2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Dynamic Locomotion of a Quadruped Robot with Active Spine via Model Predictive Control

Wanyue Li, Zida Zhou and Hui Cheng*

***Abstract*—As an active spine introduces more degree of freedoms (DOFs) as well as time-varying inertia, locomotion control of spined quadruped robots is challenging. Direct optimization on the full dynamics model causes prohibitive calculation time and is difficult to apply to embedded platforms. Model predictive control (MPC)-based on SRB dynamics is a prevalent approach for ordinary quadruped robots, regarding the whole robot as a single rigid body (SRB). However, the approach ignores the changes of the center of mass (CoM) and inertia, which seriously affects the robot's stability and could not be used in spined quadruped robots directly. To resolve the above issue, this paper presents an MPC approach that considers the movements of the spine in the SRB model. Since the mass of the robot is concentrated on its body, the whole robot is modelled as an unactuated SRB with fully-actuated internal spine joints. MPC finds the optimal ground reaction forces (GRFs) based on the SRB dynamics, in which the missing spine part is complemented by the pre-defined spine joints' states and corresponding inertia sequence. According to the GRFs, the full dynamic model calculates the precise joint torques. In addition, a quadruped robot with a 3-DOF active spine, Yat-sen Lion, is developed. With the presented approach, experimental results illustrate that Yat-sen Lion freely achieves bending, arching, and turning behaviors while trotting at speeds of 3.8 m/s in simulations and 0.5 m/s in real-world experiments.**

***Index Terms*—Legged Robots, Motion Control, Biologically-Inspired Robots**

I. INTRODUCTION

Many biological findings indicate that the active spine of quadruped plays an essential role in its locomotion [1], [2]. For example, the cheetah can reach a speed of 110 km/h by strenuously spinal flexion-extension movement. Inspired by quadrupeds, some robots with Early research took place in the 1990s when Leeser developed a planar spined-legged robot [3], and another spined quadruped robot BISAM [4] was developed. In 2012, Boston Dynamics launched a spined quadruped robot, Cheetah, reaching 29 kilometers per hour. However, related details of Cheetah have not been reported. In recent years, a series spined quadruped robots have emerged, such as Bobcat [5], Lynx [6], Inu [7] [8], SQBot [9], and Transleg robot [10]. These works show that the active spine joints benefit the robot's posture stabilization [11], dynamic gait performance [12], and energy utilization efficiency [13] [14] [15].

The authors are with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China. Corresponding Author: chengh9@mail.sysu.edu.cn

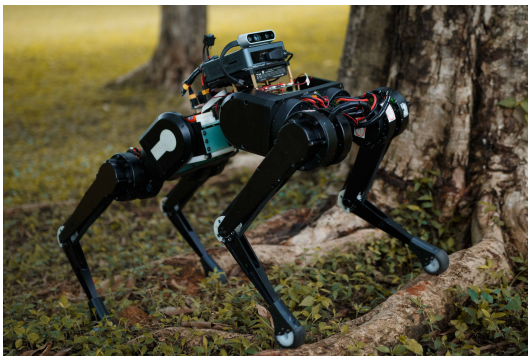


Fig. 1. The Yat-sen Lion quadruped robot platform has 15 actuated DoFs, three per leg and three in the spine.

In addition to underactuated body and dynamic constraints like ordinary quadruped robots, the active spine introduces more DoFs and time-varying inertia, making the control of highly dynamic quadruped robots challenging. In the existing works, the primary locomotion control methods are trajectory-based methods, such as CPG [16], which does not model the robot body or environment and lack the prediction of robot motion. Therefore, they are difficult to apply to control high-dynamic torque-controlled spined quadruped robots.

In recent years, MPC based on the SRB model has become increasingly popular [17] [18] [19] in the context of ordinary quadruped robots, as these methods effectively capture the main dynamics features of quadruped robots and operate in a predictive fashion. Applying the techniques of convex MPC to the SRB model, MIT Cheetah 3 can trot to a speed of 3.7 m/s and restore balance under a lateral impact of 1 m/s [20]. However, when it comes to spined quadruped robots, the spine movement causes significant changes in the whole robot's CoM and inertia. Previous methods typically regard the whole robot body as a time-invariant SRB, ignoring the changes of these parameters. As an MPC is highly dependent on the accurate parameters of the CoM position and inertia to realize satisfactory performance, applying previous methods to a spined quadruped robot typically results in instability, which makes the robot fall.

To solve the research gap, in this paper, we present an MPC approach that considers the movements of the spine in the SRB model. The whole spined quadruped robot is regarded as an SRB but has variable CoM and Inertia. The neglected

spine parts are complemented by pre-defined spine joints' state and corresponding inertia sequence. MPC predicts the optimal GRFs based on the SRB model, and then, the precise joint torques are calculated based on the full dynamic model. To the best of our knowledge, this is the first time that MPC based on SRB dynamics has been introduced into the spined quadruped robot. The spined quadruped can arbitrarily change its spinal posture at a high-speed motion for the first time.

The main contributions of this paper include:

- This paper presents a general model predictive control approach for high-dynamic torque-controlled spined quadruped robots, considering the effects of spine joint movements and variable inertia. The proposed control method supports the spined quadruped robot arbitrarily changing its spinal posture during high-speed motion.
- A 15 DoFs quadruped with three actuated spine joints, Yat-sen Lion, is developed and validated the proposed control method.

The rest of this paper is arranged as follows. Section II presents modelling of the spined quadruped robot. Section III introduces the mapping between the joint states and the system states. The proposed MPC approach is presented in Section IV. In Section V, experimental results verify the reliability of the proposed method. Finally, Section VI concludes this paper.

II. DYNAMICS MODELLING

The spined quadruped robot is a high DoF system, so direct optimization on the full dynamics model incurs prohibitively long calculation times and is difficult to apply to embedded platforms. For the whole robot, since the mass of the robot is concentrated on its body, we use the instantaneous SRB model, to capture the main features. Since the spine joints are fully actuated, it is reasonable to assume that the spine joints can track the desired state plan for a short period in the future. Therefore, we use the pre-defined spine joints' state sequence to complement the neglected spinal dynamics time evolution of the SRB model.

A. System Design

As Fig. 2 shows, because the floating-base model [21] is used, the robot has 15 active entity DoFs and 6 virtual DoFs. The robot is composed of 3 spine joints and 12 leg joints. These three spine joints are called chest, forelimb, and hindlimb, respectively. The robot base frame is fixed on the rigid body 5, on which the inertial measuring unit (IMU) is placed.

B. Full Dynamic Model

The joint-space dynamics equations depicted in Fig. 2 can be constructed as follow:

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\xi_j}\boldsymbol{\tau} + \boldsymbol{J}_c^T \boldsymbol{f}_r, \tag{1}$$

$$\boldsymbol{\xi_j} = \begin{bmatrix} \mathbf{0}_{6\times6} & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{n\times n} \end{bmatrix}, \tag{2}$$

where $\boldsymbol{H} \in \mathbb{R}^{(n+6)\times(n+6)}$, $\boldsymbol{C} \in \mathbb{R}^{n+6}$, $\boldsymbol{G} \in \mathbb{R}^{n+6}$, $\boldsymbol{\tau} \in \mathbb{R}^{n+6}$, $\boldsymbol{f}_r \in \mathbb{R}^{3n_c}$, $\boldsymbol{J}_c \in \mathbb{R}^{3n_c\times(n+6)}$ are the joint-space inertia matrix, bias force, gravitational term, joint generalized force, GRF, and contact Jacobian, respectively. $\boldsymbol{q} \in \mathbb{R}^{n+6}$, $\dot{\boldsymbol{q}} \in \mathbb{R}^{n+6}$, and $\ddot{\boldsymbol{q}} \in \mathbb{R}^{n+6}$ are vectors of joint generalized positions, velocities, and accelerations. $\boldsymbol{q} = (\boldsymbol{q}^u, \boldsymbol{q}^a)$, where the superscripts u and a correspond to the unactuated and actuated parts, respectively. n represents the DoFs of the actual joint, which is 15 in our robot. Since the virtual joints do not provide any torque, the selection matrix $\boldsymbol{\xi_j} \in \mathbb{R}^{(n+6)\times(n+6)}$ is used to remove all torque provided by virtual joints.

C. Single Body Dynamics Model

As Fig. 3 shows, by decoupling Eq. (1), the robot can be modelled as an unactuated SRB which has a fully-actuated spine (The leg masses are light compared to the body and so can be ignored). It is assumed that the actuators in the fully actuated spine have the bandwidth and torque necessary to track a desired state plan independent of the forces in the legs. Therefore, We can plan desired joint states for a short period in the future.

$$\boldsymbol{traj}_q^a = \{\boldsymbol{q}_1^a, ..., \boldsymbol{q}_t^a, ...\}, \boldsymbol{traj}_{\dot{q}}^a = \{\dot{\boldsymbol{q}}_1^a, ..., \dot{\boldsymbol{q}}_t^a, ...\}, \tag{3}$$

where $\boldsymbol{traj}_q^a$, $\boldsymbol{traj}_{\dot{q}}^a$ respectively represent the pre-defined sequence of actuated joint positions and velocities. The subscript t is the discrete-time index within the predict horizon. Then we can obtain the corresponding inertia $\bar{\boldsymbol{I}}_t \in \mathbb{R}^{3\times3}$ by

$$\bar{\boldsymbol{I}} = \mathcal{M}_I(\boldsymbol{q}^a), \tag{4}$$

where $\mathcal{M}_I$ represents a mapping that maps the joint states to the standard inertia $\bar{\boldsymbol{I}}$. The dynamics of the unactuated SRB are as follows :

$$m\ddot{\boldsymbol{P}}^c = \sum_{i=1}^{n_c} \boldsymbol{f}_i - \boldsymbol{G}, \tag{5}$$

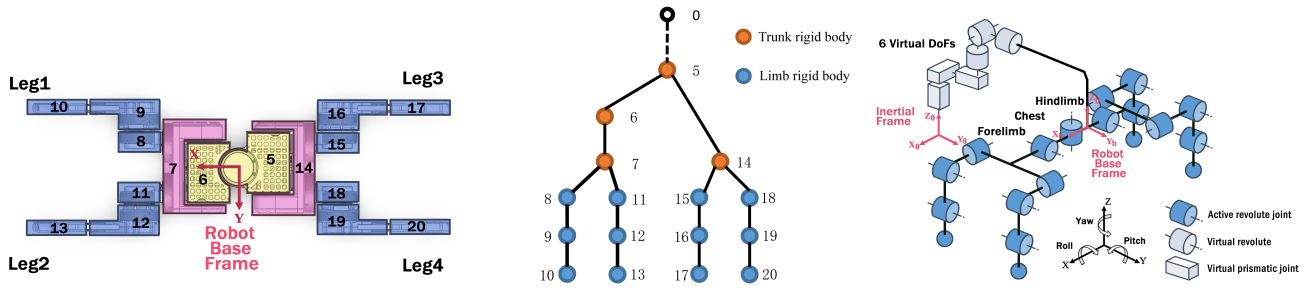
$$\frac{d}{dt}(\bar{\boldsymbol{I}}\,\boldsymbol{\omega}^b) = \sum_{i=1}^{n_c} (\boldsymbol{r}_i - \boldsymbol{P}^c) \times \boldsymbol{f}_i, \tag{6}$$

where $\boldsymbol{\omega}^b \in \mathbb{R}^3$, $\boldsymbol{P}^c \in \mathbb{R}^3$, and $\boldsymbol{G} \in \mathbb{R}^3$ represent the angular velocity of the robot base frame, the CoM position, and gravity, respectively. $\boldsymbol{f}_i \in \mathbb{R}^3$ and $\boldsymbol{r}_i \in \mathbb{R}^3$ represent the GRF and position of i-th contact point, respectively. m and n_c represent the robot's body mass and the number of feet in contact with the ground.

The system states is defined as follows:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{\Theta}^{bT} & \boldsymbol{P}^{cT} & \boldsymbol{\omega}^{bT} & \dot{\boldsymbol{P}}^{cT} \end{bmatrix}^T, \tag{7}$$

where $\boldsymbol{\Theta}^b = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T \in \mathbb{R}^3$ is the Z-Y-X Euler angles [22] of the robot base frame. ϕ, θ, and ψ represent the roll, pitch, and yaw, respectively. The rotational centroidal orientation frame of a multibody system is an open issue [23]



(a) (b) (c)

Fig. 2. Kinematic tree (a) and connectivity graph (b) of Yat-sen Lion. For (b), the orange and blue circles represent the rigid body of the trunk and limbs, respectively. The numbers next to the circles correspond to the numbers in (a) one-to-one. Dotted lines represent virtual connections. (c) shows the DoFs configuration of Yat-sen Lion. The inertial frame 0 connects to the base frame b of the robot through 6 unactuated virtual DoFs.

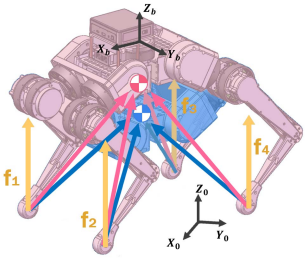


Fig. 3. The whole quadruped is regarded as an SRB at each instant. For example, when the quadruped's spine is arched, the SRB is the pink area, and the CoM is the pink point. When the spine is concaved, they are the blue parts. The yellow arrows indicate the GRFs, and the pink or blue arrows point to the CoM from the feet are the force arms of the GRFs.

[24]. Due to the assumption of instantaneous SRB, we simply adopt the floating base frame. According to the simplification in [17] that the roll and pitch angles are small, the angular velocity can be written as:

$$\dot{\boldsymbol{\Theta}}^b \approx \boldsymbol{R}_z(\psi)\boldsymbol{\omega}^b, \tag{8}$$

where $\boldsymbol{R}_z(\psi)$ represents a rotation of yaw about the Z-axis.

To accurately describe the instantaneous motion of the entire robot using the SRB model, besides calculating the real-time inertia in Eq. (4), we also capture the effect of the actuated joints' rate of change on the system states.

$$\boldsymbol{x} = \mathcal{M}_x\left(\boldsymbol{q}, \dot{\boldsymbol{q}}\right), \tag{9}$$

where $\mathcal{M}_x$ maps the joint states to system states. We will detail the mappings, Eq. (4) and Eq. (9), in III-B.

III. Calculation of System States and Inertia

We use spatial notations [25] to give the mapping between the joint space state and the system state. This mapping will be used for the following two aspects.

1) **Reference trajectory**: Unlike the ordinary quadruped robot, its CoM is located at the geometric center of the robot body. For the spined quadruped robot, we plan the trajectories in the joint space first and then map them into the corresponding system state.
2) **State estimation**: The results of state estimation are the states of each joint, and a mapping is also needed to convert them into the system states.

A. Spatial Inertia

Spatial inertia can be used to extract the CoM position and standard inertia and calculate the average velocity of the whole robot. So we describe it separately at first. The definition of the body i spatial inertia [25] is as follows:

$$\boldsymbol{I}_i = \begin{bmatrix} \bar{\boldsymbol{I}}_i - m_i[\boldsymbol{c}_i]_\times[\boldsymbol{c}_i]_\times & m_i[\boldsymbol{c}_i]_\times \\ -m_i[\boldsymbol{c}_i]_\times & m_i\mathbf{1}_{3\times3} \end{bmatrix}, \tag{10}$$

where $\boldsymbol{I}_i \in \mathbb{R}^{6\times6}$ is the spatial inertia of given rigid body i in *Plücker* coordinate systems. $\bar{\boldsymbol{I}}_i \in \mathbb{R}^{3\times3}$ is the standard inertia having origins at the CoM of body i. $\boldsymbol{c}_i \in \mathbb{R}^3$ is the body i CoM position expressed in frame i. m_i is the mass of body i. $[\boldsymbol{a}]_\times \in \mathbb{R}^3$ represents the kew-symmetric matrix, such that $[\boldsymbol{a}]_\times \boldsymbol{b} = \boldsymbol{a} \times \boldsymbol{b}$, for all $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^3$. $\mathbf{0}_{m\times n}$ and $\mathbf{1}_{m\times n}$ respectively represent m by n zero matrix and identity matrix.

In Fig. 2(b), the subtree rooted at rigid body i is treated as a single composite rigid body whose spatial inertia, $\boldsymbol{I}_i^c \in \mathbb{R}^{6\times6}$, is expressed as

$$\boldsymbol{I}_i^c = \boldsymbol{I}_i + \sum_{j\in\mu(i)} {}^i\boldsymbol{X}_j^{-T}\, \boldsymbol{I}_j^c\, {}^j\boldsymbol{X}_i, \tag{11}$$

$${}^i\boldsymbol{X}_j = \begin{bmatrix} {}^i\boldsymbol{R}_j & \mathbf{0} \\ [{}^i\boldsymbol{p}_j]_\times {}^i\boldsymbol{R}_j & {}^i\boldsymbol{R}_j \end{bmatrix}, \tag{12}$$

where $\mu(i)$ is the set of children of body i. ${}^i\boldsymbol{X}_j$ transforms spatial vectors from frame j to frame i. ${}^i\boldsymbol{R}_j$ transforms a vector in coordinate j to a vector expressed in coordinate i. ${}^i\boldsymbol{p}_j$ represents the position vector from the origin of frame i to the origin of frame j. In Eq. (11), since every rigid body's spatial inertia $\boldsymbol{I}_i$ is constant, the composite rigid body $\boldsymbol{I}_i^c$ is the function of joint configuration $\boldsymbol{q}$.

B. Mapping from Joint Space to System State Space

The subtree rooted at rigid body 5 includes all rigid bodies of the spined quadruped robot, and the frame of rigid body

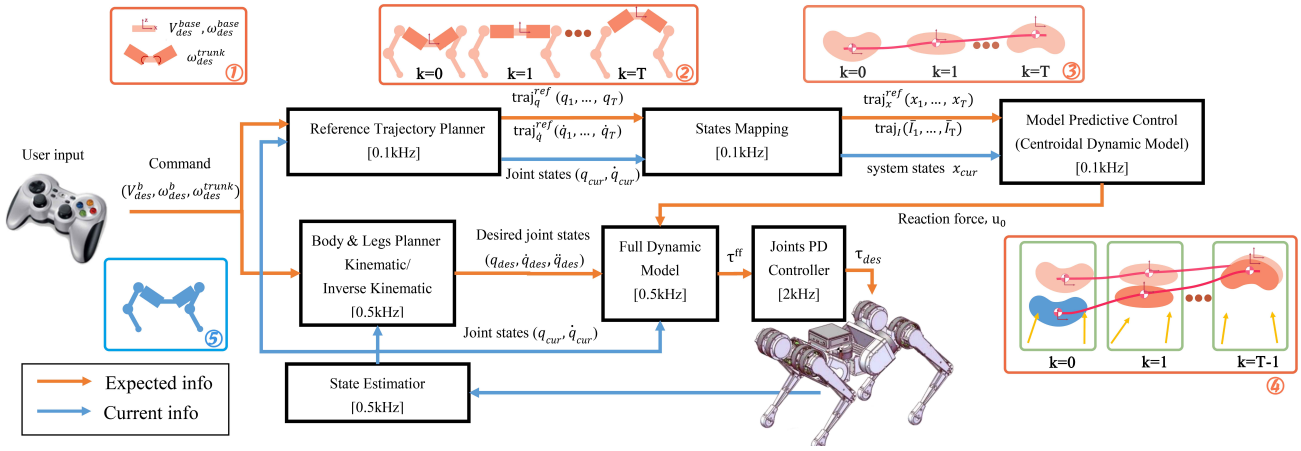


Fig. 4. Control framework. The robot operator sends commands to the robot (pic 1). According to the commands, the reference trajectory planner calculates a long-term reference trajectories of the robot joint states $\boldsymbol{traj}_q^{ref}$, $\boldsymbol{traj}_{\dot{q}}^{ref}$ (pic 2). The long-term reference trajectories are mapped to the system states $\boldsymbol{traj}_x^{ref}$ (pic 3) and sent to MPC. The state estimator estimates the current joint states (pic 5) and maps them to the system states $\boldsymbol{x}_{\text{cur}}$. Based on the SRB dynamics model, MPC solves QP and finds the optimal GRFs (pic 4). Finally, according to the desired joint states and GRFs $\boldsymbol{u}_0$, we calculate the feedforward torques $\boldsymbol{\tau}^{\text{ff}}$ based on the full dynamic model. Adding the feedback torques produced by a PD controller, the desired torques $\boldsymbol{\tau}^{\text{des}}$ can be obtained.

5 is the robot base frame. Therefore, $\boldsymbol{I}_5^c$ is the spatial inertia of the whole robot in the robot base frame. Substituting the $\boldsymbol{I}_5^c$ into Eq. 10, we can obtain the CoM position and standard inertia of the whole robot.

$$[\boldsymbol{c}_a]_\times = \begin{bmatrix} \frac{\mathbf{1}_{3\times3}}{m} & \mathbf{0}_{3\times3} \end{bmatrix} \boldsymbol{I}_5^c \begin{bmatrix} \mathbf{0}_{3\times3} \\ \frac{\mathbf{1}_{3\times3}}{m} \end{bmatrix}, \tag{13}$$

$$\begin{bmatrix} \boldsymbol{\Theta}^b \\ \boldsymbol{P}^c \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3\times1} \\ \boldsymbol{c}_a \end{bmatrix} + \boldsymbol{S}_b \boldsymbol{q}, \tag{14}$$

$$\bar{\boldsymbol{I}} = \begin{bmatrix} \mathbf{1}_{3\times3} & [\boldsymbol{c}_a]_\times^T \end{bmatrix} \boldsymbol{I}_5^c \begin{bmatrix} \mathbf{1}_{3\times3} \\ [\boldsymbol{c}_a]_\times^T \end{bmatrix}, \tag{15}$$

where $\boldsymbol{c}_a$ represents the robot's CoM position expressed in the robot base frame. $\bar{\boldsymbol{I}}$ is the standard inertia of the whole robot. $\boldsymbol{S}_b = [\mathbf{1}_{6\times6}\ \mathbf{0}_{6\times n}]$ is a selection matrix that selects the robot base frame.

To allow the SRB model to capture the motion characteristics of the real multibody system, we adopt the mapping introduced in [26], using the centroid momentum matrix (CMM), $\boldsymbol{A}_G \in \mathbb{R}^{6\times(n+6)}$, to obtain the average spatial velocity by considering the total momentum of the motion of each link.

$$\begin{bmatrix} \boldsymbol{\omega}^b \\ \dot{\boldsymbol{P}}^c \end{bmatrix} = \left({}^b\boldsymbol{X}_0^T \boldsymbol{I}_5^{cb} \boldsymbol{X}_0 \right)^{-1} \boldsymbol{A}_G(\boldsymbol{q}) \dot{\boldsymbol{q}}, \tag{16}$$

where ${}^b\boldsymbol{X}_0$ transforms spatial vectors from inertia frame 0 to robot base frame b.

According to Eq. (14) and Eq. (16), we can obtain the mapping $\mathcal{M}_x$ between the joint states and the system states. According to Eq. (15), we can obtain the mapping $\mathcal{M}_I$ between the joint configuration and corresponding inertia.

IV. MODEL PREDICTIVE CONTROL

The whole control framework is shown in Fig. 4. At each sampling time, MPC solves a finite horizon Optimal Control Problem (OCP). Given a current state estimated by the state estimator, reference trajectory planned by the reference trajectory planner, and the pre-defined inertia, MPC performs a forward rollout of the dynamics, minimizes the cost function, and finds the optional GRF sequence. The control signal for the first time step of the sequence is applied during the following sampling interval. Based on the full dynamic model, the joint torques are calculated according to GRFs. Adding the output of the PD controller, we can obtain the joint desire control torques.

A. Optimal Control Problem

The MPC control law could be obtained by solving the following OCP, and details will be described later.

$$\begin{aligned} \min_{\{u\}} & \sum_{t=0}^{T} \boldsymbol{\ell}_t(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{\ell}_{cone}(\boldsymbol{u}_k) + \boldsymbol{\ell}_{end}(\boldsymbol{x}_T) \\ \text{s.t. } & \boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, \bar{\boldsymbol{I}}_k) \\ & \bar{\boldsymbol{I}}_k = \mathcal{M}_I(\boldsymbol{q}_k^a) \\ & \boldsymbol{x}_0 = \mathcal{M}_x(\boldsymbol{q}_{cur}, \dot{\boldsymbol{q}}_{cur}), \end{aligned} \tag{17}$$

where T, $\boldsymbol{f}$, and $\boldsymbol{q}_k^a$ represent the length of MPC horizon, dynamics evolution, and pre-defined actuated joint positions at the k th step, respectively. $\boldsymbol{q}_{cur}$ and $\dot{\boldsymbol{q}}_{cur}$ represent the current joint positions and velocities. $\boldsymbol{\ell}_{end}$ is the terminal cost. $\boldsymbol{\ell}_{cone}$ is a penalty cost used to replace the friction cone inequality constraints [27]. $\boldsymbol{\ell}_t$ is the stage cost set as a quadratic function that penalizes the deviation between the

system states $\boldsymbol{x}_k$ and the reference states $\boldsymbol{x}_k^{ref}$, as well as the control $\boldsymbol{u}_k$ at the k th prediction step.

$$\boldsymbol{\ell}_t\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right)=\left\|\boldsymbol{x}_k-\boldsymbol{x}_k^{\boldsymbol{ref}}\right\|_{\boldsymbol{Q}}^2+\left\|\boldsymbol{u}_k\right\|_{\boldsymbol{R}}^2, \tag{18}$$

where $\boldsymbol{Q}$ and $\boldsymbol{R}$ are matrices of weights.

Rewriting Eq. (17) in the form of quadratic program (QP) [17], we use the open-source solvers qpOASES [28] to solve it and obtain the optional control sequence $\boldsymbol{traj}_u = \{\boldsymbol{u}_0, \boldsymbol{u}_1, ..., \boldsymbol{u}_{T-1}\}$.

B. Reference Trajectory

Given the remote controller commands, including the robot base frame's desired velocity $\boldsymbol{V}_{\text{des}}^{\text{base}}$, angular velocity $\boldsymbol{\omega}_{\text{des}}^{\text{base}}$, and the angular velocity of the trunk motors $\boldsymbol{\omega}_{\text{des}}^{trunk}$, the reference trajectory planner calculates the reference state trajectories of joints by integrating these velocities.

$$\boldsymbol{traj}_q^{ref}=\left\{\boldsymbol{q}_1, ..., \boldsymbol{q}_T\right\}, \boldsymbol{traj}_{\dot{q}}^{ref}=\left\{\dot{\boldsymbol{q}}_1, ..., \dot{\boldsymbol{q}}_T\right\}, \tag{19}$$

where $\boldsymbol{traj}_q^{ref}$ and $\boldsymbol{traj}_{\dot{q}}^{ref}$ respectively represent the trajectories of joint generalized positions and velocities. $\boldsymbol{traj}_q^a$ and $\boldsymbol{traj}_{\dot{q}}^a$ in Eq. (3) are the actuated parts of them.

According to Eq. (9), by mapping the joint states to system states, we can get reference trajectories of system states $\boldsymbol{traj}_x^{ref} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_T\}$.

$$\boldsymbol{traj}_x^{ref}=\mathcal{M}_x\left(\boldsymbol{traj}_q^{ref}, \boldsymbol{traj}_{\dot{q}}^{ref}\right). \tag{20}$$

C. Discrete-Time Dynamics

We calculate the pre-defined trajectory of the inertia $\boldsymbol{traj}_{\bar{I}}^{ref} = \left\{\bar{\boldsymbol{I}}_{\boldsymbol{0}}, \bar{\boldsymbol{I}}_{\boldsymbol{1}}, ..., \bar{\boldsymbol{I}}_{\boldsymbol{T}}\right\}$ throughout the MPC horizon at first.

$$\boldsymbol{traj}_{\bar{I}}^{ref}=\mathcal{M}_I\left(\boldsymbol{traj}_q^{ref}\right). \tag{21}$$

By assembling the pre-defined inertia sequence $\boldsymbol{traj}_{\bar{I}}^{ref}$ and the SRB dynamics, Eq. (5) and Eq. (6), the discrete-time dynamics can be expressed in the following form:

$$\begin{aligned}\boldsymbol{x}_{k+1} &= f\left(\boldsymbol{x}_k, \boldsymbol{u}_k, \bar{\boldsymbol{I}}_k\right)\\ &= \boldsymbol{A}_k\boldsymbol{x}_k+\boldsymbol{B}_k\boldsymbol{u}_k+\boldsymbol{G},\end{aligned} \tag{22}$$

where

$$\begin{aligned}\boldsymbol{A}_k &= \begin{bmatrix}\boldsymbol{1}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{R}_z\left(\psi_k\right)\Delta t & \boldsymbol{0}_{3\times3}\\ \boldsymbol{0}_{3\times3} & \boldsymbol{1}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{1}_{3\times3}\Delta t\\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{1}_{3\times3} & \boldsymbol{0}_{3\times3}\\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{1}_{3\times3}\end{bmatrix},\\ \boldsymbol{B}_k &= \begin{bmatrix}\boldsymbol{0}_{3\times3} & \cdots & \boldsymbol{0}_{3\times3}\\ \boldsymbol{0}_{3\times3} & \cdots & \boldsymbol{0}_{3\times3}\\ \Delta t\bar{\boldsymbol{I}}_k^{-1}\left[\boldsymbol{r}_1-\boldsymbol{P}_k^c\right]_\times & \cdots & \Delta t\bar{\boldsymbol{I}}_k^{-1}\left[\boldsymbol{r}_{nc}-\boldsymbol{P}_k^c\right]_\times\\ \boldsymbol{1}_{3\times3}\Delta t/m & \cdots & \boldsymbol{1}_{3\times3}\Delta t/m\end{bmatrix},\\ \boldsymbol{G} &= \begin{bmatrix}\boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times3} & \boldsymbol{g}^\top\Delta t\end{bmatrix}^\top.\end{aligned} \tag{23}$$

$\boldsymbol{g} \in \mathbb{R}^3$ represents the gravity acceleration. $\bar{\boldsymbol{I}}_k$, $\boldsymbol{P}_k^c$, and $\boldsymbol{u}_k = \begin{bmatrix}\boldsymbol{f}_1 & \cdots & \boldsymbol{f}_{nc}\end{bmatrix}^\top$ represent the whole robot's inertia and CoM position and the control input at time step k.

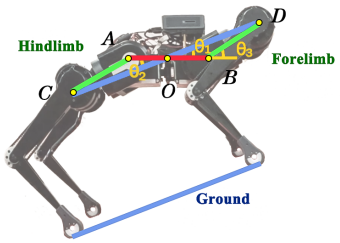


Fig. 5. The blue line *CD* is obtained by connecting the shoulders at both ends. The red line, *AB*, represents the middle part of the trunk, which is needed to remain parallel to the horizontal plane. The green lines, *AC* and *BD*. The angle between the blue line and red is called θ_1 and θ_2, and the angle between green and red is called θ_3. The front and rear motor's rotation angles are the same, so $\theta_1 = \theta_2$.

D. Swing Legs Control

The leg planner plans the trajectories of feet based on a periodic phase-based gait scheduler [17]. By applying the first time-step control input of $\boldsymbol{traj}_u$ during the following sampling interval, we calculate the feedforward torque $\boldsymbol{\tau}^{\text{ff}}$ based on the full dynamic model Eq. (1).

$$\boldsymbol{\tau}^{\text{ff}}=\boldsymbol{\xi_j}\left(J_c^T\boldsymbol{u}_0-\boldsymbol{H}\ddot{\boldsymbol{q}}_{des}-\boldsymbol{C}-\boldsymbol{G}\right). \tag{24}$$

By adding the feedback torque which created by a PD controller, we obtain the desired joint torques $\boldsymbol{\tau}_{\text{des}}$.

$$\boldsymbol{\tau}_{\text{des}}=\boldsymbol{\tau}^{\text{ff}}+\boldsymbol{K}_p\left(\boldsymbol{q}_{des}-\boldsymbol{q}_{cur}\right)+\boldsymbol{K}_d\left(\dot{\boldsymbol{q}}_{des}-\dot{\boldsymbol{q}}_{cur}\right), \tag{25}$$

where $\boldsymbol{K}_p$ and $\boldsymbol{K}_d$ are the joint position and velocity gain matrices for the feedback term. $\boldsymbol{q}_{des}$, $\dot{\boldsymbol{q}}_{des}$, $\ddot{\boldsymbol{q}}_{des}$ are the desired position, velocities, and accelerations, respectively.

V. Experimental Results

Yat-sen Lion is 0.53 meters long, 0.38 meters wide, and weighs 18 kg. The spine joints can rotate ± 30 degrees, and the max joint torque is 35 N.m. The electronic system consists of an onboard computer Intel NUC and an IMU MTI-630. We conduct experiments on both simulation and physical platforms, including keeping balance on a moving platform, freely changing spine posture during moving, and bounding gait.

A. Balancing Control

As shown in Fig. 5, due to the hardware structure, Yat-sen Lion can actively keep the middle part of the trunk parallel to the horizontal plane in different terrain. Assuming that the length of *A0* and *B0* are *L1*. *AC* and *BD* are *L2*. *CO* and *DO* are *L3*. The formulas could be conducted as follow:

$$L_3=L_1\cos\theta_3+\sqrt{L_1^2\cos^2\theta_3-L_1^2+L_2^2}, \tag{26}$$

$$\theta_1=\pi-\operatorname{acos}\left(\frac{L_1^2+L_2^2-L_3^2}{2L_1L_2}\right), \tag{27}$$

where θ_3 is the estimated angle of the ground. θ_1 is the angle that the forelimb and hindlimb joints should maintain. As Fig. 7 shows, the pitch angle deviation of the robot base frame is within ± 0.05 rad.

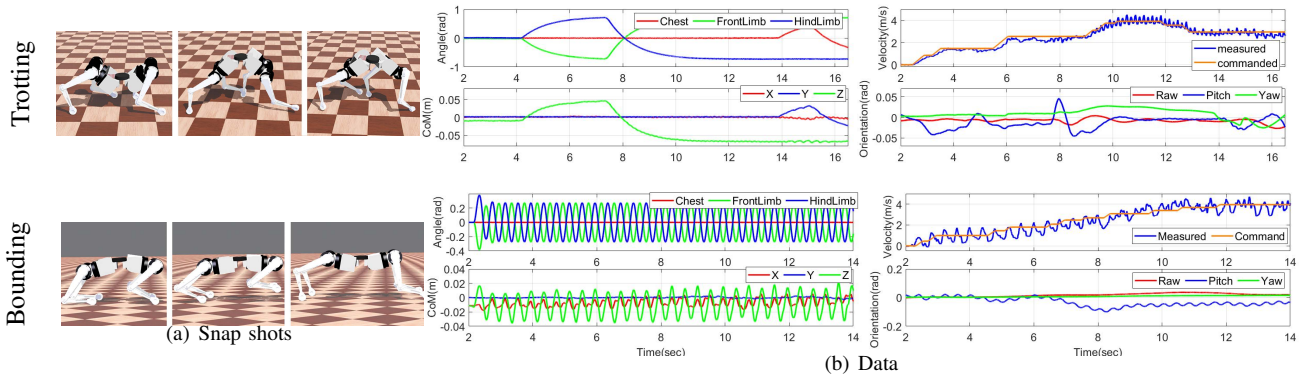


Fig. 6. The first set is that the robot freely changes the spine posture during trotting, and the second set is the robot's bounding gait based on the synergistic movement of the spine and limbs. The four images in part (b) are arranged as follows: the top left image shows the spine joint angles, the top right image shows the CoM position in the robot base frame. Note that the robot base frame also changes with the spine posture, which affects the relative position of CoM. The bottom left image shows the robot base frame's velocity tracking in the x-axis, and the bottom right image shows the orientation.

B. Bounding

The most classic gait of the spined quadruped robots is bounding. To demonstrate that our algorithm frame can inherit previous works, we presented a bounding gait simulation and actuated the spine joint angles with a minimalistic control input, the forelimb joint $\theta_1 = A_f \sin(2\pi f_r t)$ and the hindlimb joint $\theta_3 = -\theta_1$, where $A_f = 0.21$ and $f_r = 2.5$ are the amplitude and frequency, respectively. As shown in the second set of Fig. 6, the robot accelerates in the x-direction until it reaches the final velocity of 4 m/s.

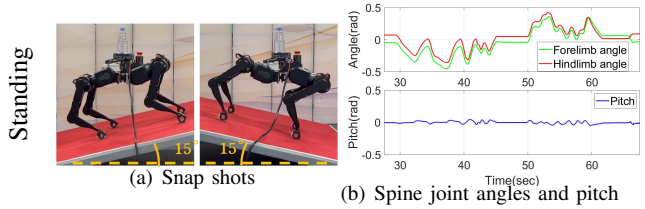


Fig. 7. We place the robot on a tiltable platform and lift the platform back and forth. The maximum lifting angle is about 15 degrees. (b) presents the angles of the robot's spine joints and the pitch angles of the base frame.

C. Changing spine posture while trotting

To demonstrate the capability of the robot to freely change spine posture during motion, we arbitrarily control the spine motion and give the desired velocity. In the simulation experiments, as shown in the first set of Fig. 6, the robot accelerates in the x-direction to the max velocity of 3.8 m/s while concaving and arching its spine. And then slow down to 3 m/s while turning its spine. Although the CoM position and inertia change significantly during the process, the velocity deviation is within $\pm$ 0.7 m/s, and orientation deviations are all within $\pm$ 0.05rad. We conducted two hardware experiments shown in Fig. 8. In the first set, the robot achieves the max velocity of 0.5 m/s while concaving and arching its spine. In the second set, the robot achieves the max Yaw angular velocity of 0.5 rad/s while bending its spine. The result indicates that the robot can track the desired velocity when its spine is arbitrarily moving.

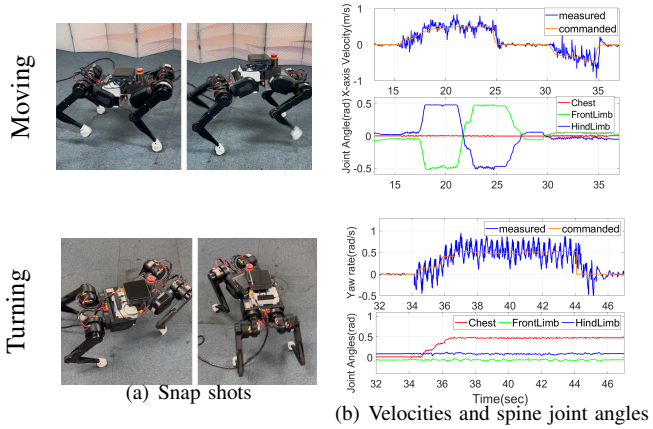


Fig. 8. Velocity tracking of trotting when changing spine posture. The two sets present the robot trotting in the x direction and turning, respectively. (b) shows the robot base frame's velocity tracking and spine joint angles.

VI. CONCLUSION

This paper proposes an MPC approach for torque-controlled spined quadruped robots. The robot was modeled as a SRB with a fully actuated spine. Based on the pre-defined actuated joint states and inertia, the reduced model can accurately describe the dynamics evolution of the original high DoF multi-body system and meet the real-time requirements of embedded platforms. Furthermore, we develop the Yat-sen Lion spined quadruped robot as a new legged robot platform consisting of three active spine joints. Yat-sen Lion can freely change its spine posture during motion. In the simulation, Yat-sen Lion transforms its trunk when trotting at a speed of 3.8 m/s or an angular velocity of 3 rad/s. In the physical platform experiment, it can achieve 0.5 m/s and 0.5 rad/s. In the future work, we will improve the hardware performance and increase its energy efficiency.

REFERENCES

[1] M. Hildebrand, "Motions of the running cheetah and horse," *Journal of Mammalogy*, vol. 40, no. 4, pp. 481–495, 1959.

[2] R. M. Alexander, N. J. Dimery, and R. Ker, "Elastic structures in the back and their role in galloping in some mammals," *Journal of zoology*, vol. 207, no. 4, pp. 467–482, 1985.

[3] K. F. Leeser, "Locomotion experiments on a planar quadruped robot with articulated spine," Ph.D. dissertation, Massachusetts Institute of Technology, 1996.

[4] K. Berns, W. Ilg, M. Deck, J. Albiez, and R. Dillmann, "Mechanical construction and computer architecture of the four-legged walking machine bisam," *IEEE/ASME transactions on mechatronics*, vol. 4, no. 1, pp. 32–38, 1999.

[5] M. Khoramshahi, A. Spröwitz, A. Tuleu, M. N. Ahmadabadi, and A. J. Ijspeert, "Benefits of an active spine supported bounding locomotion with a small compliant quadruped robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 3329–3334.

[6] P. Eckert, A. Spröwitz, H. Witte, and A. J. Ijspeert, "Comparing the effect of different spine and leg designs for a small bounding quadruped robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 3128–3133.

[7] J. Duperret, B. Kramer, and D. E. Koditschek, "Core actuation promotes self-manipulability on a direct-drive quadrupedal robot," in *International Symposium on Experimental Robotics*, 2016, pp. 147–159.

[8] J. Duperret and D. E. Koditschek, "Empirical validation of a spined sagittal-plane quadrupedal model," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 1058–1064.

[9] C. Wang, T. Zhang, X. Wei, Y. Long, and S. Wang, "Bio-inspired control strategy study for the quadruped robot with a segmented spine," *Industrial Robot: An International Journal*, 2017.

[10] Z. Wei, G. Song, H. Sun, Q. Qi, Y. Gao, and G. Qiao, "Turning strategies for the bounding quadruped robot with an active spine," *Ind Rob*, 2018.

[11] Z. Li and Y. Tan, "Trotting motion of the quadruped model with two spinal joints and its dynamics features," *Journal of Robotics*, 2020.

[12] Q. Shi, J. Gao, S. Wang, X. Quan, G. Jia, Q. Huang, and T. Fukuda, "Development of a small-sized quadruped robotic rat capable of multimodal motions," *IEEE Trans. Robot.*, 2022.

[13] U. Culha and U. Saranli, "Quadrupedal bounding with an actuated spinal joint," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1392–1397.

[14] K. Ye and K. Karydis, "Modeling and trajectory optimization for standing long jumping of a quadruped with a preloaded elastic prismatic spine," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 902–908.

[15] L. Li, S. Ma, I. Tokuda, F. Asano, M. Nokata, Y. Tian, and L. Du, "Synergetic effect between limbs and spine dynamics in quadruped walking robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 6818–6823.

[16] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: a review," *Neural networks*, vol. 21, no. 4, pp. 642–653, 2008.

[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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1–9.

[18] 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 Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1458–1465, 2018.

[19] Y. Ding, A. Pandala, C. Li, Y.-H. Shin, and H.-W. Park, "Representation-free model predictive control for dynamic motions in quadrupeds," *IEEE Trans. Robot.*, vol. 37, no. 4, pp. 1154–1171, 2021.

[20] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," *arXiv preprint arXiv:1909.06586*, 2019.

[21] T. Li, J. Won, S. Ha, and A. Rai, "Model-based motion imitation for agile, diverse and generalizable quadupedal locomotion," *arXiv preprint arXiv:2109.13362*, 2021.

[22] J. J. Craig, *Introduction to robotics: mechanics and control*. Pearson Educacion, 2005.

[23] W. Du, Z. Wang, E. Moullet, and F. Benamar, "Meaningful centroidal frame orientation of multi-body floating locomotion systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 3061–3067.

[24] A. Saccon, S. Traversaro, F. Nori, and H. Nijmeijer, "On centroidal dynamics and integrability of average angular velocity," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 943–950, 2017.

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

[26] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous robots*, vol. 35, no. 2, pp. 161–176, 2013.

[27] T. Corbères, T. Flayols, P.-A. Léziart, R. Budhiraja, P. Souères, G. Saurel, and N. Mansard, "Comparison of predictive controllers for locomotion and balance recovery of quadruped robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 5021–5027.

[28] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpoases: A parametric active-set algorithm for quadratic programming," *Math Program Comput*, vol. 6, no. 4, pp. 327–363, 2014.