IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Force Allocation Control for Redundant-Torso Biomimetic Quadrupeds via Virtual Force and Fully Actuated System Approach

Zhiqin Zhuo, Ke Huang, Zhigang Wu, and Jianping Jiang

***Abstract*—Fully actuated system approach (FASA) provides a promising control framework for robots with redundant actuation, offering simplified controller design and increased design freedom. However, its application to legged robots remains challenging due to hybrid actuation from intermittent ground contact and redundant inputs. To address this, we propose Virtual Force-based FASA (VF-FASA), which introduces virtual forces as intermediaries to construct full-actuation conditions required by FASA. FASA generates virtual control laws based on a simplified torso dynamics model, and a matrix-weighted pseudoinverse optimization is employed to map these virtual inputs into actual torso joint torques and foot contact forces. This method achieves coordinated control of both the floating base and redundant torso, effectively leveraging joint redundancy for improved whole-body motion. Simulation results on a redundant-torso quadruped robot demonstrate robust trajectory tracking and effective whole-body coordination under dynamic locomotion. The framework expands FASA to legged systems, providing an effective approach for controlling quadruped robots.**

***Index Terms*—Redundant robotics, Legged robots, Force Control, Fully actuated system approach, Optimization control.**

I. Introduction

LEGGED robots have emerged as a prominent research topic in robotics due to their superior adaptability to complex and unstructured environments. Torso-actuated quadruped robots [1-4], equipped with redundant joints in torso and legs, offer higher degrees of freedom (DoFs), demonstrating potential advantages such as motion redundancy and biomimetic agility. Compared to traditional rigid-bodied quadruped robots [5], however, this structural redundancy introduces additional complexity into force transmission pathways from the feet to the base, posing significant challenges for control.

Various force control strategies have been developed for traditional rigid-bodied quadruped robots. Simplified single-body model approaches, such as spring-loaded inverted pendulum (SLIP) [6, 7] and virtual model control (VMC) [8, 9], employ PD controllers to generate virtual spring-damper forces for motion stabilization. In contrast, full dynamic model-based methods, such as whole-body control (WBC) [10], leverage the complete dynamics of the robot to enable precise control. Optimization-based control methods, including linear quadratic regulator (LQR) [11] and model predictive control (MPC) [12-14], are also widely used. Hierarchical architectures that combine simplified high-level models with accurate low-level controllers have been extensively explored, such as MPC with quadratic programming (QP) [15], zero moment point (ZMP) with WBC [16], VMC with WBC [17], MPC with WBC [18], and VMC with central pattern generator (CPG) and SLIP models [19]. Meanwhile, CPG-based approaches inspired by biological neural mechanisms [20] and learning-based control approaches [21] have also attracted increasing attention.

However, applying the aforementioned force control methods to redundant-torso quadruped robots remains challenging. Simplified single-rigid-body models (e.g., VMC) cannot coordinate torso dynamics and force distribution, thus failing to exploit structural redundancy. In contrast, full-order optimization methods (e.g., WBC, MPC) entail prohibitive computational costs that limit real-time control frequencies. Meanwhile, CPG methods are ill-suited for the non-periodic demands of torso control, and learning-based approaches suffer from data dependency and lack formal stability guarantees. Consequently, many torso-actuated quadruped robots still rely on traditional kinematic control strategies [1-4], which often fall short in achieving the robustness and flexibility required for dynamic locomotion. This highlights the need for a control framework that leverages redundancy while maintaining design simplicity.

Fully actuated system approach (FASA) proposed by Duan et al. [22-25] simplifies controller design by directly handling second-order dynamics without transforming them into first-order state-space formulations. It enables flexible controller design and has been successfully applied to constrained high-order strict-feedback systems[26], robotic manipulators and quadcopters for trajectory tracking [27, 28], missile guidance [29], and spacecraft attitude-orbit control [30, 31]. These studies highlight FASA's potential for legged robot control, yet its direct implementation has rarely been explored. Applying FASA to legged robots remains challenging due to hybrid actuation, which involves frequent transitions between under-actuated and over-actuated states. Discontinuous foot-ground interaction, friction-dependent lateral forces, and excessive control inputs relative to the system's DoFs introduce

Manuscript received: July 27, 2025; Revised: October 24, 2025; Accepted: November 22, 2025.

This paper was recommended for publication by Editor Clement Gosselin upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the National Natural Science Foundation of China under Grant No. 62388101 and Grant No. 52472390.

The authors are with the School of Aeronautics and Astronautics, Sun Yat-sen University, Shenzhen, China (e-mail: zhuozhq@mail2.sysu.edu.cn, huangk86@mail2.sysu.edu.cn, wuzhigang@mail.sysu.edu.cn). Corresponding author: Jianping Jiang (e-mail: jiangjp8@mail.sysu.edu.cn).

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

redundancy and inequality constraints, violating the full-actuation assumption [32] required by FASA.

To overcome this challenge, we propose a novel control framework, Virtual Force-based FASA (VF-FASA). For tractable modeling and controller synthesis, leg dynamics are neglected and only foot contact forces are considered, since each leg contributes a small fraction of the total mass and operates at low frequencies. This simplification focuses the controller on torso-foot force coordination under hybrid actuation. We replace these torso-foot forces with full-order virtual forces corresponding to the system states, thereby satisfying the full-actuation condition required by FASA. The FASA controller then computes these virtual forces as control laws. While conceptually similar to traditional VMC [8, 9], VF-FASA incorporates torso dynamics within the FASA framework and enables the redundant torso actuation to be effectively utilized for coordinated whole-body control.

However, the virtual forces computed by FASA introduce a force-torque allocation problem when mapped to actual torso joint torques and foot contact forces. The traditional Jacobian pseudoinverse mapping [33] performs uniform force allocation without considering optimization, limiting the exploitation of structural redundancy. To address this limitation, we formulate a constrained quadratic optimization problem to minimize a weighted control output, yielding a matrix-weighted pseudo-inverse of the mapping matrix. This allows optimized mapping of virtual forces to torso joints and foot contact forces, enabling effective whole-body control.

In summary, the main contribution of this paper is the proposed VF-FASA, a unified control framework that extends FASA to quadrupeds with torso redundancy. The framework introduces a virtual force construction strategy to establish full-actuation conditions under hybrid actuation and formulates a matrix-weighted pseudoinverse optimization to exploit redundancy in force-torque allocation. Together, these components enable coordinated whole-body control while maintaining simplicity and computational efficiency. For clarity, the main variables and symbols used throughout the paper are summarized in Table I.

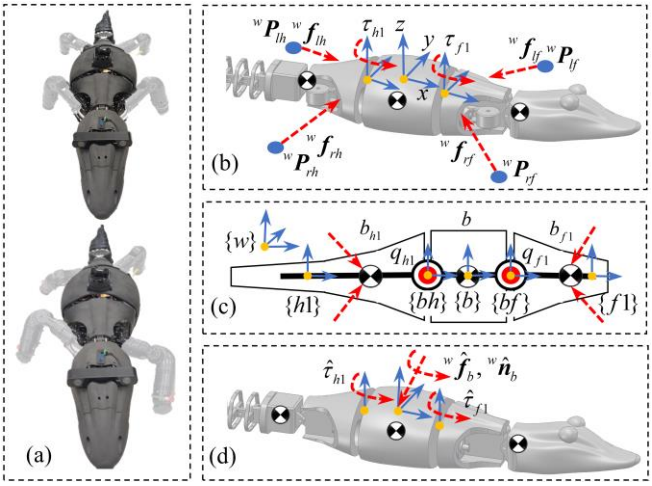

Fig. 1 Robot architecture and simplified modeling framework. (a) Redundant-torso quadruped robot prototype. (b) Simplified configuration of the robot. (c) Equivalent three-link model. (d) Virtual force representation.

TABLE. I NOMENCLATURE

Sym.	Desc.	Sym.	Desc.	Sym.	Desc.
$\{w\}$ $\{b\}$	World frame and base frame	${}^{w}\hat{\boldsymbol{f}}_b$ ${}^{w}\hat{\boldsymbol{n}}_b$	Virtual forces and moments on the base in frame $\{w\}$	${}^{w}\boldsymbol{P}_{leg}$	Foot position in frame $\{w\}$
$\{f1\}$ $\{h1\}$	Front torso link frame and rear torso link frame	$\hat{\tau}_{f1}$ $\hat{\tau}_{h1}$	Virtual torque applied at the front and rear torso joints	${}^{a}\boldsymbol{R}_b$	Rotation matrix from $\{b\}$ to $\{a\}$ (${}^{w}\boldsymbol{R}_b$, ${}^{b}\boldsymbol{R}_{f1}$, ${}^{b}\boldsymbol{R}_{h1}$, ${}^{f1}\boldsymbol{R}_{bf}$, ${}^{h1}\boldsymbol{R}_{bh}$)
$\{bf\}$ $\{bh\}$	Frames attached to the front and rear torso joints	$\hat{\boldsymbol{u}}$, $\hat{\boldsymbol{u}}_{eq}$	Virtual control law and virtual feedback control law	$\boldsymbol{M}$, $\boldsymbol{C}$ $\boldsymbol{G}$	Inertia, Coriolis/centrifugal, and gravity matrices of the system
b_{f1} , b, b_{h1}	Front torso link, base link and rear torso link	${}^{w}\boldsymbol{f}_{leg}$	Foot contact force in frame $\{w\}$ ($leg = rf, lf, rh, lh$)	$\boldsymbol{f}_r$, $\boldsymbol{c}$	Joint damping force and damping coefficient
q_{f1} , q_{h1}	Front torso joint angle and rear torso joint angle	τ_{f1} τ_{h1}	Actual torque applied at the front and rear torso joints	$\boldsymbol{f}$, $\boldsymbol{B}$	The dynamics and input matrix of the fully actuated system
$\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$	Generalized state variables and its derivative	$\boldsymbol{u}$	Actual control law	$\boldsymbol{\Phi}$ $\boldsymbol{A}_i$	Closed-loop dynamic matrix and desired eigenvalue matrix
$\boldsymbol{q}_d$, $\dot{\boldsymbol{q}}_d$, $\ddot{\boldsymbol{q}}_d$	Desired state and its derivative	${}^{bf}\boldsymbol{P}_{f1}$ ${}^{bh}\boldsymbol{P}_{h1}$	Position vectors from frames $\{bf\}$ $\{bh\}$ to $\{f1\}$ $\{h1\}$	$\boldsymbol{F}$, $\boldsymbol{Z}$	Core and free parameter matrices of the FASA controller
$\boldsymbol{e}$, $\dot{\boldsymbol{e}}$, $\ddot{\boldsymbol{e}}$	Tracking error and its derivative	${}^{b}\boldsymbol{p}_{bf,bc}$ ${}^{b}\boldsymbol{p}_{bh,bc}$	Vectors from frames $\{bf\}$ and $\{bh\}$ to the CoM of link b	$\boldsymbol{J}_{leg}$	Jacobian matrix for single leg displacement
${}^{xx}\boldsymbol{F}_{xx}$ ${}^{xx}\boldsymbol{N}_{xx}$	The inertial forces and moments of each link	${}^{f1}\boldsymbol{r}_{f1,c}$ ${}^{h1}\boldsymbol{r}_{h1,c}$	The CoM position vectors of links b_{f1} and b_{h1}	$\hat{\boldsymbol{B}}$ $\hat{\boldsymbol{B}}^{\dagger}$	Mapping matrix and its weighted pseudoinverse
${}^{xx}\boldsymbol{f}_{xx}$ ${}^{xx}\boldsymbol{n}_{xx}$	Force and moment from the base link to adjacent links	${}^{b}\boldsymbol{P}_{f1}$ ${}^{b}\boldsymbol{P}_{h1}$	Position vectors of frames $\{f1\}$ and $\{h1\}$ in frame $\{b\}$	$\boldsymbol{W}_n$	The weight matrix for the weighted pseudoinverse $\hat{\boldsymbol{B}}^{\dagger}$

II. MODEL OF QUADRUPED ROBOT

A. Redundant-Torso Biomimetic Quadrupeds

Figure 1(a) illustrates a quadruped robot inspired by crocodilian morphology, featuring torso and leg joint redundancy with a flexible segmented structure. The robot comprises 21 actuated DoFs: 16 in legs, 1 for head pitch, 2 for torso oscillation, and 2 for tail pitch and swing. An additional 6 unactuated DoFs describe the absolute attitude and position of the floating base in the world frame $\{w\}$.

Given that each leg accounts for less than 7% of total mass (2 kg out of 28.6 kg) and operates below 1 Hz, leg dynamics are neglected, and only foot contact forces are modeled (Fig. 1(b)). These foot contact forces are mapped to joint torques via kinematic transformations. Head and tail motions are treated as negligible disturbances during crawling. In contrast, the torso comprises serially connected revolute joints with nearly uniform masses, and its dynamics are preserved as they significantly influence the robot's center of mass (CoM).

The robot is simplified as a three-link serial chain with two torso joints (Fig. 1(c)). The head and front torso form link b_{f1}, the middle torso segment is floating base b, and the rear torso with tail form link b_{h1}. The generalized state variables of the robot are defined as:

$$\boldsymbol{q} = \left[\alpha, \beta, \gamma, p_x, p_y, p_z, q_{f1}, q_{h1}\right]^{T} \tag{1}$$

where $\boldsymbol{q} \in \mathrm{R}^{8\times 1}$. The generalized coordinates $[\alpha, \beta, \gamma, p_x, p_y, p_z]$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

describe the attitude (roll, pitch, yaw) and position of the floating base b in frame $\{w\}$. Front leg contact forces act on link b_{f1} and are transmitted to link b through joint q_{f1}, while rear leg contact forces act on link b_{h1} via joint q_{h1}. A high-level planner provides the desired foot placements.

B. Simplified Robot Dynamics

A virtual force dynamics model is established using the Newton-Euler method. External forces and torques acting on the robot include foot contact forces and joint torques (Fig. 1(b)). The foot contact forces are neglected in this analysis. The inertial forces and moments acting on each link's CoM, denoted as ${}^{xx}\boldsymbol{F}_{xx}$ and ${}^{xx}\boldsymbol{N}_{xx}$ $(xx = f1, b, h1)$, are derived from the Newton-Euler equations. By applying equilibrium equations to links b_{f1} and b_{h1}, the internal forces and moments exerted by link b are obtained as follows:

$$\begin{cases} {}^{f1}\boldsymbol{f}_{f1} = {}^{f1}\boldsymbol{F}_{f1} \\ {}^{f1}\boldsymbol{n}_{f1} = {}^{f1}\boldsymbol{N}_{f1} + ({}^{f1}\boldsymbol{R}_{bf}\,{}^{bf}\boldsymbol{P}_{f1} + {}^{f1}\boldsymbol{r}_{f1,c}) \times {}^{f1}\boldsymbol{f}_{f1} \end{cases} \tag{2}$$

$$\begin{cases} {}^{h1}\boldsymbol{f}_{h1} = {}^{h1}\boldsymbol{F}_{h1} \\ {}^{h1}\boldsymbol{n}_{h1} = {}^{h1}\boldsymbol{N}_{h1} + ({}^{h1}\boldsymbol{R}_{bh}\,{}^{bh}\boldsymbol{P}_{h1} + {}^{h1}\boldsymbol{r}_{h1,c}) \times {}^{h1}\boldsymbol{f}_{h1} \end{cases} \tag{3}$$

As shown in Fig. 1(c), frames $\{f1\}$ and $\{h1\}$ are the fixed coordinate systems attached to links b_{f1} and b_{h1}. Frames $\{bf\}$ and $\{bh\}$ are located at joints q_{f1} and q_{h1}, and are aligned with frame $\{b\}$. Matrices ${}^{f1}\boldsymbol{R}_{bf}$ and ${}^{h1}\boldsymbol{R}_{bh}$ denote the transformation matrices from frame $\{bf\}$ to $\{f1\}$, and from frame $\{bh\}$ to $\{h1\}$, respectively. Vectors ${}^{bf}\boldsymbol{P}_{f1}$ and ${}^{bh}\boldsymbol{P}_{h1}$ represent the position vectors from frame $\{bf\}$ to $\{f1\}$, and from frame $\{bh\}$ to $\{h1\}$, respectively. Vectors ${}^{f1}\boldsymbol{r}_{f1,c}$ and ${}^{h1}\boldsymbol{r}_{h1,c}$ denote the CoM position vectors of links b_{f1} and b_{h1}, expressed in frames $\{f1\}$ and $\{h1\}$, respectively.

Combining (2) and (3) yields the generalized forces and moments at the CoM of link b:

$$\begin{cases} {}^{b}\boldsymbol{f}_{b} = {}^{b}\boldsymbol{F}_{b} + {}^{b}\boldsymbol{R}_{f1}\,{}^{f1}\boldsymbol{f}_{f1} + {}^{b}\boldsymbol{R}_{h1}\,{}^{h1}\boldsymbol{f}_{h1} \\ {}^{b}\boldsymbol{n}_{b} = {}^{b}\boldsymbol{N}_{b} + {}^{b}\boldsymbol{R}_{f1}\,{}^{f1}\boldsymbol{n}_{f1} + {}^{b}\boldsymbol{R}_{h1}\,{}^{h1}\boldsymbol{n}_{h1} + {}^{b}\boldsymbol{r}_{bc} \times {}^{b}\boldsymbol{f}_{b} \\ \quad - {}^{b}\boldsymbol{p}_{bf,bc} \times ({}^{b}\boldsymbol{R}_{f1}\,{}^{f1}\boldsymbol{f}_{f1}) - {}^{b}\boldsymbol{p}_{bh,bc} \times ({}^{b}\boldsymbol{R}_{h1}\,{}^{h1}\boldsymbol{f}_{h1}) \end{cases} \tag{4}$$

where ${}^{b}\boldsymbol{f}_{b}$ and ${}^{b}\boldsymbol{n}_{b}$ are the assumed external force and moment acting on the base link b. Matrices ${}^{b}\boldsymbol{R}_{f1}$ and ${}^{b}\boldsymbol{R}_{h1}$ denote the transformation matrices from frame $\{f1\}$ to $\{b\}$, and from frame $\{h1\}$ to $\{b\}$, respectively. Vectors ${}^{b}\boldsymbol{p}_{bf,bc}$ and ${}^{b}\boldsymbol{p}_{bh,bc}$ are defined in frame $\{b\}$ as pointing from the origins of frames $\{bf\}$ and $\{bh\}$ to the CoM of link b, respectively. By transforming ${}^{b}\boldsymbol{f}_{b}$ and ${}^{b}\boldsymbol{n}_{b}$ into frame $\{w\}$ and incorporating the torques from joints $\boldsymbol{q}_{f1}$ and $\boldsymbol{q}_{h1}$, we obtain:

$$\begin{cases} {}^{w}\hat{\boldsymbol{n}}_{b} = {}^{w}\boldsymbol{R}_{b}\,{}^{b}\boldsymbol{n}_{b} \\ {}^{w}\hat{\boldsymbol{f}}_{b} = {}^{w}\boldsymbol{R}_{b}\,{}^{b}\boldsymbol{f}_{b} \\ \hat{\tau}_{f1} = {}^{f1}\boldsymbol{n}_{f1}^{T}\,{}^{f1}\boldsymbol{R}_{bf}\,{}^{bf}\boldsymbol{z}_{bf} \\ \hat{\tau}_{h1} = {}^{h1}\boldsymbol{n}_{h1}^{T}\,{}^{h1}\boldsymbol{R}_{bh}\,{}^{bh}\boldsymbol{z}_{bh} \end{cases} \tag{5}$$

As shown in Fig. 1(d), we define the generalized moment ${}^{w}\hat{\boldsymbol{n}}_{b} = [{}^{w}\hat{n}_{bx}, {}^{w}\hat{n}_{by}, {}^{w}\hat{n}_{bz}]^{T}$ and force ${}^{w}\hat{\boldsymbol{f}}_{b} = [{}^{w}\hat{f}_{bx}, {}^{w}\hat{f}_{by}, {}^{w}\hat{f}_{bz}]^{T}$ as the virtual wrench on the base link b, with torques $\hat{\tau}_{f1}$ and $\hat{\tau}_{h1}$ as the virtual torques on the torso joints. Vectors ${}^{bf}\boldsymbol{z}_{bf}$ and ${}^{bh}\boldsymbol{z}_{bh}$ denote the z-axis unit vectors of frames $\{bf\}$ and $\{bh\}$, respectively. The overall virtual force is given by:

$$\hat{\boldsymbol{u}} = [{}^{w}\hat{\boldsymbol{n}}_{b}, {}^{w}\hat{\boldsymbol{f}}_{b}, \hat{\tau}_{f1}, \hat{\tau}_{h1}]^{T} \in \mathrm{R}^{8\times 1} \tag{6}$$

We differentiate the virtual force equation (5) and incorporate joint damping to derive the dynamics of the floating-base virtual generalized force. The joint damping force is given by $\boldsymbol{f}_{r}(\dot{\boldsymbol{q}}) = -\boldsymbol{c}\dot{\boldsymbol{q}}$, based on D'Alembert's principle, where $\boldsymbol{c} = \mathrm{diag}(\boldsymbol{0}_{1\times 6}, c_{f1}, c_{h1})$ is the damping coefficient matrix. The derived dynamics are as follows:

$$\begin{gathered} \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) - \boldsymbol{f}_{r}(\dot{\boldsymbol{q}}) = \hat{\boldsymbol{u}} \\ \begin{cases} \boldsymbol{M} = \boldsymbol{J}_{\hat{\boldsymbol{u}}}(\ddot{\boldsymbol{q}}) \\ \boldsymbol{G} = \hat{\boldsymbol{u}}(\ddot{\boldsymbol{q}} = 0^{8\times 1}, \dot{\boldsymbol{q}} = 0^{8\times 1}) \\ \boldsymbol{C} = \boldsymbol{J}_{\hat{\boldsymbol{u}}-\mathbf{M}\ddot{\boldsymbol{q}}-\mathbf{G}}(\dot{\boldsymbol{q}}) \end{cases} \end{gathered} \tag{7}$$

where $\boldsymbol{M} \in \mathrm{R}^{8\times 8}$, $\boldsymbol{C} \in \mathrm{R}^{8\times 8}$ and $\boldsymbol{G} \in \mathrm{R}^{8\times 1}$.

III. Control Architecture

The complete control block diagram is shown in Fig. 2. Its core components are the virtual force dynamics, the FASA controller and the weighted optimal allocation module.

A. Fully Actuated System Approaches

According to FASA [23-25], the general form of an r-order fully actuated system is as follows:

$$\boldsymbol{E}\boldsymbol{x}^{(r)} = \boldsymbol{f}(\boldsymbol{x}^{(0\sim r-1)}) + \boldsymbol{B}(\boldsymbol{x}^{(0\sim r-1)})\boldsymbol{u} \tag{8}$$

where $\boldsymbol{E}$ is a constant, nonsingular coefficient matrix, $\boldsymbol{x} \in \mathrm{R}^{m}$ represents the system state vector, $\boldsymbol{x}^{(r)}$ denotes the r-th order derivative of $\boldsymbol{x}$, and $\boldsymbol{x}^{(0\sim r-1)} = [\boldsymbol{x}\ \ \dot{\boldsymbol{x}}\ \ \cdots\ \ \boldsymbol{x}^{(r-1)}]^{T}$. Here, $\boldsymbol{f} \in \mathrm{R}^{m}$ is the state vector function, $\boldsymbol{B} \in \mathrm{R}^{m\times n}$ is the input matrix, and $\boldsymbol{u} \in \mathrm{R}^{n}$ denotes the control input. The system is considered fully actuated when it satisfies the following condition:

$$m = n \ \&\ \det \boldsymbol{B} \neq 0, (\forall \boldsymbol{x}^{(i)} \in \mathrm{R}^{m}, i = 0 \sim m-1) \tag{9}$$

This condition implies that the rank of matrix $\boldsymbol{B}$ equals the state dimension, ensuring that $\boldsymbol{B}$ always maintains full row

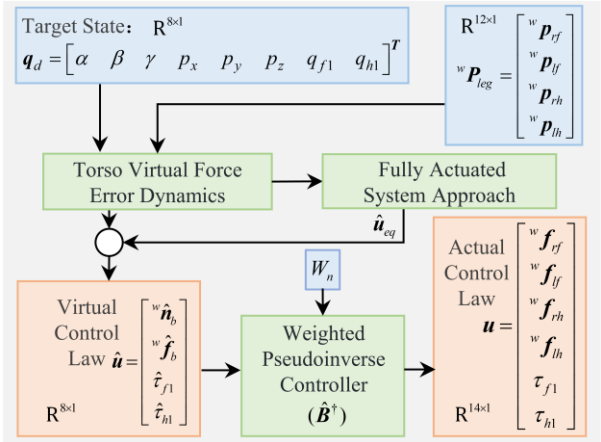

Fig. 2 Operational framework of the proposed VF-FASA. The controller leverages the desired state ($\boldsymbol{q}_{d}$, ${}^{w}\boldsymbol{p}_{leg}$) and torso error dynamics to generate feedforward and feedback ($\hat{\boldsymbol{u}}_{eq}$) virtual control laws. The combined virtual control law ($\hat{\boldsymbol{u}}$) is mapped into actual joint torques and foot contact forces ($\boldsymbol{u}$) via weighted pseudoinverse allocation ($\hat{\boldsymbol{B}}^{+}$), achieving coordinated motion under system redundancy and dynamic coupling.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

rank. Consequently, the input can independently influence the highest-order derivatives of all state variables. Conversely, the system is over-actuated if $\text{rank}\boldsymbol{B} = m$ and $m < n$. According to [24], we can establish the following control laws directly:

$$\begin{aligned} \boldsymbol{u} &= -[\boldsymbol{B}]^{-1}[\boldsymbol{A}_{(0\sim r-1)}\boldsymbol{x}^{(0\sim r-1)} + \boldsymbol{f}] \\ &\begin{cases} \boldsymbol{A}_{(0\sim r-1)} = [\boldsymbol{A}_0, \boldsymbol{A}_1 \cdots, \boldsymbol{A}_{r-1}] \\ \boldsymbol{A}_i \in \mathrm{R}^{m\times m}(i = 1\sim r-1) \end{cases} \end{aligned} \tag{10}$$

By substituting (10) into system (8), we obtain a first-order pseudo-linear system:

$$\dot{\boldsymbol{x}}^{(0\sim r-1)} = \boldsymbol{\Phi}(\boldsymbol{A}_{(0\sim r-1)})\boldsymbol{x}^{(0\sim r-1)} \tag{11}$$

$$\boldsymbol{\Phi}(\boldsymbol{A}_{(0\sim r-1)}) = \begin{bmatrix} 0 & \boldsymbol{I} & & \\ \vdots & & \ddots & \\ 0 & & & \boldsymbol{I} \\ -\boldsymbol{A}_0 & -\boldsymbol{A}_1 & \cdots & \boldsymbol{A}_{r-1} \end{bmatrix} \tag{12}$$

The system is asymptotically stable if the coefficient matrix $\boldsymbol{\Phi} \in \mathrm{R}^{rm\times rm}$ is Hurwitz [23]. $\boldsymbol{I}$ denotes identity matrix. Matrix $\boldsymbol{A}_i$ serves as the eigenvalue matrix for $\boldsymbol{\Phi}(\boldsymbol{A}_{0\sim r-1})$. The control problem is transformed into designing matrix $\boldsymbol{A}_i$ to attain the desired closed-loop system performance. Consequently, we introduce a parameter matrix $\boldsymbol{F} \in \mathrm{R}^{rm\times rm}$ and an arbitrary parameter matrix $\boldsymbol{Z} \in \mathrm{R}^{m\times rm}$, which are defined set as follows:

$$\boldsymbol{V}(\boldsymbol{Z},\boldsymbol{F}) = [\boldsymbol{Z} \quad \boldsymbol{ZF} \quad \cdots \quad \boldsymbol{ZF}^{r-1}]^{\boldsymbol{T}} \in \mathrm{R}^{rm\times rm} \tag{13}$$

$$\boldsymbol{\Phi}(\boldsymbol{A}_{0\sim r-1}) = \boldsymbol{VFV}^{-1} \tag{14}$$

$$\boldsymbol{A}_{0\sim r-1} = -\boldsymbol{ZF}^{r}\boldsymbol{V}^{-1} \tag{15}$$

where matrices $\boldsymbol{V}$ and $\boldsymbol{F}$ must satisfy the following condition to ensure $\boldsymbol{\Phi}$ is a Hurwitz matrix:

$$\begin{cases} \det \boldsymbol{V}(\boldsymbol{Z},\boldsymbol{F}) \neq 0 \\ \mathrm{Re}\,\lambda_i(\boldsymbol{F}) < 0, i = 1,2\cdots mr \end{cases} \tag{16}$$

This condition enhances the control system design flexibility. By setting matrix $\boldsymbol{F}$ as diagonal and ensuring that matrix $\boldsymbol{Z}$ satisfies $\det \boldsymbol{V}(\boldsymbol{Z},\boldsymbol{F}) \neq 0$, system stability is guaranteed. The parameters of $\boldsymbol{F}$ are optimized using the control parametrization method [28].

B. Virtual Force Fully-Actuated Control

We now synthesize the controller by casting the system into the FASA form. Equation (7) is transformed into its error dynamics and formulated in the standard FASA form (8), from which the control law is directly obtained according to the general law (10).

Let the desired trajectories be $\boldsymbol{q}_d$. The tracking errors are defined as $\boldsymbol{e} = \boldsymbol{q} - \boldsymbol{q}_d$. Taking the time derivative gives:

$$\begin{cases} \dot{\boldsymbol{e}} = \dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_d \\ \ddot{\boldsymbol{e}} = \ddot{\boldsymbol{q}} - \ddot{\boldsymbol{q}}_d \end{cases} \tag{17}$$

Substituting (17) into the (7) yields the following result:

$$\boldsymbol{M}\ddot{\boldsymbol{e}} + \boldsymbol{C}\dot{\boldsymbol{e}} = \hat{\boldsymbol{u}} - [\boldsymbol{M}\ddot{\boldsymbol{q}}_d + \boldsymbol{C}\dot{\boldsymbol{q}}_d + \boldsymbol{G} - \boldsymbol{f}_r] \tag{18}$$

The primary task of the controller is to apply the virtual control law ($\hat{\boldsymbol{u}}$) to drive the error ($\boldsymbol{e}$) converge to zero. The virtual control law can be designed as:

$$\hat{\boldsymbol{u}} = \underbrace{\hat{\boldsymbol{u}}_{eq}}_{\text{Feedback Term}} + \underbrace{[\mathbf{M}\ddot{\boldsymbol{q}}_d + \mathbf{C}\dot{\boldsymbol{q}}_d + \mathbf{G} - \boldsymbol{f}_r]}_{\text{Feedforward Compensation}} \tag{19}$$

The virtual error control law, denoted as $\hat{\boldsymbol{u}}_{eq}$, serves as the feedback term, while the feedforward term compensates for the known components of the target state dynamics. Substituting (19) into (18) yields:

$$\mathbf{M}\ddot{\boldsymbol{e}} + \mathbf{C}\dot{\boldsymbol{e}} = \hat{\boldsymbol{u}}_{eq} \tag{20}$$

When the robot is in a non-singular configuration and the matrix $\mathbf{M}$ is invertible, we rewrite (20) in a standard second-order fully-actuated form similar to (8).

$$\ddot{\boldsymbol{e}} = \underbrace{-\mathbf{M}^{-1}[\mathbf{C}\dot{\boldsymbol{e}}]}_{\boldsymbol{f}} + \underbrace{\mathbf{M}^{-1}}_{\boldsymbol{B}}\hat{\boldsymbol{u}}_{eq} \tag{21}$$

According to (10), we directly obtain the control law:

$$\hat{\boldsymbol{u}}_{eq} = -\boldsymbol{B}^{-1}([\boldsymbol{A}_0 \quad \boldsymbol{A}_1]\begin{bmatrix}\boldsymbol{e} \\ \dot{\boldsymbol{e}}\end{bmatrix} + \boldsymbol{f}) \tag{22}$$

By substituting the control law (22) into the system model (21), we design the matrices $\boldsymbol{A}_0$ and $\boldsymbol{A}_1$ according to (13)-(15) to ensure the asymptotic convergence of the state error vector $[\boldsymbol{e} \quad \dot{\boldsymbol{e}}]^{\boldsymbol{T}} \in \mathrm{R}^{16\times 1}$ to zero.

FASA eliminates the process of constructing first-order state-space equations and directly yields the control law. The design process only requires specifying the parameter matrix $\boldsymbol{F} = \mathrm{diag}(\kappa_1, \kappa_2, \cdots, \kappa_{16})$ according to the desired performance, with dominant poles satisfying $\kappa_i < 0 \ \ (i = 1,2,\cdots,16)$. Then, the parameter matrix $\boldsymbol{Z} = [\boldsymbol{I}^{8\times 8} \quad \boldsymbol{I}^{8\times 8}]$ is configured to satisfy the non-singularity condition $\det \boldsymbol{V}(\boldsymbol{Z},\boldsymbol{F}) \neq 0$.

C. Weighted Virtual Forces Allocation

The virtual control law (19) is implemented through weighted allocation to the actual torso joint torques and foot contact forces. The foot positions in frames $\{f1\}$ and $\{h1\}$ are denoted as ${}^{f1}\boldsymbol{P}_{leg}(leg = rf, lf)$ and ${}^{h1}\boldsymbol{P}_{leg}(leg = rh, lh)$, respectively. From the base attitude ${}^{w}\boldsymbol{R}_b$, position ${}^{w}\boldsymbol{P}_b$, and torso kinematics (${}^{b}\boldsymbol{P}_{f1}$, ${}^{b}\boldsymbol{R}_{f1}$, ${}^{b}\boldsymbol{P}_{h1}$, ${}^{b}\boldsymbol{R}_{h1}$), the foot positions in frame $\{w\}$ are computed as:

$${}^{w}\boldsymbol{P}_{leg} = \begin{cases} {}^{w}\boldsymbol{P}_b + {}^{w}\boldsymbol{R}_b({}^{b}\boldsymbol{P}_{f1} + {}^{b}\boldsymbol{R}_{f1}\,{}^{f1}\boldsymbol{P}_{leg}) \\ {}^{w}\boldsymbol{P}_b + {}^{w}\boldsymbol{R}_b({}^{b}\boldsymbol{P}_{h1} + {}^{b}\boldsymbol{R}_{h1}\,{}^{h1}\boldsymbol{P}_{leg}) \end{cases} \tag{23}$$

where $leg = \{rf, lf, rh, lh\}$ denote the four legs, and the subscript "leg" indicates variables defined for each leg. Taking partial derivatives of the equation concerning state $\boldsymbol{q}$ in sequence yields the partial differential matrix:

$$\boldsymbol{J}_{leg}(\boldsymbol{q}) = \frac{\partial\,{}^{w}\boldsymbol{P}_{leg}}{\partial \boldsymbol{q}} \tag{24}$$

$${}^{w}\dot{\boldsymbol{P}}_{leg} = \boldsymbol{J}_{leg}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{25}$$

where $\boldsymbol{J}_{leg}(\boldsymbol{q}) \in \mathrm{R}^{3\times 8}$ denotes the Jacobian matrix. Equation (25) describes the dependence of ${}^{w}\boldsymbol{P}_{leg}$ variations on state $\boldsymbol{q}$. The resultant force ${}^{w}\boldsymbol{f}_{leg} = [{}^{w}f_{leg}^{x} \quad {}^{w}f_{leg}^{y} \quad {}^{w}f_{leg}^{z}]$ is defined as the vector sum of vertical and friction forces at each foot contact. At equilibrium, according to the principle of virtual work, the work of the foot contact forces equals that of the virtual forces, which can be expressed as:

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

$$ {}^{w}\boldsymbol{f}_{leg}^{T}\delta\boldsymbol{\chi} = \hat{\boldsymbol{u}}_{leg}^{T}\delta\Theta \tag{26} $$

where $\delta\boldsymbol{\chi}$ denotes the virtual displacement of the foot position, $\hat{\boldsymbol{u}}_{leg}$ represents the single-leg virtual force, and $\delta\Theta$ is the virtual displacement corresponding to the state $\boldsymbol{q}$. Combining (25) with (26) yields the mapping from a single-leg contact force to virtual forces:

$$ \hat{\boldsymbol{u}}_{leg} = \boldsymbol{J}_{leg}^{T}\,{}^{w}\boldsymbol{f}_{leg} \tag{27} $$

We aggregate all foot contact forces and torso joint torques into an actual control input vector $\boldsymbol{u}$, and assemble the corresponding Jacobian transpose mappings into a matrix $\hat{\boldsymbol{B}}$:

$$ \boldsymbol{u} = [\,{}^{w}\boldsymbol{f}_{rf} \quad {}^{w}\boldsymbol{f}_{lf} \quad {}^{w}\boldsymbol{f}_{rh} \quad {}^{w}\boldsymbol{f}_{lh} \quad \tau_{f1} \quad \tau_{h1}\,]^{T} \in \mathrm{R}^{14\times 1} \tag{28} $$

$$ \hat{\boldsymbol{B}} = [\boldsymbol{J}_{rf}^{T} \quad \boldsymbol{J}_{lf}^{T} \quad \boldsymbol{J}_{rh}^{T} \quad \boldsymbol{J}_{lh}^{T} \quad \boldsymbol{E}_{\tau}\,] \in \mathrm{R}^{8\times 14} \tag{29} $$

where $\boldsymbol{E}_{\tau} = [0^{2\times 6} \quad \boldsymbol{I}^{2\times 2}]^{T}$. The real inputs map to the virtual control law through Jacobian transpose, yielding the total virtual control law $\hat{\boldsymbol{u}}$:

$$ \hat{\boldsymbol{u}} = \sum \hat{\boldsymbol{u}}_{leg} + \hat{\tau}_{f1} + \hat{\tau}_{h1} = \hat{\boldsymbol{B}}\boldsymbol{u} \tag{30} $$

Direct inversion of matrix $\hat{\boldsymbol{B}}$ to compute the mapping from $\hat{\boldsymbol{u}}$ to $\boldsymbol{u}$ is infeasible. Although the pseudo-inverse of $\hat{\boldsymbol{B}}$ can accomplish this mapping, such an approach is unbiased and fails to exploit the system's redundancy. To address this issue, we formulate a constrained quadratic problem that minimizes a weighted norm of input $\boldsymbol{u}$ subject to the linear mapping constraint (30) and actual input constraints:

$$ \begin{aligned} \min_{\boldsymbol{u}} \quad & \boldsymbol{u}^{T}\boldsymbol{W}_{n}\boldsymbol{u} \\ \text{s.t.} \quad & \hat{\boldsymbol{B}}\boldsymbol{u} = \hat{\boldsymbol{u}},\ 0 < {}^{w}f_{leg}^{z} < {}^{w}f_{leg}^{\max} \\ & \left|{}^{w}f_{leg}^{x}\right| < \mu\,{}^{w}f_{leg}^{z},\ \left|{}^{w}f_{leg}^{y}\right| < \mu\,{}^{w}f_{leg}^{z} \\ & |\tau_i| < \tau_i^{\max}\ \ (i = f1, h1) \end{aligned} \tag{31} $$

Here, $\boldsymbol{W}_{n} = \mathrm{diag}(w_1, w_2, \cdots, w_{14})$ denotes the input weight matrix. ${}^{w}f_{leg}^{\max}$ and $\tau_i^{\max}$ denote the maximum foot contact force and joint torque, respectively. μ represents the static friction between the foot and the ground. The constrained optimization problem is solved using the method of Lagrange multipliers, resulting in the following Lagrangian function:

$$ L(\boldsymbol{u}, \boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{u}^{T}\boldsymbol{W}_{n}\boldsymbol{u} + \boldsymbol{\lambda}^{T}(\hat{\boldsymbol{B}}\boldsymbol{u} - \hat{\boldsymbol{u}}) \tag{32} $$

where $\boldsymbol{\lambda}$ denotes the Lagrange multiplier vector. We solve the Lagrange function:

$$ \frac{\partial L}{\partial \boldsymbol{u}} = \boldsymbol{W}_{n}\boldsymbol{u} + \hat{\boldsymbol{B}}^{T}\boldsymbol{\lambda} = 0 \tag{33} $$

$$ \boldsymbol{u} = -\boldsymbol{W}_{n}^{-1}\hat{\boldsymbol{B}}^{T}\boldsymbol{\lambda} \tag{34} $$

$$ \frac{\partial L}{\partial \boldsymbol{\lambda}} = -\hat{\boldsymbol{B}}(\boldsymbol{W}_{n}^{-1}\hat{\boldsymbol{B}}^{T}\boldsymbol{\lambda}) - \hat{\boldsymbol{u}} = 0 \tag{35} $$

$$ \boldsymbol{\lambda} = -(\hat{\boldsymbol{B}}\boldsymbol{W}_{n}^{-1}\hat{\boldsymbol{B}}^{T})^{-1}\hat{\boldsymbol{u}} \tag{36} $$

Substituting (36) into (34) yields the weighted optimal mapping from $\hat{\boldsymbol{u}}$ to $\boldsymbol{u}$ and derives the explicit expression for the weighted pseudoinverse of $\hat{\boldsymbol{B}}$, denoted as $\hat{\boldsymbol{B}}^{\dagger}$.

$$ \boldsymbol{u} = \boldsymbol{W}_{n}^{-1}\hat{\boldsymbol{B}}^{T}(\hat{\boldsymbol{B}}\boldsymbol{W}_{n}^{-1}\hat{\boldsymbol{B}}^{T})^{-1}\hat{\boldsymbol{u}} \tag{37} $$

$$ \hat{\boldsymbol{B}}^{\dagger} = \boldsymbol{W}_{n}^{-1}\hat{\boldsymbol{B}}^{T}(\hat{\boldsymbol{B}}\boldsymbol{W}_{n}^{-1}\hat{\boldsymbol{B}}^{T})^{-1} \tag{38} $$

To ensure $(\hat{\boldsymbol{B}}\boldsymbol{W}_{n}^{-1}\hat{\boldsymbol{B}}^{T})$ is invertible, $\boldsymbol{W}_{n}$ must be strictly positive definite and $\hat{\boldsymbol{B}}$ must satisfy the full rank condition. When the robot is in a non-singular configuration, $(\hat{\boldsymbol{B}}\boldsymbol{W}_{n}^{-1}\hat{\boldsymbol{B}}^{T})$ is positive definite, ensuring invertibility and validating the weighted pseudoinverse. However, in singular configurations, the rank of $\hat{\boldsymbol{B}}$ decreases, $|\,\hat{\boldsymbol{B}}\boldsymbol{W}_{n}^{-1}\hat{\boldsymbol{B}}^{T}\,|$ approaches zero, and the output tends to infinity. Since joint angles are physically constrained, the robot cannot reach singular configurations. By adjusting $\boldsymbol{W}_{n}$, we can satisfy optimal force distribution requirements for different states.

IV. Numerical Simulations

We implement a simplified dynamics model of the redundant-torso quadruped robot in MATLAB to evaluate the proposed VF-FASA framework (Fig. 3). The robot parameters are listed in the Table. II. and Supplementary Material Video S1 and S2 provide the simulation animation.

To simulate real-world uncertainties, we inject additive Gaussian noise into the following signals: foot position, foot contact force, joint torque, base attitude, base position, and torso joint angles. The corresponding standard deviations are 5 mm, 2 N, 1 N·m, 1°, 10 mm, and 0.1°, respectively. Two noise intensity levels are defined, high and low, corresponding to 1 and 0.1 times the standard deviation, respectively. The non-diagonal elements of the inertia matrix are neglected in the controller's dynamic model. Simulation evaluates three typical scenarios: standing (z-direction: 0-200 mm), lateral disturbance (100 N impact force in the y-direction for 0.5 s), and base sine trajectory tracking while standing. Reference trajectories are assumed to be provided by a high-level planner. A fixed time step of 0.002 s is used, with standing starting at 0.25 s, the lateral impact is applied from 1 to 1.5 s, and trajectory tracking performed from 2

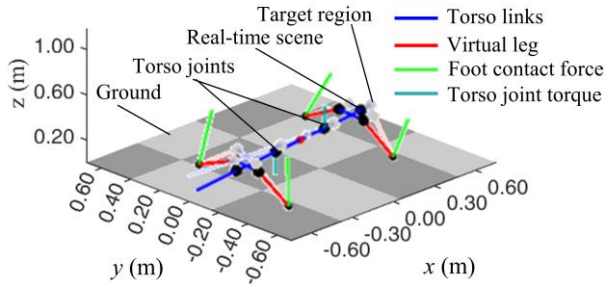

Fig. 3 Simplified quadruped robot dynamics simulation model.

TABLE. II Structural Parameters of Robot

Link	Length (mm)	Mass (kg)	CoM position ${}^{i}\boldsymbol{r}_{i,c}$ (mm)	Inertia matrix (kg·m²×10⁻³)	Foot positions ${}^{w}\boldsymbol{p}_{leg}$ (mm)	Joint damping
b_{f1}	183.5	6.5	(-17.15, 4.70, 0)	$\boldsymbol{I}_{f1}$=[25.42, 0.61, 0; 0.61, 136.42, -0.01; 0, -0.01, 1288.45]	(256, ±219, 0)	c_{f1} : 0.007
b	240	6.5	(-0.99, 0, 11.46)	$\boldsymbol{I}_{b}$=[53.01, 0, -0.42; 0, 38.92, 0; -0.42, 0, 60.7]	/	/
b_{h1}	196.5	7.6	(4.54, 4.56, 0)	$\boldsymbol{I}_{h1}$=[30.99, 10.99, -0.06; 10.99, 417.08, 0; -0.06, 0, 407.2]	(-266, ±221.8, 0)	c_{h1} : 0.007

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

to 4 s. Additionally, robot locomotion was demonstrated at two gait frequencies on flat and rugged terrain with two ground friction coefficients, as well as on planar and rugged 20° slopes.

For baseline comparison, the Model-Free PD Controller (MF-PD) and the PD Controller with Dynamic Compensation (DC-PD) are also evaluated. Additionally, a Virtual Force-based MPC controller (VF-MPC) is included, which generates virtual control forces using the simplified torso model (7) and allocates them through the weighted pseudoinverse $\hat{\boldsymbol{B}}^{\dagger}$. Its cost function considers the state error, its derivative, and changes in control outputs between steps.

The key and boundary parameters of the VF-FASA controller are configured as follows:

$\boldsymbol{F} = -\text{diag}(41.13, 20.80, 32.58, 15.51, 15.21, 15.42, 30.57, 25.30, 33.19, 27.79, 31.63, 32.13, 32.12, 32.13, 35.70, 34.82)$,

$\mu = 0.5$, ${}^{w}f_{leg}^{\max} = 300\ \text{N}$, $\tau_i^{\max} = 100\ \text{N}\cdot\text{m}$.

The PD gains for MF-PD and DC-PD, and the optimized parameters for VF-MPC are manually tuned for optimal tracking performance while ensuring stability. The key controller parameters are configured as follows:

MF-PD: $\boldsymbol{k}_p = [120,120,140,3000,2800,5200,110,110]$,
$\boldsymbol{k}_d = [10,10,10,300,300,800,10,10]$.

DC-PD: $\boldsymbol{k}_p = [120,120,140,3000,2800,5200,110,110]$,
$\boldsymbol{k}_d = [10,10,10,300,300,400,10,10]$.

VF-MPC: The prediction/control horizons ($\boldsymbol{N}_p = 8$, $\boldsymbol{N}_c = 3$).

V. Results and Discussion

A. State Analysis

Figure 4-5 present the tracking results. Figure 4 shows the base attitude and displacement under three typical scenarios (i-iii) and four controllers in low noise conditions with $\boldsymbol{W}_n = \boldsymbol{I}_{14}$. VF-FASA achieves rapid standing stabilization (Fig. 4(a)). The vertical displacement p_z rises from 0 to 200 mm within 1 mm error in approximately 0.436 s, comparable to DC-PD, faster than VF-MPC (0.942 s, steady-state 185 mm), and 47.6% faster than MF-PD (0.832 s, 167 mm). Planar displacements (p_x, p_y) and attitudes remain within tight bounds (±0.5 mm, ±0.3°) without observable oscillations.

During a lateral impact (ii), VF-FASA shows a peak lateral deviation of 54.2 mm in p_y and recovers within 2 mm in 0.3 s, comparable to VF-MPC, outperforming MF-PD (87.7 mm, 0.33 s) and DC-PD (89.2 mm, 0.32 s), achieving up to 38.2% improvement in disturbance rejection with negligible reverse overshoot (Fig. 4(b)). For base roll angle (Fig. 4(c)), VF-FASA shows a 15.8° overshoot, compared to 11.3° for MF-PD, 15.1° for DC-PD and 7.7° for VF-MPC. After the impact ends, VF-FASA returns within 0.5° in 0.2 s, faster than VF-MPC (0.26 s), MF-PD (0.26 s) and DC-PD (0.37 s). During trajectory tracking (iii), VF-FASA shows comparable displacement tracking performance to DC-PD and superior height maintenance in the z-direction compared to MF-PD and VF-MPC (Fig. 4(a)-(b)), while significantly outperforming both PD controllers in all attitude angles and exhibits smaller lag than VF-MPC (Fig. 4(c)-(d)).

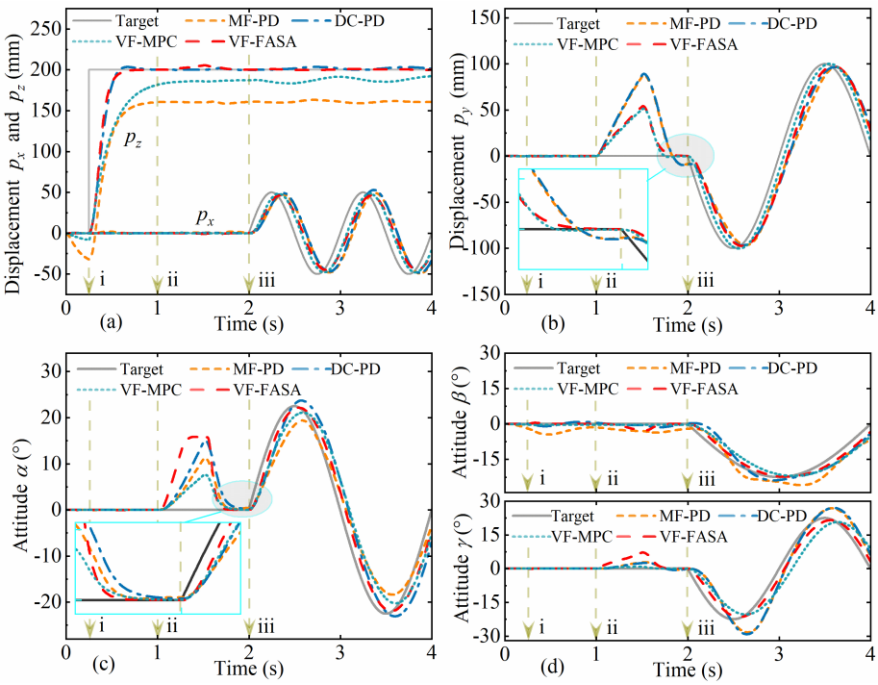

Fig. 4 Time-domain response of the base state under different controllers. Experimental phases: (i) Standing (0.25 s), (ii) Lateral impact (1-1.5 s), (iii) Tracking (2-4 s). (a) Vertical (p_z) and forward (p_x) displacements. (b) Lateral displacement (p_y). (c) Roll angle (α). (d) Pitch (β) and yaw angles (γ).

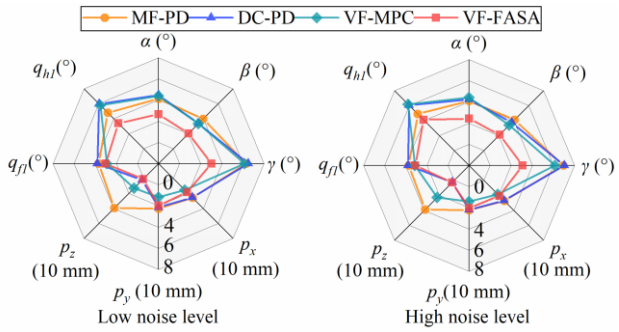

Fig. 5 Radar chart of root mean square error (RMSE) for state tracking under different controllers during 2-4 s.

Figure 5 presents the root mean square error (RMSE) charts of robot state tracking under both noise levels from 2 to 4 s. The VF-FASA regions are consistently closer to the center, indicating slightly better accuracy in base attitude and rear torso joint angle, while performance in other states remains comparable to the baseline controllers. This consistency persists under increased noise, demonstrating robust performance.

B. Virtual Force and Actual Force

Figure 6 presents the virtual control laws generated by VF-FASA and their allocation to the torso joint torques and the vertical foot contact forces. During standing stabilization (i), lateral disturbance rejection (ii), and trajectory tracking (iii), VF-FASA generates coordinated virtual forces to regulate vertical motion, counteract lateral

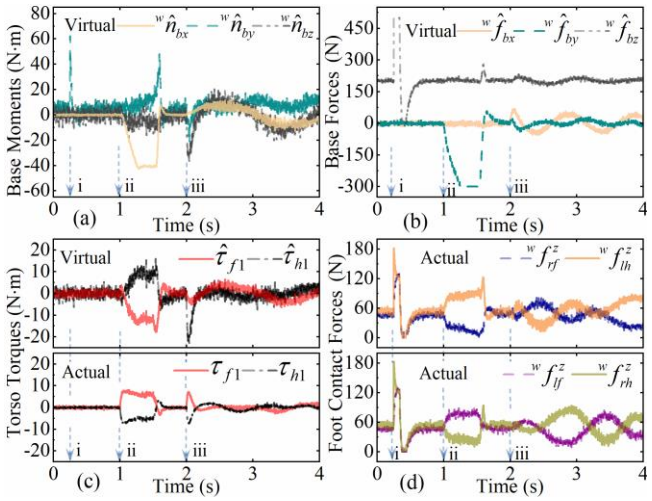

Fig. 6 Virtual control law and actual control forces. Experimental phases: (i) Standing, (ii) Lateral impact, (iii) Tracking. (a) Base virtual moments. (b) Base virtual forces. (c) Torso virtual and actual joint torques. (d) Vertical actual foot contact forces.

disturbances, and realize smooth trajectory tracking. These virtual forces are weighted and mapped into torso joint torques and foot contact forces to realize body support. During lateral impact rejection, the left front leg force rises to approximately 80 N, while the right front leg force drops to about 15 N. In addition, Fig. 6(c) illustrates that the torso joint torques are not directly applied but optimally redistributed through weighted ratios to produce smoother joint torques.

C. Redundancy and Fault Tolerance

Figure 7 illustrates the effects of weight matrix adjustment on the vertical foot contact forces and torso joint torques. During 2-4 s, the weights of the right front leg (w_1, w_2, w_3) and torso joints (w_{13}, w_{14}) in $\boldsymbol{W}_n$ are increased from 1 to 100, respectively. Simultaneously, the base position is shifted by -40 mm in the x-direction and +40 mm in the y-direction, keeps the robot's overall CoM within the leg support polygon and preserving the full row-rank condition for matrix $\hat{\boldsymbol{B}}$.

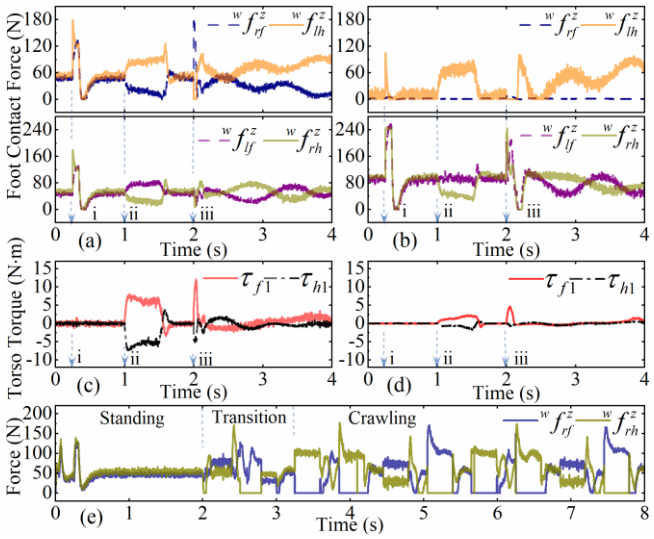

Fig. 7 Force distribution under selective weight tuning ($\boldsymbol{W}_n$). Experimental phases: (i) Standing, (ii) Lateral impact, (iii) Tracking. (a) Vertical foot contact forces with $\boldsymbol{W}_n = \boldsymbol{I}_{14}$. (b) Vertical foot contact forces with $w_{1:3} = 100$ for the right front leg. (c) Torso torque with $\boldsymbol{W}_n = \boldsymbol{I}_{14}$. (d) Torso torques with $w_{13:14} = 100$. (e) Vertical foot contact forces of the right front and right hind legs during crawling.

As shown in Fig. 7(a)-(b), the vertical force on the right front leg (${}^{w}f_{rf}^{z}$) is reduced by 98.6% from 175 N to 2.45 N. To maintain overall balance, the controller redistributes the contact forces, decreasing the load on the left hind leg while increasing those on the left front and right hind legs. Correspondingly, Fig. 7(c)-(d) show a 62% reduction in peak torso joint torque (τ_{f1}) from 12.1 N·m to 4.6 N·m, indicating effective torque mitigation through weight modulation. These results show that the proposed VF-FASA enables fault-tolerant control through weighted force redistribution, allows regulation of torso joint torques by constraining peak demands via adaptive weighting, and enhances gait adaptability by redistributing forces among stance limbs.

Figure 7(e) shows the vertical foot contact forces of the two right legs during crawling on flat ground with a 2.4 s gait period and 0.22m stride, including standing, gait transition, and crawling phases. By adjusting the weights of each leg, the controller dynamically modulates the foot forces to follow the desired trajectory. Video S2 shows that the controller maintains consistent gait and stable trajectory tracking under variations in gait period, stride length, terrain, and ground friction. The visualized force distribution confirms that the framework dynamically adapts to ground topology and friction, enabling robust locomotion.

D. Computational Efficiency

Table. III lists the average simulation time over the interval 0-4 s (Intel i7-9700, 3 GHz), excluding the initial 10 warm-up iterations. A 250 Hz benchmark is used as it matches the typical range of mid-level controllers in quadrupeds and balances performance with computational cost. The VF-FASA achieves computation times comparable to DC-PD (approximately 3.14 ms, std: 0.53 ms), and maintains a 250 Hz control frequency in 92.7% of control cycles. In contrast, VF-MPC shows much longer average computation time (>800 ms) due to the nonlinear and nonconvex nature of its optimization problem, where solver convergence heavily depends on the initial guesses and constraint activation. This variability results in unstable real-time performance, especially with complex prediction models (e.g., torso dynamics) or constraints. VF-FASA thus provides a higher and more stable control update rate, better meeting real-time requirements in legged robotics.

The performance advantage stems from VF-FASA's theory-based force mapping, which eliminates the need to solve large-scale optimization problems. Although the algorithm introduces additional steps, the core virtual force generation and optimization processes remain structured and efficient. This characteristic is critical for robotic control applications.

TABLE. III COMPARISON OF COMPUTATIONAL TIME CONSUMPTION AND REAL-TIME PERFORMANCE

Control Method	Average Computation Time (ms)	Sample Standard Deviation (ms)	Control Cycle Achievement Rate (> 250 Hz)
MF-PD	0.207	0.038	100%
DC-PD	3.135	0.531	92.9%
VF-MPC	808.509	560.166	0%
VF-FASA	3.143	0.539	92.7%

VI. Conclusion

This paper presents VF-FASA, a virtual force-based control framework that extends FASA to quadruped robots with redundant actuation. By introducing virtual forces to construct full-actuation conditions and formulating a matrix-weighted pseudoinverse for force/torque allocation, the method enables coordinated control of both the floating base and torso joints. Simulations show that VF-FASA outperforms MF-PD, DC-PD, and VF-MPC controllers, achieving faster vertical stabilization, up to 38.2% better disturbance rejection, and lower RMSE in trajectory tracking. VF-FASA also maintains computation times comparable to DC-PD while offering a faster and more stable control frequency than VF-MPC. Although the framework effectively leverages structural redundancy for performance and robustness, its output smoothness remains sensitive to noise. Future work will integrate disturbance observers and robust control under the fully actuated paradigm and implement VF-FASA on a physical quadruped for real-world validation.

Supplementary Material

Video S1: Comparison of state tracking performance and force distribution under selective weight tuning.

Video S2: Comparison of crawling performance on different terrains and different ground friction coefficients.

References

[1] A. J. Ijspeert, A. Crespi, D. Ryczko, and J.-M. Cabelguen, "From Swimming to Walking with a Salamander Robot Driven by a Spinal Cord Model," *Science,* vol. 315, no. 5817, pp. 1416-1420, 2007.

[2] A. Crespi, K. Karakasiliotis, A. Guignard, and A. J. Ijspeert, "Salamandra Robotica II: An Amphibious Robot to Study Salamander-Like Swimming and Walking Gaits," *IEEE Transactions on Robotics,* vol. 29, no. 2, pp. 308-320, 2013.

[3] K. Melo, T. Horvat, and A. J. Ijspeert, "Animal robots in the African wilderness: Lessons learned and outlook for field robotics," *Science Robotics,* vol. 8, no. 85, p. eadd8662, 2023.

[4] G. Chen, L. Qiao, Z. Zhou, L. Richter, and A. Ji, "Development of a Lizard-Inspired Robot for Mars Surface Exploration," *Biomimetics-Basel,* vol. 8, no. 1, 2023.

[5] S. Dong, F. Fan, Y. Chen, S. Guo, and J. Liu, "Gait Planning, and Motion Control Methods for Quadruped Robots: Achieving High Environmental Adaptability: A Review," *Computer Modeling in Engineering & Sciences* vol. 143, no. 1, 2025.

[6] S. Soyguder, and H. Alli, "Computer simulation and dynamic modeling of a quadrupedal pronking gait robot with SLIP model," *Computers & Electrical Engineering,* vol. 38, no. 1, pp. 161-174, 2012.

[7] J. Cheng, Y. G. Alqaham, and Z. Gan, "Harnessing Natural Oscillations for High-Speed, Efficient Asymmetrical Locomotion in Quadrupedal Robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* pp. 6864–6869, 2024.

[8] 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.

[9] J. ZHAO, S. GONG, S. MA, and J. WANG, "Fractional-order virtual model control for single leg of hydraulic quadruped robot," *Transactions of Beijing institute of Technology,* vol. 42, no. 3, pp. 304-311, 2022.

[10] S. Fahmi, C. Mastalli, M. Focchi, and C. Semini, "Passive whole-body control for quadruped robots: Experimental validation over challenging terrain," *IEEE Robotics and Automation Letters,* vol. 4, no. 3, pp. 2553-2560, 2019.

[11] V. Klemm *et al.*, "LQR-assisted whole-body control of a wheeled bipedal robot with kinematic loops," *IEEE Robotics and Automation Letters,* vol. 5, no. 2, pp. 3745-3752, 2020.

[12] W. Chi, X. Jiang, and Y. Zheng, "A linearization of centroidal dynamics for the model-predictive control of quadruped robots," in *2022 International Conference on Robotics and Automation (ICRA)*, May 2022, pp. 4656-4663.

[13] 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," *Ieee Int C Int Robot,* pp. 7440-7447, 2018.

[14] Y. R. Ding, A. Pandala, C. Z. Li, Y. H. Shin, and H. W. Park, "Representation-Free Model Predictive Control for Dynamic Motions in Quadrupeds," *Ieee Transactions on Robotics,* vol. 37, no. 4, pp. 1154-1171, 2021.

[15] K. A. Hamed, J. Kim, and A. Pandala, "Quadrupedal locomotion via event-based predictive control and QP-based virtual constraints," *IEEE Robotics and Automation Letters,* vol. 5, no. 3, pp. 4463-4470, 2020.

[16] C. D. Bellicoso, F. Jenelten, P. Fankhauser, C. Gehring, J. Hwangbo, and M. Hutter, "Dynamic Locomotion and Whole-Body Control for Quadrupedal Robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* Vancouver, CANADA, Sep 24-28 2017, pp. 3359-3365.

[17] A. Xie, T. Chen, X. Rong, G. Zhang, Y. Li, and Y. Fan, "A robust and compliant framework for legged mobile manipulators using virtual model control and whole-body control," *Robotics and Autonomous Systems,* vol. 164, p. 104411, 2023.

[18] 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.

[19] X. Zhang, Q. Wang, S. Huang, and L. Jiang, "A Multi-model Fusion Based Complex Motion Control Approach for a Cheetah-mimicking Quadruped Robot," *ROBOT,* vol. 44, no. 6, pp. 682-693, 2022.

[20] J. Q. Zhang, F. Gao, X. L. Han, X. B. Chen, and X. Y. Han, "Trot Gait Design and CPG Method for a Quadruped Robot," *Journal of Bionic Engineering,* vol. 11, no. 1, pp. 18-25, 2014.

[21] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, "ANYmal parkour: Learning agile navigation for quadrupedal robots," *Science Robotics,* vol. 9, no. 88, p. eadi7566, 2024.

[22] G. R. Duan, "Fully Actuated System Approach for Control: An Overview," *IEEE transactions on cybernetics,* vol. PP, pp. 7285 - 7306, 2024.

[23] G. R. Duan, "High-order fully actuated system approaches: Part I. Models and basic procedure," *International Journal of Systems Science,* vol. 52, no. 2, pp. 422-435, 2021.

[24] G. R. Duan, "High-order fully-actuated system approaches: - part II. Generalized strict-feedback systems," *International Journal of Systems Science,* vol. 52, no. 3, pp. 437-454, 2021.

[25] G. R. Duan, "High-order fully actuated system approaches: Part III. Robust control and high-order backstepping," *International Journal of Systems Science,* vol. 52, no. 5, pp. 952-971, 2021.

[26] G. Cai, Y. Xiao, C. Hu, X. Yang, and Y. Fan, "Time-varying Output Constraint Control of High-order Strict-feedback Systems Based on Fully Actuated System Approach," *Acta Autom. Sin.,* vol. 50, no. 2, pp. 372-385, 2024.

[27] G. T. Tian, J. Tan, B. Li, and G. R. Duan, "Optimal Fully Actuated System Approach-Based Trajectory Tracking Control for Robot Manipulators," *Ieee Transactions on Cybernetics,* vol. 54, no. 12, pp. 7469-7478, 2024.

[28] G. Liu, B. Li, and G. Duan, "An Optimal FASA Approach for UAV Trajectory Tracking Control," *Guidance, Navigation and Control: An International Journal of Technical Committee on Guidance, Navigation and Control, CSAA,* vol. 3, no. 3, p. 23, 2023.

[29] H. ZHANG, W. WANG, S. CHEN, Y. JI, and J. LIU, "Integrated guidance and control design based on fully actuated system method," *Acta Aeronautica et Astronautica Sinica,* vol. 45, no. 01, pp. 137-149, 2024.

[30] M. Liu, R. Fan, S. Qiu, and X. Cao, "Spacecraft attitude-orbit prescribed performance control based on fully actuated system approach," *Acta Aeronautica et Astronautica Sinica,* vol. 45, no. 01, pp. 20-35, 2024.

[31] Z. Qin, and D. Guang-Ren, "Fully Actuated System Approach for 6DOF Spacecraft Control Based on Extended State Observer," *Journal of Systems Science & Complexity,* vol. 35, no. 2, pp. 604-622, 2022.

[32] G. R. Duan, "High-order System Approaches: II. Controllability and Full-actuation," *Acta Autom. Sin.,* vol. 46, no. 08, pp. 1571-1581, 2020.

[33] I. Agustian, N. Daratha, R. Faurina, A. Suandi, and Sulistyaningsih, "Robot Manipulator Control with Inverse Kinematics PD-Pseudoinverse Jacobian and Forward Kinematics Denavit Hartenberg," *Jurnal Elektronika dan Telekomunikasi,* vol. 21, pp. 8-18, 2021.