation. Note that the subscript c in the matrix notations $\boldsymbol{A}_c$, $\boldsymbol{B}_c$, and $\boldsymbol{d}_c$ indicates they need to be updated by the current measurements for each MPC iteration. To sum up, the nonlinear dynamics (3) is linearly approximated around the operating point, which results in a linear time-invariant system and this system is assumed to be valid for a short period over the MPC prediction horizon.

III. Convex Model Predictive Control

This section illustrates the proposed convex MPC framework of SRB model on $\mathsf{SO}(3)$. The robot reference tracking problem is formulated as a mathematical TO problem. Starting from the current state, the goal is to determine an optimal control strategy over a finite time horizon while satisfying the various physical constraints, so as to guide the robot along the reference trajectory. This TO problem can be further transcribed into a standard QP in the form of

$$\underset{\boldsymbol{z}}{\text{minimize}} \quad \frac{1}{2}\boldsymbol{z}^T \boldsymbol{P} \boldsymbol{z} + \boldsymbol{q}^T \boldsymbol{z} \quad (10a)$$

$$\text{subject to} \quad \boldsymbol{A}_{ineq}\boldsymbol{z} \preceq \boldsymbol{b}_{ineq} \quad (10b)$$

$$\boldsymbol{A}_{eq}\boldsymbol{z} = \boldsymbol{b}_{eq}, \quad (10c)$$

where $\boldsymbol{z}$ is the vector of the decision variables, the constant vector $\boldsymbol{q}$ has the same length as $\boldsymbol{z}$, the constant matrix $\boldsymbol{P}$ is symmetric positive semi-definite, and (10b) and (10c) captures all the inequality and equality constraints, respectively.

A. Decision Variables

Given the prediction horizon T as well as the number of total time steps $N = 1 + T/\Delta t$, let us define the vector of decision variables as follows:

$$\boldsymbol{z} := \left[\boldsymbol{X}^\top, \boldsymbol{U}^\top\right]^\top \in \mathbb{R}^{9(N-1)+3n(N-1)}, \quad (11a)$$

where the variables

$$\boldsymbol{X} := \left[\boldsymbol{x}_2^\top, \boldsymbol{x}_3^\top, \ldots, \boldsymbol{x}_N^\top\right]^\top \in \mathbb{R}^{9(N-1)}, \tag{11b}$$

$$\boldsymbol{U} := \left[\boldsymbol{u}_1^\top, \boldsymbol{u}_2^\top, \ldots, \boldsymbol{u}_{N-1}^\top\right]^\top \in \mathbb{R}^{3n(N-1)}, \tag{11c}$$

collect the state vectors up to the Nth time step and control inputs up to the $(N-1)$th time step, respectively. Note that the state vector at the first time step $\boldsymbol{x}_1$ is set equal to the current state $\boldsymbol{x}_c$ from measurements, i.e., $\boldsymbol{x}_1 = \boldsymbol{x}_c$ is known, and thus does not need to be included for simplicity.

B. Cost Function

For robot reference tracking problem, the very common quadratic function

$$J = \frac{1}{2}\sum_{k=2}^{N} \|\boldsymbol{e}_{p,k}\|^2_{\boldsymbol{Q}_{p,k}} + \|\boldsymbol{e}_{v,k}\|^2_{\boldsymbol{Q}_{v,k}} + \|\boldsymbol{e}_{R,k}\|^2_{\boldsymbol{Q}_{R,k}} + \|\boldsymbol{e}_{\omega,k}\|^2_{\boldsymbol{Q}_{\omega,k}} + \|\boldsymbol{u}_{k-1}\|^2_{\boldsymbol{Q}_{u,k-1}} \tag{12a}$$

is used, where the error functions are given by

$$\boldsymbol{e}_{p,k} = \boldsymbol{p}_k - \boldsymbol{p}_{d,k}, \tag{12b}$$

$$\boldsymbol{e}_{v,k} = \boldsymbol{v}_k - \boldsymbol{v}_{d,k}, \tag{12c}$$

$$\boldsymbol{e}_{R,k} = \log\left(\boldsymbol{R}_{d,k}^\top \boldsymbol{R}_k\right)^\vee, \tag{12d}$$

$$\boldsymbol{e}_{\omega,k} = \boldsymbol{\omega}_k - \boldsymbol{R}_k^\top \boldsymbol{R}_{d,k}\boldsymbol{\omega}_{d,k}, \tag{12e}$$

with the weighted vector norm square $\|\boldsymbol{e}\|^2_{\boldsymbol{Q}} := \boldsymbol{e}^\top \boldsymbol{Q}\boldsymbol{e}$. Note that $\boldsymbol{Q}_{p,k}$, $\boldsymbol{Q}_{v,k}$, $\boldsymbol{Q}_{\omega,k}$, $\boldsymbol{Q}_{R,k} \in \mathbb{S}^3_+$, and $\boldsymbol{Q}_{u,k-1} \in \mathbb{S}^{3n}_+$ are the diagonal positive semi-definite weighting matrices, while $\boldsymbol{p}_{d,k}$, $\boldsymbol{v}_{d,k}$, $\boldsymbol{\omega}_{d,k} \in \mathbb{R}^3$, and $\boldsymbol{R}_{d,k} \in \mathsf{SO}(3)$ are the corresponding desired references. Therefore, J will be minimized in terms of overall tracking errors and control efforts in the least-squares sense. Note that the matrix logarithm operator $\log(\cdot)$ from $\mathsf{SO}(3)$ to $\mathfrak{so}(3)$, which is also the inverse of the matrix exponential operator, is given by

$$\log(\boldsymbol{R}) = \frac{\theta}{2\sin\theta}\left(\boldsymbol{R} - \boldsymbol{R}^\top\right), \tag{13}$$

where $\theta = \arccos\left((\mathrm{tr}(\boldsymbol{R}) - 1)/2\right)$ and $\mathrm{tr}(\cdot)$ calculates the trace of a square matrix. It also reduces to $\boldsymbol{0}_{3\times3}$ for $\boldsymbol{R} = \mathbb{I}$. The vee operator $\cdot^\vee$ from $\mathfrak{so}(3)$ to $\mathbb{R}^3$ is the inverse of the hat operator, e.g., $\widehat{\boldsymbol{a}}^\vee = \boldsymbol{a}$, which extracts the vector from the corresponding skew-symmetric matrix.

The error functions for the rotation matrix $\boldsymbol{e}_R$ and angular velocity $\boldsymbol{e}_\omega$ are proposed in [24]. Both of them are functions with respect to $\boldsymbol{R}$, which need to be reparametrized with the decision variables (11). For $\boldsymbol{e}_R$, substituting (5b) at the kth time step into (12d) yields

$$\boldsymbol{e}_{R,k} \overset{(5b)}{=} \log\left(\boldsymbol{R}_{d,k}^\top \boldsymbol{R}_c \prod_{j=1}^{k-1}\exp\left(\widehat{\boldsymbol{\omega}}_j \Delta t\right)\right)^\vee \tag{14a}$$

$$\approx \underbrace{\log\left(\boldsymbol{R}_{d,k}^\top \boldsymbol{R}_c\right)^\vee}_{\boldsymbol{\xi}_{cd,k}} + \boldsymbol{J}_r^{-1}\left(\boldsymbol{\xi}_{cd,k}\right)\log\left(\prod_{j=1}^{k-1}\exp\left(\widehat{\boldsymbol{\omega}}_j \Delta t\right)\right)^\vee \tag{14b}$$

$$\approx \boldsymbol{\xi}_{cd,k} + \Delta t \boldsymbol{J}_r^{-1}\left(\boldsymbol{\xi}_{cd,k}\right)\sum_{j=1}^{k-1}\boldsymbol{\omega}_j, \tag{14c}$$

where the inverse of the right Jacobian matrix for exponential coordinates, given by

$$\boldsymbol{J}_r^{-1}(\boldsymbol{a}) = \mathbb{I} + \frac{\widehat{\boldsymbol{a}}}{2} + \left(\frac{1}{\|\boldsymbol{a}\|^2} - \frac{1 + \cos\|\boldsymbol{a}\|}{2\|\boldsymbol{a}\|\sin\|\boldsymbol{a}\|}\right)\widehat{\boldsymbol{a}}^2, \tag{15}$$

relates the small variation between $\mathsf{SO}(3)$ and $\mathfrak{so}(3)$ [32]. It also reduces to $\mathbb{I}$ for $\|\boldsymbol{a}\| = 0$. Note that to get (14b) we have taken the first-order approximation for the logarithm operator [25], e.g., if $\boldsymbol{b}$ is assumed small, we have

$$\log\left(\exp\left(\widehat{\boldsymbol{a}}\right)\exp\left(\widehat{\boldsymbol{b}}\right)\right)^\vee \approx \boldsymbol{a} + \boldsymbol{J}_r^{-1}(\boldsymbol{a})\,\boldsymbol{b}. \tag{16}$$

A similar first-order approximation is also applied to the second logarithm in (14b) while all the resulting right Jacobian matrices can be reduced to $\mathbb{I}$ under the assumption of small variation in rotation matrix, as stated previously. (14c) is now linear with respect to the history of $\boldsymbol{\omega}$ and the subscript cd of $\boldsymbol{\xi}_{cd}$ indicates that it needs to be updated by the current measurements as well as the desired references for each MPC iteration. As for $\boldsymbol{e}_\omega$, substituting (6) at the kth time step into (12e) we get

$$\boldsymbol{e}_{\omega,k} \overset{(6)}{\approx} \boldsymbol{\omega}_k - \left(\mathbb{I} + \Delta t\sum_{j=1}^{k-1}\widehat{\boldsymbol{\omega}}_j\right)^\top \boldsymbol{R}_c^\top \boldsymbol{R}_{d,k}\boldsymbol{\omega}_{d,k} \tag{17a}$$

$$= \boldsymbol{\omega}_k - \left(\mathbb{I} - \Delta t\sum_{j=1}^{k-1}\widehat{\boldsymbol{\omega}}_j\right)\underbrace{\boldsymbol{R}_c^\top \boldsymbol{R}_{d,k}\boldsymbol{\omega}_{d,k}}_{\boldsymbol{\eta}_{cd,k}} \tag{17b}$$

$$= \boldsymbol{\omega}_k - \Delta t\widehat{\boldsymbol{\eta}}_{cd,k}\sum_{j=1}^{k-1}\boldsymbol{\omega}_j - \boldsymbol{\eta}_{cd,k}, \tag{17c}$$

which is also linear with respect to the history of $\boldsymbol{\omega}$. Note that we have applied the properties of skew-symmetric matrices $\widehat{\boldsymbol{a}}^\top = -\widehat{\boldsymbol{a}}$ to get (17b) and $\widehat{\boldsymbol{a}}\boldsymbol{b} = -\widehat{\boldsymbol{b}}\boldsymbol{a}$ to get (17c), and $\boldsymbol{\eta}_{cd}$ needs to be updated for each MPC iteration as well.

Now that all the error functions (12b), (12c), (14c), and (17c) are linear with respect to the decision variables (11). After some calculations and rearrangements, the cost function J of (12a) can be formulated into (10a) without considering the trivial constant term.

C. Constraints

Several constraints need to be imposed in order to satisfy the physical requirements, which are all linear in terms of the decision variables (11) and thus can be formulated into (10b) and (10c).

1) Robot Dynamics Constraint: The evolution of the robot states needs to respect the system dynamics (9) subject to the control inputs at each time step. We thus have

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_c\boldsymbol{x}_k + \boldsymbol{B}_c\boldsymbol{u}_k + \boldsymbol{d}_c \tag{18}$$

for $k = 1, \ldots, N-1$. Note that we also have $\boldsymbol{x}_1 = \boldsymbol{x}_c$.

2) Ground Reaction Force Constraint: When the ith leg is in stance phase, the normal component of its GRF f_i^z should first be nonnegative, which indicates

$$f_{i,k}^z \geq 0, \tag{19a}$$

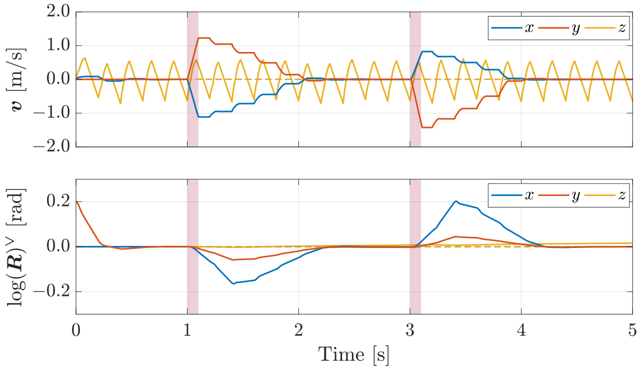

Fig. 2. Simulation results of the monopod robot for the first five seconds. The figure shows the time history of CoM velocity and body orientation in each direction. The shaded areas indicate the duration of the disturbances and the dashed lines indicate the corresponding desired references.

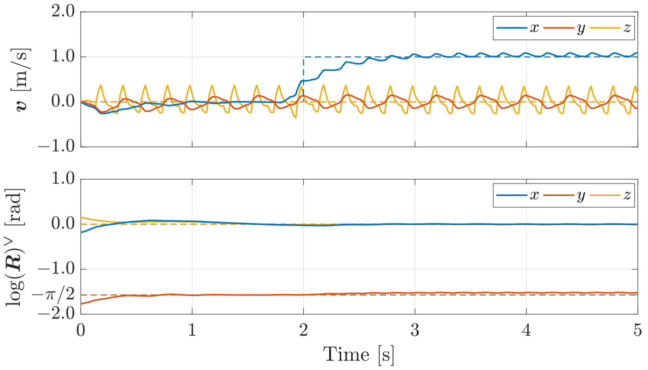

Fig. 3. Simulation results of the biped robot for the first five seconds. The figure shows the time history of CoM velocity and body orientation in each direction.

as well as upper bounded so as to avoid actuator overtorque, which implies

$$f_{i,k}^z \leq f_{i,\max}^z. \tag{19b}$$

In addition, to prevent slippage, the GRF should lie within the local friction cone. The friction cone is linearly approximated by a square pyramid [33], which gives

$$\left|f_{i,k}^{x/y}\right| \leq \mu f_{i,k}^z \Rightarrow -\mu f_{i,k}^z \leq f_{i,k}^{x/y} \leq \mu f_{i,k}^z, \tag{19c}$$

where μ is the ground friction coefficient. Note that (19a) is already embedded in (19c) and thus can be removed from the constraints. On the contrary, if the leg is in swing phase, we need to impose its GRF

$$\boldsymbol{f}_{i,k} = \boldsymbol{0}, \tag{19d}$$

or simply set $f_{i,\max}^z$ to zero, which ensures zero contribution to the system dynamics when the leg is off the ground.

IV. Simulation Results

The proposed convex MPC framework of SRB model on $\mathsf{SO}(3)$ was implemented on various legged robots with different numbers of legs. Each leg was assumed massless with a conventional three-degree-of-freedom configuration and a point foot. The original nonlinear dynamic model (1) was simulated using MATLAB's *ode45* function with $m = 5.0$ kg, $\boldsymbol{I} = \mathbf{diag}\,(0.026, 0.047, 0.054)$ kg·m^2, and $\boldsymbol{g} = [0, 0, -9.81]^\top$ m/s^2.

The computation speed of the QP (10) depends on the number of legs n as well as the number of time steps N. For a quadruped robot with $n = 4$ and $N = 6$ ($\Delta t = 0.05$ s), the processing time including problem formulation can achieve a frequency of 800 Hz with the off-the-shelf solver OSQP [34] on a laptop with an AMD Ryzen 5 4500U CPU at 2.1 GHz, which is sufficient for real-time feedback control. The optimal solution of the control input at the first time step $\boldsymbol{u}_1^*$ was used directly on the robot. Several simulations were tested to evaluate the MPC performance, demonstrated in the video attachment. Note that the legs were only kinematically modeled in the simulation for the sake of better visualization.

For all the dynamic locomotion tests, the desired translational reference was always set in the form of

$$\boldsymbol{p}_d(t) = \left[\int_0^t v_d^x(\tau)\mathrm{d}\tau, \int_0^t v_d^y(\tau)\mathrm{d}\tau, p_d^z\right]^\top, \tag{20a}$$

$$\boldsymbol{v}_d(t) = \left[v_d^x(t), v_d^y(t), 0\right]^\top. \tag{20b}$$

That is, the robot was desired to move freely in the horizontal plane while being constrained at some constant height. The desired orientation reference might vary. The desired foot placement for each swing leg was determined at touch-down using the linear combination of Raibert [35] heuristics and a capture point [36] based feedback term, similar to the technique used in [22].

A. Monopod Robot

The only feasible locomotion gait of a monopod robot is hopping. The proposed convex MPC framework was implemented on a monopod with a constant desired orientation reference $\boldsymbol{R}_d(t) = \mathbb{I}$ and $\boldsymbol{\omega}_d(t) = \left[0, 0, 0\right]^\top$. A time-based gait pattern was executed with $T_{st} = T_{sw} = 0.1$ s, where T_{st} and T_{sw} are the nominal durations of stance and swing phases, respectively. Fig. 2 shows the simulation results for the first five seconds. The monopod was released slightly out of balance but was able to quickly stabilize itself by adjusting its footstep location. To gauge the overall system robustness in terms of disturbance rejection, two random external forces with a duration of 0.1 s and a magnitude of 80 N were then exerted on the monopod but the robot managed to recover. Afterwards, the monopod was commanded to follow some predefined velocity reference and ended up with a satisfying tracking performance, as shown in the video supplement.

B. Biped Robot

The proposed convex MPC framework was implemented on a biped robot then. For locomotion, a desired orientation reference was specified in the form of

$$\boldsymbol{R}_d(t) = \boldsymbol{R}_y\left(-\frac{\pi}{2}\right)\boldsymbol{R}_x\left(\int_0^t \omega_d^x(\tau)\mathrm{d}\tau\right), \tag{21a}$$

$$\boldsymbol{\omega}_d(t) = \left[\omega_d^x(t), 0, 0\right]^\top, \tag{21b}$$

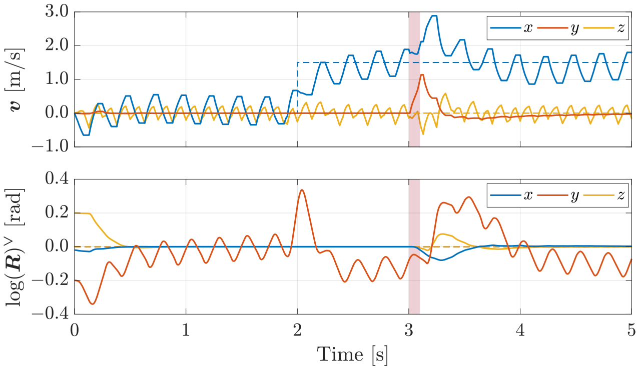

Fig. 4. Simulation results of the quadruped robot during bounding for the first five seconds. The figure shows the time history of CoM velocity and body orientation in each direction.

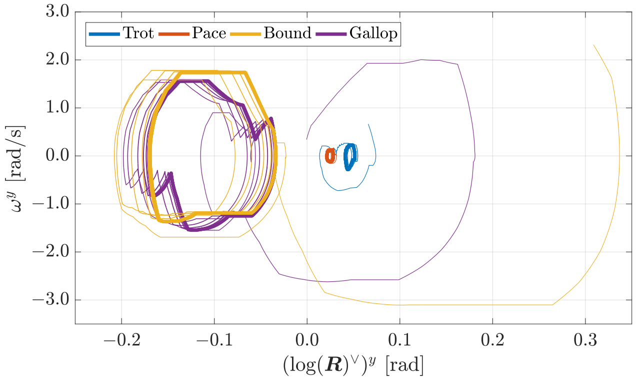

Fig. 5. Phase portraits of the body orientation and angular velocity in the y direction with a constant desired forward velocity of 1.5 m/s and without external disturbances.

where $\boldsymbol{R}_x(\cdot)$ and $\boldsymbol{R}_y(\cdot)$ are the basic rotation matrices about the x and $y-$axes, respectively. Note that the robot body was intentionally rotated to the vertical plane. This is usually the case where some conventional MPC frameworks fail using the Euler angles representation as the singularity occurs, which however can still be handled with our approach. The gait pattern was still time-based with $T_{st} = T_{sw} = 0.2$ s and without considering a double support phase. Fig. 3 shows the simulation results for the first five seconds. The biped robot was able to quickly stabilize to the steady state at the beginning from some random initial condition. From $t = 2.0$ s, it was commanded to track a forward velocity of 1.0 m/s. Note that the MPC framework enabled the robot to speed up before the sudden reference change so as to have an overall better tracking performance. Afterwards, as shown in the video, the biped robot managed to change the facing direction while still moving forward. By further adding an aerial phase, the biped robot could also achieve a stable running motion. In the end, the robot was commanded to perform a hopping motion and even backflip. The hopping motion was naturally achieved by gradually synchronizing the two leg movements, while the backflip was further realized by carefully choosing the desired references. After landing, the robot still managed to stabilize and keep its balance by adjusting its foot placements using a nominal walking gait with zero velocity reference.

C. Quadruped Robot

The proposed convex MPC framework was implemented on a quadruped robot finally. It was verified to be able to stabilize several typical quadrupedal locomotion gaits, including trot, pace, bound, and gallop. Note that the same cost function was used across all cases with a constant desired orientation reference $\boldsymbol{R}_d(t) = \mathbb{I}$ and $\boldsymbol{\omega}_d(t) = \left[0, 0, 0\right]^\top$, and only the time-based gait pattern needed to be modified accordingly. For all the gait tests, the robot started with the same random initial configuration but was able to quickly converge to the steady state. From $t = 2.0$ s, it was commanded to track a forward velocity of 1.5 m/s. Two external forces with a duration of 0.1 s and a magnitude of 100 N were then randomly exerted on the quadruped body and the robot managed to resist them. Fig. 4 shows the simulation results of the bound gait with $T_{st} = 0.1$ s and $T_{sw} = 0.15$ s for the first five seconds. Fig. 5 further shows the phase portrait of the body orientation and angular velocity in the y direction, which demonstrates the convergence to some limit cycle attractor for each gait. More information can be viewed in the supplementary video. At the last, we kindly note that trivial reference trajectories are sufficient to stabilize a wide range of dynamic legged motions, which illustrates the great advantage of the proposed MPC framework.

V. Conclusion

In this work, a novel convex model predictive control (MPC) framework of single rigid body (SRB) model on $\mathsf{SO}(3)$ is proposed for versatile dynamic legged motions. Specifically, it exploits the linearized SRB model around the operating point, which is verified valid for a short period over the prediction horizon. With ground reaction forces as direct control inputs to the system, no reference control trajectory needs to be specified ahead of time. By working on the rotation matrix, issues arising from other representations such as singularity of Euler angles and unwinding phenomenon of quaternion can be avoided. Moreover, it has a compact structure since we are able to parametrize the rotation matrix using the history of angular velocity without introducing additional variables. The robot reference tracking trajectory optimization problem is then formulated into a standard quadratic program, which can be solved efficiently to meet the real-time constraint. The proposed convex MPC framework is versatile since it is applicable to various legged robots with different numbers of legs for performing different kinds of dynamic motions, as shown in the simulation environment. Strong robustness is also justified in terms of external disturbance rejection for robots in operation. We thus envision a promising future of the proposed convex MPC framework in legged robots and potentially even in other applications as well. Future works will focus on how to properly involve the leg dynamics when it is not negligible as well as implementations on the real hardware platforms.

References

[1] J. Pratt, C.-M. Chew, A. Torres, P. Dilworth, and G. Pratt, "Virtual model control: An intuitive approach for bipedal locomotion," *The International Journal of Robotics Research*, vol. 20, no. 2, pp. 129–143, 2001.

[2] 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)*, pp. 3346–3351, 2015.

[3] M. Neunert, M. Stäuble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli, "Whole-body nonlinear model predictive control through contacts for quadrupeds," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1458–1465, 2018.

[4] 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, no. 1, pp. 69–81, 2014.

[5] X. Lin, J. Zhang, J. Shen, G. Fernandez, and D. W. Hong, "Optimization based motion planning for multi-limbed vertical climbing robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1918–1925, 2019.

[6] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard, "Crocoddyl: An efficient and versatile framework for multi-contact optimal control," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2536–2542, 2020.

[7] J. Shen, Y. Liu, X. Zhang, and D. Hong, "Optimized jumping of an articulated robotic leg," in *2020 17th International Conference on Ubiquitous Robots (UR)*, pp. 205–212, 2020.

[8] J. Zhang, X. Lin, and D. W. Hong, "Transition motion planning for multi-limbed vertical climbing robots using complementarity constraints," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2033–2039, 2021.

[9] R. Full and D. Koditschek, "Templates and anchors: neuromechanical hypotheses of legged locomotion on land," *Journal of Experimental Biology*, vol. 202, no. 23, pp. 3325–3332, 1999.

[10] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *2003 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1620–1626, 2003.

[11] P. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *2006 6th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 137–142, 2006.

[12] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online Walking Motion Generation with Automatic Foot Step Placement," *Advanced Robotics*, vol. 24, no. 5-6, pp. 719–737, 2010.

[13] 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 (Humanoids)*, pp. 295–302, 2014.

[14] S. Kuindersma, R. Deits, M. F. Fallon, A. K. 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, 2016.

[15] Y. Liu, J. Shen, J. Zhang, X. Zhang, T. Zhu, and D. Hong, "Design and control of a miniature bipedal robot with proprioceptive actuation for dynamic behaviors," in *2022 IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[16] G. Bledt, P. M. Wensing, and S. Kim, "Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the mit cheetah," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4102–4109, 2017.

[17] 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)*, pp. 1–9, 2018.

[18] 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, no. 3, pp. 1560–1567, 2018.

[19] J. R. Hooks, *Real-Time Optimization for Control of a Multi-Modal Legged Robotic System*. PhD thesis, UCLA, 2019.

[20] O. Villarreal, V. Barasuol, P. M. Wensing, D. G. Caldwell, and C. Semini, "Mpc-based controller with terrain insight for dynamic legged locomotion," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2436–2442, 2020.

[21] S. Hong, J. Kim, and H. Park, "Real-time constrained nonlinear model predictive control on so(3) for dynamic legged locomotion," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3982–3989, 2020.

[22] Y. Ding, A. Pandala, C. Li, Y. H. Shin, and H. W. Park, "Representation-free model predictive control for dynamic motions in quadrupeds," *IEEE Transactions on Robotics*, pp. 1–18, 2021.

[23] J. Shen and D. Hong, "A novel model predictive control framework using dynamic model decomposition applied to dynamic legged locomotion," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4926–4932, 2021.

[24] F. Bullo and R. M. Murray, "Proportional derivative (pd) control on the euclidean group," in *3rd European Control Conference*, pp. 1091–1097, 1995.

[25] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2017.

[26] M. Moakher, "Means and averaging in the group of rotations," *SIAM Journal on Matrix Analysis and Applications*, vol. 24, no. 1, p. 1–16, 2002.

[27] J. Stuelpnagel, "On the parametrization of the three-dimensional rotation group," *SIAM Review*, vol. 6, no. 4, pp. 422–430, 1964.

[28] S. P. Bhat and D. S. Bernstein, "A topological obstruction to continuous global stabilization of rotational motion and the unwinding phenomenon," *Systems & Control Letters*, vol. 39, no. 1, pp. 63 – 70, 2000.

[29] G. Wu and K. Sreenath, "Variation-based linearization of nonlinear systems evolving on $so(3)$ and $\mathbb{S}^2$," *IEEE Access*, vol. 3, pp. 1592–1604, 2015.

[30] M. Chignoli and P. M. Wensing, "Variational-based optimal control of underactuated balancing for dynamic quadrupeds," *IEEE Access*, vol. 8, pp. 49785–49797, 2020.

[31] J. Shen and D. Hong, "Optimal linearization via quadratic programming," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4572–4579, 2020.

[32] G. Chirikjian, *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*. Applied and Numerical Harmonic Analysis, Birkhäuser Boston, 2011.

[33] J. C. Trinkle, J.-S. Pang, S. Sudarsky, and G. Lo, "On dynamic multi-rigid-body contact problems with coulomb friction," *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 77, no. 4, pp. 267–279, 1997.

[34] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.

[35] M. H. Raibert, J. H. Benjamin Brown, and M. Chepponis, "Experiments in balance with a 3d one-legged hopping machine," *The International Journal of Robotics Research*, vol. 3, no. 2, pp. 75–92, 1984.

[36] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *2006 6th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 200–207, 2006.