

# Adaptive Force-based Control for Legged Robots

Mohsen Sombolestan\*, Yiyu Chen\* and Quan Nguyen

**Abstract**— Adaptive control can address model uncertainty in control systems. However, it is preliminarily designed for tracking control. Recent advancements in the control of quadruped robots show that force control can effectively realize agile and robust locomotion. In this paper, we present a novel adaptive force-based control framework for legged robots. We introduce a new architecture in our proposed approach to incorporate adaptive control into quadratic programming (QP) force control. Since our approach is based on force control, it also retains the advantages of the baseline framework, such as robustness to uneven terrain, controllable friction constraints, or soft impacts. Our method is successfully validated in both simulation and hardware experiments. While the baseline QP control has shown a significant degradation in the body tracking error with a small load, our proposed adaptive force-based control can enable the 12-kg Unitree A1 robot to walk on rough terrains while carrying a heavy load of up to 6 kg (50% of the robot weight). When standing with four legs, our proposed adaptive control can even allow the robot to carry up to 11 kg of load (92% of the robot weight) with less than 5-cm tracking error in the robot height.

## I. INTRODUCTION

Legged robots have great potential for applications in disaster and rescue missions. In contrast to wheeled or tracked robots, legged robots represent remarkable performance for navigating uneven terrains. Designing and controlling machines to realize these potentials has motivated work across the legged robotics community and highly-capable quadrupeds (e.g., [13], [19], [7], [16]), beginning to assist humans in demanding situations.

Impulse-based gait design introduced in [20] can be used to achieve high-speed bounding for quadruped robots. Nevertheless, the approach was primarily designed for 2D motion. The recent development of model predictive control (MPC) approach for quadruped robots [8] utilizes convex optimization to solve for optimal ground reaction force. The approach is based on the simplified rigid body dynamics of the robot, enabling real-time computation for the controller. This framework has achieved a wide range of agile motion for 3D quadruped robots.

However, these controllers assume accurate knowledge of the dynamic model, or in other words, do not address substantial model uncertainty in the system. Many safety-critical missions, such as firefighting, disaster response, exploration, etc., require the robot to operate swiftly and

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

M. Sombolestan, Y. Chen, and Q. Nguyen are with the Department of Aerospace and Mechanical Engineering, University of Southern California, Los Angeles, CA 90089, email: sombole@usc.edu, yiyuc@usc.edu, quann@usc.edu.

\* These authors contributed equally to this work.



Fig. 1: Our proposed adaptive force-control approach is successfully validated in experiment on the 12-kg A1 robot walking while carrying an unknown load of up to 6 kg (50% of body weight). Experiment video: <https://youtu.be/tw107b-EhP4>

stably while dealing with high levels of uncertainty and large external disturbances. The demand for practical requirements motivates our research on adaptive control for quadruped robots.

The introduction of the  $L_1$  adaptive control technique has enabled the decoupling of adaptation and robustness in adaptive control techniques. In particular, applying a low-pass filter as part of the adaptation laws helps the  $L_1$  adaptive controller to guarantee not only stability [5] and transient performance [4] but also smooth control inputs, which can be critical for robotics applications. Our prior work on  $L_1$  adaptive control for bipedal robots [18] uses a control Lyapunov function (CLF) based controller to create a closed-loop nonlinear reference model for the  $L_1$  adaptive controller. However, the control paradigm in this prior work is based on Hybrid Zero Dynamics [11], which uses joint position control to track the desired trajectory from optimization for each robot joint.

In this work, we present a novel adaptive force-based control framework to utilize the advantages of force-based control [9] in dynamic legged robots including robustness to rough terrains, flexibility in obtaining a wide range of gaits, and soft impacts during locomotion [2]. Due to fundamental differences between trajectory tracking and force-based control, it requires the development of a new control architecture to integrate adaptive control into the force-based control framework. To the best of our knowledge, this is the first adaptive force-based controller successfully developed for quadruped robots. Our approach is successfully validated in both a high-fidelity simulation and hardware experiment. Although the baseline controller fails to maintain the robot balance under small model uncertainty, our proposed adap-

tive controller can satisfy the expectation while carrying an unknown load up to 50% of the robot weight (shown in Fig. 1). Thanks to the combination with the force-based controller, our approach can also allow the robot to navigate rough terrains while carrying an unknown and time-varying load.

The followings are the main contribution of the paper:

- We introduce a novel control architecture to incorporate adaptive control into the force-based control framework to adapt to significant model uncertainty of the system dynamics.
- Since our approach is based on force control, it retains critical advantages of the baseline framework, including soft impact, robustness to rough terrains, controllable friction constraints, and the flexibility in adapting to different locomotion gaits.
- We prove that our approach yields Input-to-State (ISS) stability for the control system.
- We successfully validate our approach in the simulation of a quadruped robot walking on an uneven, steep slope while carrying an unknown heavy load up to 50% of the robot weight and subjecting to unknown force disturbance to different parts of the body during walking.
- We successfully validate our approach on the real robot hardware of A1, a 12-kg quadruped robot. With our framework, the robot stands up and balances with unknown loads of up to 11 kg, which is 92% of the robot's weight. For comparison, the baseline non-adaptive controller can not even stand up with only 6 kg of load.
- We successfully validate our approach on the real robot hardware of the A1 robot walking stably while carrying unknown loads of up to 6 kg, which is 50% of the robot weight. For comparison, the baseline non-adaptive controller fails to control the robot to walk with only 3 kg of load.

The remainder of the paper is organized as follows. Sec. II presents the background on the force-based control architecture for quadruped robots. The proposed adaptive controller to compensate uncertainties is elaborated in Sec. III. Then, the stability proof of the whole system is described in Sec. IV. Furthermore, the numerical and experimental validation are shown in Sec. V and Sec. VI, respectively. Finally, Sec. VII provides concluding remarks.

## II. BACKGROUND

In this section, we present the background on force-based control of quadruped robots. The control architecture of the robot consists of several modules, [2] including high-level controller, low-level controller, state estimation, and gait scheduler as presented in Fig. 2. From user input and state estimation, a reference trajectory can be generated for high-level control. The gait scheduler defines the gait timing and sequence to switch between swing and stance phases for each leg. The high-level controller calculates position control for swing legs and force control for stance legs based on the

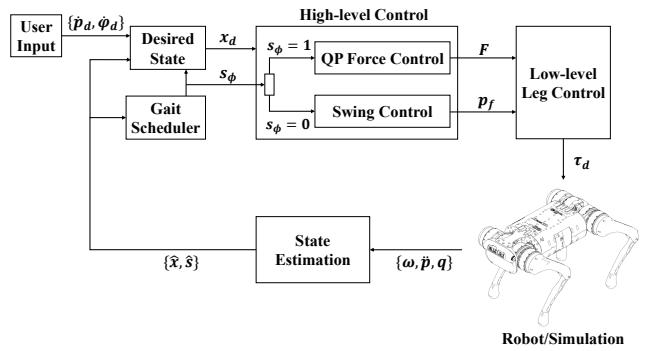


Fig. 2: **Control Architecture Overview.** Block diagram of control architecture for A1 robot.

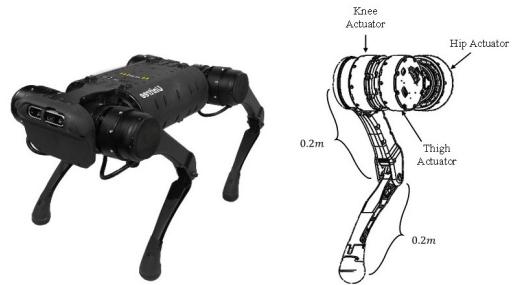


Fig. 3: **Robot Configuration.** Overview of A1 robot and leg configuration

user commands and gait timing. The low-level leg control converts the command generated by high-level control into joint torques for each motor. Each module of the control architecture will be elaborated in the following sections. The  $L_1$  adaptive controller is built on this baseline architecture and will be elaborated in Sec. III.

### A. Robot Specification

In this paper, we will validate our controller on Unitree A1, a mini dynamic quadruped robot (see Fig. 3). The A1 robot is weighted 12 kg and has low-inertial legs. The robot is equipped with high torque density electric motors using planetary gear reduction. It is capable of ground force control without using any force or torque sensors. The A1 robot uses these high-performance actuators for all the hip, thigh, and knee joints to enable full 3D control of ground reaction forces. It is also equipped with contact sensors on each foot which are used for contact detection.

Each of A1's actuators consists of a custom high torque density electric motor coupled to a single-stage 9:1 planetary gear reduction. The lower link is driven by a bar linkage that passes through the upper link. The legs are serially actuated, but to keep leg inertia low, the hip and knee actuators are co-axially located at the hip of each leg. Each robot joint has the maximum torque of 33.5 ( $Nm$ ) and the maximum speed of 21 ( $rad/s$ ).

### B. Gait Scheduler

The A1's gait is defined by a finite state machine using a leg-independent phase variable to schedule contact and

swing phases for each leg [2]. The gait scheduler utilizes independent boolean variables to define contact states scheduled  $s_\phi \in \{1 = contact, 0 = swing\}$  and switch each leg between swing and stance phases. Based on the contact schedule, the controller will execute either position control during swing or force control during stance for each leg. In this paper, in order to introduce significant model uncertainty to the system dynamics, we focus on the application of load-carrying task, where the load is unknown to the robot or the control system. Having more legs on the ground during walking could also mean that the robot could produce a larger total ground reaction force to support the heavy load. Therefore, for this task, we use quasi-static walking gait to maximize the number of legs on the grounds during walking (i.e., 3 stance legs and 1 swing leg throughout the gait). Note that while we decide to use this gait to better show the effectiveness of our approach in addressing model uncertainty, our framework is not limited by any specific gait. Similar to the baseline force-based control approach, the approach can work for different gaits by only changing the gait definition in the gait scheduler.

### C. Simplified Dynamics for Control Design

Due to the inherent nonlinear nature of the legged system, we utilized a simplified rigid-body dynamics to optimize the ground reaction forces to balance the whole body motion and enable real-time optimization. By design, the robot has light limbs with low inertia as compared to the overall body. Therefore, it is reasonable to ignore the effects of the legs on the whole body motion to plan ground reaction forces. In particular, the A1 robot controller model employs a commonly used linear relationship [9], [22] between the body's linear acceleration ( $\ddot{\mathbf{p}}_c \in \mathbb{R}^3$ ), angular acceleration ( $\dot{\boldsymbol{\omega}}_b \in \mathbb{R}^3$ ), and the ground reaction forces ( $\mathbf{F} = [\mathbf{F}_1^T, \mathbf{F}_2^T, \mathbf{F}_3^T, \mathbf{F}_4^T]^T \in \mathbb{R}^{12}$ ) acting on each of the robot's four feet. The following linear model is expressed:

$$\underbrace{\begin{bmatrix} \mathbf{I}_3 & \dots & \mathbf{I}_3 \\ [\mathbf{p}_1 - \mathbf{p}_c] \times & \dots & [\mathbf{p}_4 - \mathbf{p}_c] \times \end{bmatrix}}_{\mathbf{A} \in \mathbb{R}^{6 \times 12}} \mathbf{F} = \underbrace{\begin{bmatrix} m(\ddot{\mathbf{p}}_c + \mathbf{g}) \\ \mathbf{I}_G \dot{\boldsymbol{\omega}}_b \end{bmatrix}}_{\mathbf{b} \in \mathbb{R}^6}, \quad (1)$$

where  $m$  and  $\mathbf{I}_G \in \mathbb{R}^{3 \times 3}$  are the robot's mass and moment of inertia,  $\mathbf{g} \in \mathbb{R}^3$  is the gravity vector,  $\mathbf{p}_c \in \mathbb{R}^3$  is the position of the center of mass (COM), and  $\mathbf{p}_i \in \mathbb{R}^3$  ( $i \in \{1, 2, 3, 4\}$ ) are the positions of the feet. The term  $[\mathbf{p}_i - \mathbf{p}_c] \times$  is the skew-symmetric matrix representing the cross product  $(\mathbf{p}_i - \mathbf{p}_c) \times \mathbf{F}_i$ . The term  $\mathbf{I}_G \dot{\boldsymbol{\omega}}_b$  is actually an approximation of following equation:

$$\frac{d}{dt}(\mathbf{I}_G \boldsymbol{\omega}_b) = \mathbf{I}_G \dot{\boldsymbol{\omega}}_b + \boldsymbol{\omega}_b \times (\mathbf{I}_G \boldsymbol{\omega}_b) \approx \mathbf{I}_G \dot{\boldsymbol{\omega}}_b. \quad (2)$$

Under the assumption of small angular velocities, the term  $\boldsymbol{\omega}_b \times (\mathbf{I}_G \boldsymbol{\omega}_b)$  is relatively small and therefore will be ignored in this framework (see [9]). The vector  $\mathbf{b}$  in (1) can be

rewritten as:

$$\mathbf{b} = \underbrace{\begin{bmatrix} m\mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_G \end{bmatrix}}_{\mathbf{M} \in \mathbb{R}^{6 \times 6}} \begin{bmatrix} \ddot{\mathbf{p}}_c \\ \dot{\boldsymbol{\omega}}_b \end{bmatrix} + \underbrace{\begin{bmatrix} mg \\ \mathbf{0} \end{bmatrix}}_{\mathbf{G} \in \mathbb{R}^6}. \quad (3)$$

### D. Balance Controller

Since the model (1) is linear, the controller can naturally be formulated as the following quadratic programming (QP) problem [10], which can be solved in real-time at 1 kHz:

$$\begin{aligned} \mathbf{F}^* = \underset{\mathbf{F} \in \mathbb{R}^{12}}{\operatorname{argmin}} \quad & (\mathbf{AF} - \mathbf{b}_d)^T \mathbf{S} (\mathbf{AF} - \mathbf{b}_d) \\ & + \gamma_1 \|\mathbf{F}\|^2 + \gamma_2 \|\mathbf{F} - \mathbf{F}_{\text{prev}}^*\|^2 \\ \text{s.t.} \quad & \mathbf{d} \leq \mathbf{CF} \leq \bar{\mathbf{d}} \\ & \mathbf{F}_{\text{swing}}^z = 0 \end{aligned} \quad (4)$$

where  $\mathbf{b}_d$  is the desired dynamics and will be described in Sec. III. The cost function in (4) includes terms that consider three goals, including (1) driving the COM position and orientation to the desired trajectories; (2) minimizing the force commands; and (3) filtering the change of the current solution  $\mathbf{F}^*$  with respect to the solution from the previous time-step,  $\mathbf{F}_{\text{prev}}^*$ . The priority of each goal in the cost function is defined by the weight parameters  $\mathbf{S} \in \mathbb{R}^{6 \times 6}$ ,  $\gamma_1, \gamma_2$  respectively. The constraints in the QP formulation enforce friction constraints, input saturation, and contact constraints. More details about this QP controller can be seen in [9].

Besides the friction constraint, we will enforce the force constraints for the swing legs,  $\mathbf{F}_{\text{swing}} = \mathbf{0}$ . The swing legs are then kept at the posing position using PD control described [2] until it switches to stance phase. Based on this controller, a walking controller with a static walking gait is implemented on the robot.

## III. PROPOSED APPROACH: ADAPTIVE FORCE-BASED CONTROL

Based on the control architecture in Sec. II, in this Section, we will present a novel control architecture to integrate adaptive control into the force control framework. While our approach is not limited to any specific adaptive control approach, we decide to use  $L_1$  adaptive control [12], [18] thanks to its advancement in guaranteeing fast adaptation and smooth control signals.

Our prior work on [18] introduced adaptive control for bipedal robots based on Hybrid Zero Dynamics (HZD) [23], which uses joint position control to track the desired trajectory designed by trajectory optimization. HZD is commonly used for control of bipedal robots to address hybrid and underactuated dynamics of the system. The approach was successfully validated for walking [17] and running [21] on dynamic bipedal robots. In this paper, our approach is, however, based on force control, which optimizes ground reaction forces (GRFs) to achieve dynamic locomotion for legged robots [2]. The force control approach is robust to rough terrains [9] and capable of realizing a wide range of dynamic motions [8]. Therefore, in addition to adapting

to model uncertainty, utilizing force control could maintain these advantages for the system. In this section, we present a proposed control architecture to incorporate adaptive control into the force-based control framework.

### A. Closed-loop Dynamics

To incorporate adaptive control into the force-based control framework, we firstly reformulate the system as follow.

Let the state variable define as  $\eta = [e, \dot{e}]^T \in \mathbb{R}^{12}$ , with

$$e = \begin{bmatrix} p_c - p_{c,d} \\ \log(R_d R^T) \end{bmatrix} \in \mathbb{R}^6, \quad \dot{e} = \begin{bmatrix} \dot{p}_c - \dot{p}_{c,d} \\ \omega_b - \omega_{b,d} \end{bmatrix} \in \mathbb{R}^6, \quad (5)$$

where  $p_{c,d} \in \mathbb{R}^3$  is the desired position of COM,  $\dot{p}_{c,d} \in \mathbb{R}^3$  is the desired body's linear velocity, and  $\omega_{b,d} \in \mathbb{R}^3$  is the desired body's angular velocity. The desired and actual body orientations are described using rotation matrices  $R_d \in \mathbb{R}^{3 \times 3}$  and  $R \in \mathbb{R}^{3 \times 3}$ , respectively. The orientation error is obtained using the exponential map representation of rotations [3], [15], where the  $\log(\cdot) : \mathbb{R}^{3 \times 3} \rightarrow \mathbb{R}^3$  is a mapping from a rotation matrix to the associated rotation vector [9]. Therefore, the closed-loop error dynamics in state-space form can be represented as follow:

$$\dot{\eta} = D\eta + Hu, \quad (6)$$

where

$$D = \begin{bmatrix} 0_6 & 1_6 \\ 0_6 & 0_6 \end{bmatrix} \in \mathbb{R}^{12 \times 12}, \quad H = \begin{bmatrix} 0_6 \\ 1_6 \end{bmatrix} \in \mathbb{R}^{12 \times 6}, \quad (7)$$

and  $u \in \mathbb{R}^6$  is the control input function. By employing a PD control law, we have

$$u = [-K_P \quad -K_D] \eta, \quad (8)$$

where  $K_P \in \mathbb{R}^{6 \times 6}$  and  $K_D \in \mathbb{R}^{6 \times 6}$  are diagonal positive definite matrices.

The goal of the controller is to find out optimal leg forces  $F$  that achieve the control input function described above and accordingly maintain the error (state variable  $\eta$ ) within a bounded range. Thus, we need to find a relation between the linear model (1) and the closed-loop error dynamics (6).

First, from equation (6) it can be obtained that

$$\ddot{e} = \begin{bmatrix} \ddot{p}_c - \ddot{p}_{c,d} \\ \ddot{\omega}_b - \ddot{\omega}_{b,d} \end{bmatrix} = u, \quad (9)$$

where  $\ddot{p}_{c,d}$  and  $\ddot{\omega}_{b,d}$  are the desired COM linear acceleration and the desired angular acceleration, respectively. The desired trajectory for the robot is obtained from the velocity command. Therefore, both  $\ddot{p}_{c,d}$  and  $\ddot{\omega}_{b,d}$  are zero vectors. Then from (3) and (9), the desired dynamics can be given by

$$b_d = Mu + G, \quad (10)$$

where  $M$  and  $G$  are defined in (3).

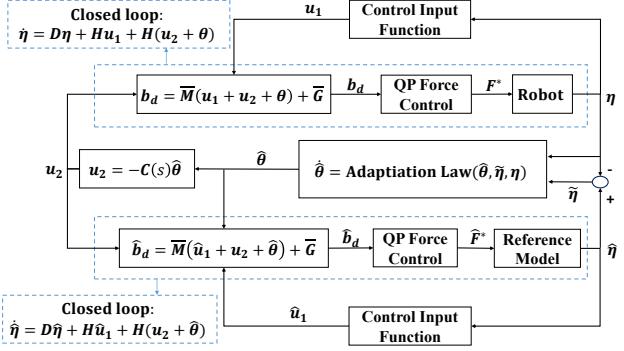


Fig. 4: Block diagram of the proposed adaptive force-based controller.

### B. Effects of uncertainty in dynamic

The QP formulation described in (4) provides input-to-state stability for quadruped during walking and standing that requires the accurate dynamical model of the system. The uncertainty comes from the mass, inertia, or rough terrain that has adverse effects on the dynamics of the system. Sometimes, it may cause instability in the control of the robot.

If we consider uncertainty in the dynamics and assume that the matrices  $M$  and  $G$  of the real dynamics are unknown, we then have to design our controller based on nominal matrices  $\bar{M}$ ,  $\bar{G}$ . Therefore, the desired dynamic can be represented as

$$b_d = \bar{M}(u + \theta) + \bar{G} \quad (11)$$

where,

$$\theta = \bar{M}^{-1} [(M - \bar{M})u + (G - \bar{G})] \in \mathbb{R}^6, \quad (12)$$

and the closed-loop system now takes the form

$$\dot{\eta} = D\eta + H(u + \theta). \quad (13)$$

### C. $L_1$ adaptive controller for compensating the uncertainties

From the previous subsection, we describe the system with uncertainty by (13). As a result, for any time  $t$ , we can always find out  $\alpha(t) \in \mathbb{R}^6$  and  $\beta(t) \in \mathbb{R}^6$  such that [6]:

$$\theta(\eta, t) = \alpha(t)\|\eta\| + \beta(t). \quad (14)$$

The goal of our method is to design a combined controller  $u = u_1 + u_2$ , where  $u_1$  is to control the model to follow the desired reference model and  $u_2$  is to compensate the nonlinear uncertainties  $\theta$ . The reference model is similar to linear model described in (1) which, instead of  $M$  and  $G$ , the nominal matrices ( $\bar{M}$ ,  $\bar{G}$ ) are being used. Moreover, the model updates itself in real-time using ODE solvers. The diagram of our proposed force-based adaptive control is presented in Fig. 4.

We present a method to consider a reference model for  $L_1$  adaptive control that arises from a QP controller with input-to-state stability describe in (4). The state predictor can then

be expressed as follows,

$$\dot{\hat{\eta}} = \mathbf{D}\hat{\eta} + \mathbf{H}\hat{u}_1 + \mathbf{H}(u_2 + \hat{\theta}), \quad (15)$$

where,

$$\hat{\theta} = \hat{\alpha}||\eta|| + \hat{\beta}, \quad (16)$$

and  $\hat{u}_1$  is defined as:

$$\hat{u}_1 = [-\mathbf{K}_P \ -\mathbf{K}_D] \hat{\eta}. \quad (17)$$

According to (13), the  $\hat{b}_d$  get the form

$$\hat{b}_d = \bar{\mathbf{M}}(\hat{u}_1 + u_2 + \hat{\theta}) + \bar{\mathbf{G}}, \quad (18)$$

and the optimal distribution of leg forces  $\hat{\mathbf{F}}$  for the reference model can be achieved by

$$\begin{aligned} \hat{\mathbf{F}}^* &= \underset{\hat{\mathbf{F}} \in \mathbb{R}^{12}}{\operatorname{argmin}} (\hat{\mathbf{A}}\hat{\mathbf{F}} - \hat{b}_d)^T \mathbf{S}(\hat{\mathbf{A}}\hat{\mathbf{F}} - \hat{b}_d) \\ &\quad + \gamma_1 \|\hat{\mathbf{F}}\|^2 + \gamma_2 \|\hat{\mathbf{F}} - \hat{\mathbf{F}}_{\text{prev}}^*\|^2 \\ \text{s.t. } & \mathbf{C}\hat{\mathbf{F}} \leq \mathbf{d} \\ & \hat{\mathbf{F}}_{\text{swing}}^z = 0. \end{aligned} \quad (19)$$

In order to compensate the estimated uncertainty  $\hat{\theta}$ , we can just simply choose  $u_2 = -\hat{\theta}$  to obtain

$$\dot{\hat{\eta}} = \mathbf{D}\hat{\eta} + \mathbf{H}\hat{u}_1. \quad (20)$$

However,  $\hat{\theta}$  typically has high frequency due to fast estimation in the adaptation law. For the reliability and robustness of the control scheme, it is essential to obtain smooth control signals, especially for robotic applications. Thus, we apply the  $L_1$  adaptive control scheme to decouple estimation and adaptation [4]. Therefore, we will have

$$u_2 = -C(s)\hat{\theta}, \quad (21)$$

where  $C(s)$  is a second-order low-pass filter with a magnitude of 1:

$$C(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}. \quad (22)$$

Define the difference between the real model and the reference model  $\tilde{\eta} = \hat{\eta} - \eta$ , we then have,

$$\dot{\tilde{\eta}} = \mathbf{D}\tilde{\eta} + \mathbf{H}\tilde{u}_1 + \mathbf{H}(\tilde{\alpha}||\eta|| + \tilde{\beta}), \quad (23)$$

where

$$\tilde{u}_1 = \hat{u}_1 - u_1, \quad \tilde{\alpha} = \hat{\alpha} - \alpha, \quad \tilde{\beta} = \hat{\beta} - \beta. \quad (24)$$

As a result, we will estimate  $\theta$  indirectly through  $\alpha$  and  $\beta$ , or the values of  $\hat{\alpha}$  and  $\hat{\beta}$  computed by the following adaptation laws based on the projection operators [14],

$$\dot{\hat{\alpha}} = \Gamma \text{Proj}(\hat{\alpha}, \mathbf{y}_\alpha), \quad \dot{\hat{\beta}} = \Gamma \text{Proj}(\hat{\beta}, \mathbf{y}_\beta). \quad (25)$$

where  $\Gamma \in \mathbb{R}^{6 \times 6}$  is a symmetric positive definite matrix. The projection functions  $\mathbf{y}_\alpha \in \mathbb{R}^6$  and  $\mathbf{y}_\beta \in \mathbb{R}^6$  are

$$\begin{aligned} \mathbf{y}_\alpha &= -\mathbf{H}^T \mathbf{P} \tilde{\eta} ||\eta||, \\ \mathbf{y}_\beta &= -\mathbf{H}^T \mathbf{P} \tilde{\eta}, \end{aligned} \quad (26)$$

where  $\mathbf{P} \in \mathbb{R}^{12 \times 12}$  will be defined in Sec. IV.

#### IV. STABILITY OF PROPOSED SYSTEM

The goal of the QP formulation described in (4) is to find a solution that drives the real dynamics  $\mathbf{A}\mathbf{F}$  to the desired dynamics  $\mathbf{b}_d$ . Nevertheless, if the desired dynamic vector  $\mathbf{b}_d$  violates the inequality constraints (such as force limits and friction constraints), the controller provides the optimal solution  $\mathbf{F}^*$  that may not satisfy the desired dynamics. With this solution, we define:

$$\mathbf{b}_d^* = \mathbf{A}\mathbf{F}^*, \quad (27)$$

$$\mathbf{u}^* = \mathbf{M}^{-1}(\mathbf{b}_d^* - \mathbf{G}). \quad (28)$$

Based on the friction constraints present in [9], the value of  $\mathbf{F}^*$  is always bounded. Besides, according to the definition of  $\mathbf{A}$ ,  $\mathbf{M}$ , and  $\mathbf{G}$ , these matrices also have bounded values. Thus, it implies that

$$\|\mathbf{u}^*\| \leq \delta_{u^*}. \quad (29)$$

##### A. Linear quadratic Lyapunov theory

According to Lyapunov, the theory [1], the PD control described in (8) will asymptotically stabilize the system if

$$\mathbf{A}_m = \begin{bmatrix} \mathbf{0}_6 & \mathbf{I}_6 \\ -\mathbf{K}_P & -\mathbf{K}_D \end{bmatrix} \in \mathbb{R}^{12 \times 12} \quad (30)$$

is Hurwitz. This means that by choosing a control Lyapunov function candidate as follows

$$V(\eta) = \eta^T \mathbf{P} \eta, \quad (31)$$

where  $\mathbf{P} \in \mathbb{R}^{12 \times 12}$  is the solution of the Lyapunov equation

$$\mathbf{A}_m^T \mathbf{P} + \mathbf{P} \mathbf{A}_m = -\mathbf{Q}, \quad (32)$$

and  $\mathbf{Q} \in \mathbb{R}^{12 \times 12}$  is any symmetric positive-definite matrix. We then have

$$\begin{aligned} \dot{V}(\eta, \mathbf{u}) + \lambda V(\eta) &= \eta^T (\mathbf{D}^T \mathbf{P} + \mathbf{P} \mathbf{D}) \eta \\ &\quad + \lambda V(\eta) + 2\eta^T \mathbf{P} \mathbf{H} \mathbf{u} \leq 0, \end{aligned} \quad (33)$$

where,

$$\lambda = \frac{\lambda_{\min}(\mathbf{Q})}{\lambda_{\max}(\mathbf{P})} > 0. \quad (34)$$

As a result, the state variable  $\eta$  and the control input  $\mathbf{u}$  always remain bounded.

$$\|\eta\| \leq \delta_\eta, \quad \|\mathbf{u}\| \leq \delta_u. \quad (35)$$

However, the control signal  $\mathbf{u}^*$  we construct by solving QP problem (4), is not always the same as  $\mathbf{u}$ . Therefore, it can be rewritten as

$$\Delta = \mathbf{u}^* - \mathbf{u} \quad (36)$$

where  $\Delta \in \mathbb{R}^6$  is the difference caused by QP optimization between the desired PD control signal  $\mathbf{u}$  and the real signal  $\mathbf{u}^*$ . This vector is also bounded according to (29) and (35)

$$\|\Delta\| \leq \delta_\Delta. \quad (37)$$

By substituting  $\mathbf{u}^*$  in (33), we have,

$$\dot{V}(\boldsymbol{\eta}, \mathbf{u}^*) + \lambda V(\boldsymbol{\eta}) \leq 2\mathbf{P}^T \mathbf{H} \Delta \leq \varepsilon_V, \quad (38)$$

where

$$\varepsilon_V = 2\|\mathbf{P}\| \delta_\eta \delta_\Delta. \quad (39)$$

### B. Stability Proof

We consider the following control Lyapunov candidate function

$$\tilde{V} = \tilde{\boldsymbol{\eta}}^T \mathbf{P} \tilde{\boldsymbol{\eta}} + \tilde{\boldsymbol{\alpha}}^T \Gamma^{-1} \tilde{\boldsymbol{\alpha}} + \tilde{\boldsymbol{\beta}}^T \Gamma^{-1} \tilde{\boldsymbol{\beta}}, \quad (40)$$

therefore, its time derivative will be

$$\begin{aligned} \dot{\tilde{V}} &= \dot{\tilde{\boldsymbol{\eta}}}^T \mathbf{P} \tilde{\boldsymbol{\eta}} + \tilde{\boldsymbol{\eta}}^T \mathbf{P} \dot{\tilde{\boldsymbol{\eta}}} + \dot{\tilde{\boldsymbol{\alpha}}}^T \Gamma^{-1} \tilde{\boldsymbol{\alpha}} + \tilde{\boldsymbol{\alpha}}^T \Gamma^{-1} \dot{\tilde{\boldsymbol{\alpha}}} \\ &\quad + \dot{\tilde{\boldsymbol{\beta}}}^T \Gamma^{-1} \tilde{\boldsymbol{\beta}} + \tilde{\boldsymbol{\beta}}^T \Gamma^{-1} \dot{\tilde{\boldsymbol{\beta}}}, \end{aligned} \quad (41)$$

in which we have

$$\begin{aligned} \dot{\tilde{\boldsymbol{\eta}}}^T \mathbf{P} \tilde{\boldsymbol{\eta}} + \tilde{\boldsymbol{\eta}}^T \mathbf{P} \dot{\tilde{\boldsymbol{\eta}}} &= (\mathbf{D}\tilde{\boldsymbol{\eta}} + \mathbf{H}\tilde{\mathbf{u}}_1^*)^T \mathbf{P} \tilde{\boldsymbol{\eta}} + \tilde{\boldsymbol{\eta}}^T \mathbf{P} (\mathbf{D}\tilde{\boldsymbol{\eta}} + \mathbf{H}\tilde{\mathbf{u}}_1^*) \\ &\quad + \tilde{\boldsymbol{\alpha}}^T \mathbf{H}^T \|\boldsymbol{\eta}\| \mathbf{P} \tilde{\boldsymbol{\eta}} + \tilde{\boldsymbol{\eta}}^T \mathbf{P} \mathbf{H} \tilde{\boldsymbol{\alpha}} \|\boldsymbol{\eta}\| \\ &\quad + \tilde{\boldsymbol{\beta}}^T \mathbf{H}^T \mathbf{P} \tilde{\boldsymbol{\eta}} + \tilde{\boldsymbol{\eta}}^T \mathbf{P} \mathbf{H} \tilde{\boldsymbol{\beta}}. \end{aligned} \quad (42)$$

Because  $\tilde{\boldsymbol{\eta}} = \hat{\boldsymbol{\eta}} - \boldsymbol{\eta}$  satisfies the condition imposed by (38), it implies that

$$(\mathbf{D}\tilde{\boldsymbol{\eta}} + \mathbf{H}\tilde{\mathbf{u}}_1^*)^T \mathbf{P} \tilde{\boldsymbol{\eta}} + \tilde{\boldsymbol{\eta}}^T \mathbf{P} (\mathbf{D}\tilde{\boldsymbol{\eta}} + \mathbf{H}\tilde{\mathbf{u}}_1^*) \leq -\lambda \tilde{\boldsymbol{\eta}}^T \mathbf{P} \tilde{\boldsymbol{\eta}} + \varepsilon_{\tilde{V}}, \quad (43)$$

where

$$\varepsilon_{\tilde{V}} = 2\|\mathbf{P}\| \delta_{\tilde{\boldsymbol{\eta}}} \delta_{\tilde{\Delta}}. \quad (44)$$

Furthermore, with the property of the projection operator [14], we have:

$$\begin{aligned} (\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha})^T (\text{Proj}(\hat{\boldsymbol{\alpha}}, \mathbf{y}_\alpha) - \mathbf{y}_\alpha) &\leq 0, \\ (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T (\text{Proj}(\hat{\boldsymbol{\beta}}, \mathbf{y}_\beta) - \mathbf{y}_\beta) &\leq 0. \end{aligned} \quad (45)$$

From (25) and (45), we can imply that

$$\begin{aligned} \tilde{\boldsymbol{\alpha}}^T \Gamma^{-1} \dot{\tilde{\boldsymbol{\alpha}}} &\leq \tilde{\boldsymbol{\alpha}}^T \mathbf{y}_\alpha - \tilde{\boldsymbol{\alpha}}^T \Gamma^{-1} \boldsymbol{\alpha}, \\ \tilde{\boldsymbol{\beta}}^T \Gamma^{-1} \dot{\tilde{\boldsymbol{\beta}}} &\leq \tilde{\boldsymbol{\beta}}^T \mathbf{y}_\beta - \tilde{\boldsymbol{\beta}}^T \Gamma^{-1} \boldsymbol{\beta}. \end{aligned} \quad (46)$$

We now replace (42), (43) and (46) to (41), which results in

$$\begin{aligned} \dot{\tilde{V}} &\leq -\lambda \tilde{\boldsymbol{\eta}}^T \mathbf{P} \tilde{\boldsymbol{\eta}} + \varepsilon_{\tilde{V}} \\ &\quad + \tilde{\boldsymbol{\alpha}}^T (\mathbf{y}_\alpha + \mathbf{H}^T \mathbf{P} \tilde{\boldsymbol{\eta}} \|\boldsymbol{\eta}\|) - \tilde{\boldsymbol{\alpha}}^T \Gamma^{-1} \dot{\boldsymbol{\alpha}} \\ &\quad + (\mathbf{y}_\alpha^T + \tilde{\boldsymbol{\eta}}^T \mathbf{P} \mathbf{H} \|\boldsymbol{\eta}\|) \tilde{\boldsymbol{\alpha}} - \dot{\tilde{\boldsymbol{\alpha}}}^T \Gamma^{-1} \boldsymbol{\alpha} \\ &\quad + \tilde{\boldsymbol{\beta}}^T (\mathbf{y}_\beta + \mathbf{H}^T \mathbf{P} \tilde{\boldsymbol{\eta}}) - \tilde{\boldsymbol{\beta}}^T \Gamma^{-1} \dot{\boldsymbol{\beta}} \\ &\quad + (\mathbf{y}_\beta^T + \tilde{\boldsymbol{\eta}}^T \mathbf{P} \mathbf{H}) \tilde{\boldsymbol{\beta}} - \dot{\tilde{\boldsymbol{\beta}}}^T \Gamma^{-1} \boldsymbol{\beta} \end{aligned} \quad (47)$$

So, by using the chosen projection functions (26), then we conclude that.

$$\begin{aligned} \dot{\tilde{V}} + \lambda \tilde{V} &\leq \varepsilon_{\tilde{V}} + \lambda \tilde{\boldsymbol{\alpha}}^T \Gamma^{-1} \tilde{\boldsymbol{\alpha}} + \lambda \tilde{\boldsymbol{\beta}}^T \Gamma^{-1} \tilde{\boldsymbol{\beta}} \\ &\quad - \tilde{\boldsymbol{\alpha}}^T \Gamma^{-1} \dot{\boldsymbol{\alpha}} - \dot{\tilde{\boldsymbol{\alpha}}}^T \Gamma^{-1} \boldsymbol{\alpha} \\ &\quad - \tilde{\boldsymbol{\beta}}^T \Gamma^{-1} \dot{\boldsymbol{\beta}} - \dot{\tilde{\boldsymbol{\beta}}}^T \Gamma^{-1} \boldsymbol{\beta}. \end{aligned} \quad (48)$$

We assume that the uncertainties  $\boldsymbol{\alpha}$ ,  $\boldsymbol{\beta}$  and their time derivatives are bounded. Furthermore, the projection operators (25) will also keep  $\tilde{\boldsymbol{\alpha}}$  and  $\tilde{\boldsymbol{\beta}}$  bounded (see [6] for a detailed proof about these properties.) We define these bounds as follows:

$$\begin{aligned} \|\tilde{\boldsymbol{\alpha}}\| &\leq \tilde{\alpha}_b, \quad \|\tilde{\boldsymbol{\beta}}\| \leq \tilde{\beta}_b, \\ \|\dot{\tilde{\boldsymbol{\alpha}}}\| &\leq \dot{\alpha}_b, \quad \|\dot{\tilde{\boldsymbol{\beta}}}\| \leq \dot{\beta}_b. \end{aligned} \quad (49)$$

Combining this with (48), we have,

$$\dot{\tilde{V}} + \lambda \tilde{V} \leq \lambda \delta_{\tilde{V}}, \quad (50)$$

where

$$\delta_{\tilde{V}} = 2\|\Gamma\|^{-1} (\tilde{\alpha}_b^2 + \tilde{\beta}_b^2 + \frac{1}{\lambda} \tilde{\alpha}_b \dot{\alpha}_b + \frac{1}{\lambda} \tilde{\beta}_b \dot{\beta}_b) + \frac{1}{\lambda} \varepsilon_{\tilde{V}}. \quad (51)$$

Thus, if  $\tilde{V} \geq \delta_{\tilde{V}}$  then  $\dot{\tilde{V}} \leq 0$ . As a result, we always have  $\tilde{V} \leq \delta_{\tilde{V}}$ . In other words, by choosing the adaptation gain  $\Gamma$  sufficiently large and  $\mathbf{P}$  quite small, we can limit the Control Lyapunov Function (40) in an arbitrarily small neighborhood  $\delta_{\tilde{V}}$  of the origin. According to (30) and (32), achieving a small value for  $\mathbf{P}$  depends on choosing a proper value for  $\mathbf{K}_P$ ,  $\mathbf{K}_D$ , and  $\mathbf{Q}$ . Therefore, the value of PD gains affects the stability of the whole system. Finally, the tracking errors between the dynamics model (13) and the reference model (15),  $\tilde{\boldsymbol{\eta}}$ , and the error between the real and estimated uncertainty,  $\tilde{\boldsymbol{\alpha}}$ ,  $\tilde{\boldsymbol{\beta}}$  are bounded as follows:

$$\|\tilde{\boldsymbol{\eta}}\| \leq \sqrt{\frac{\delta_{\tilde{V}}}{\|\mathbf{P}\|}}, \quad \|\tilde{\boldsymbol{\alpha}}\| \leq \sqrt{\|\Gamma\| \delta_{\tilde{V}}}, \quad \|\tilde{\boldsymbol{\beta}}\| \leq \sqrt{\|\Gamma\| \delta_{\tilde{V}}}. \quad (52)$$

### V. NUMERICAL VALIDATION

This section presents validation conducted in a high-fidelity simulation of the A1 robot. The control system is implemented in ROS Melodic with Gazebo 9 simulator. For the adaptive controller, we set the parameters as presented in table I. We set the adaptive gains larger for height, pitch, and roll because our simulation emphasizes the adaptation for these 3 states. The robot is simulated to carry a load of up to 50% of the robot weight during walking based on the approach we have developed in previous sections. Fig. 5 shows that our proposed adaptive force control can enable the robot to climb stably to an uneven slope while adapting to time-varying model uncertainty (carrying an unknown load while being pushed by a varying external force in the z-direction). This result not only proves our proposed approach's capability in adapting to a high level of model uncertainty but also illustrates that our approach can retain the robustness of force control in navigating rough terrains.

In the simulation, to introduce model uncertainty in both mass and inertia to the robot dynamics, a 6-kg load, which is 50% of the robot's weight, is applied to the robot's back. As shown in Fig. 6, while the non-adaptive controller fails to keep the robot balanced with about 30° error in the pitch angle and 9 cm error in the robot height, our proposed adaptive control can keep the robot balanced with a very small tracking error in both pitch angle (less than 8°) and

TABLE I: Controller Setting

Parameter	Value	Parameter	Value
$\mathbf{K}_P$	$diag(30, 30, 50, 80, 80, 80)$	$\gamma_1$	0.01
$\mathbf{K}_D$	$diag(10, 10, 10, 50, 50, 50)$	$\gamma_2$	0.001
$\mathbf{S}$	$diag(5, 5, 10, 50, 25, 20)$	$\zeta$	0.7
$\Gamma$	$diag(1, 1, 5, 2, 5, 1) \times 10^3$	$\omega_n$	400

robot height (less than 3 cm). Since our simulation does not emphasize the significant difference in tracking errors of position along the x-axis, y-axis, roll, and yaw angles, we select to show plots of tracking errors in the z-axis and the pitch angle.

## VI. EXPERIMENTAL VALIDATION

We have also successfully implemented our approach on the real robot hardware (see Fig. 1). This section will show our comparison between (a) non-adaptive control and (b) adaptive control for the problem of quadruped robots standing and walking while carrying an unknown load on the trunk. In the experiment, the weight and inertia of the load are unknown to neither the non-adaptive control nor the adaptive control approach. To demonstrate the effectiveness of our approach, we tested the baseline controller and adaptive controller with the same load and control parameters. We

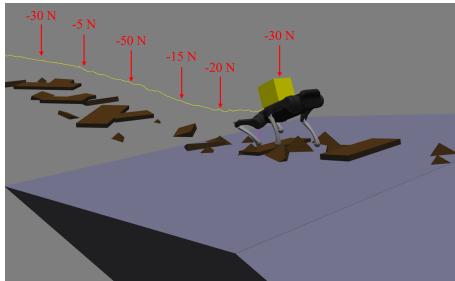


Fig. 5: **The A1 Robot Simulation.** The robot walking on high-sloped terrain with obstacles while carrying a 6kg load with varying external force. Simulation video: <https://youtu.be/UHz8h-CuZ6w>

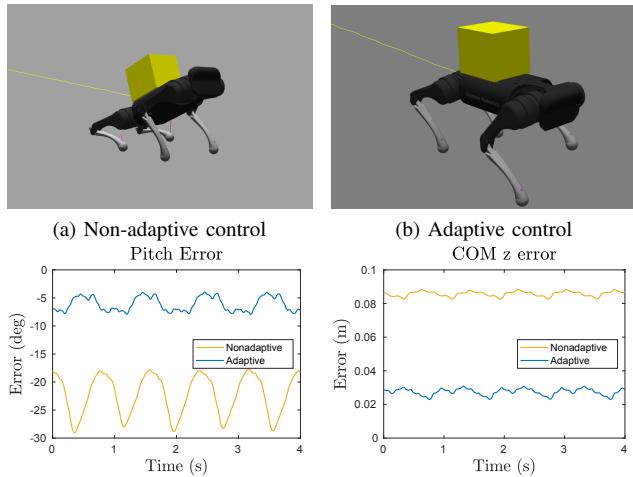
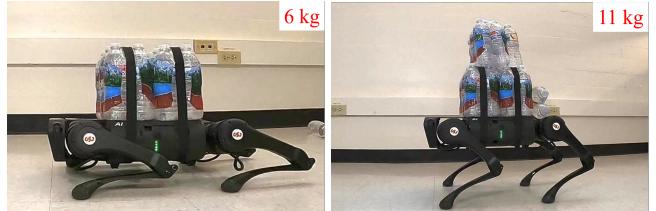


Fig. 6: Simulation results of the robot walking with a 6kg load on the back using (a) non-adaptive control and (b) adaptive control.



(a) Non-adaptive control

(b) Adaptive control

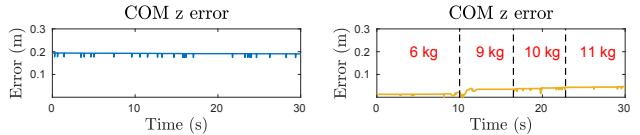
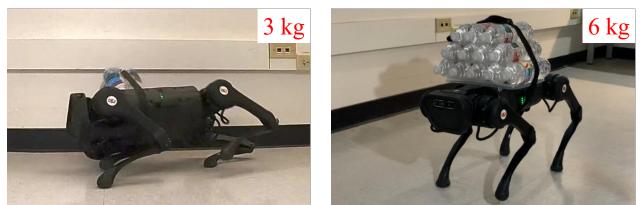


Fig. 7: **Standing experiment results.** a) Non-adaptive controller with 6-kg load; b) Adaptive controller with up to 11-kg load.



(a) Non-adaptive control

(b) Adaptive control

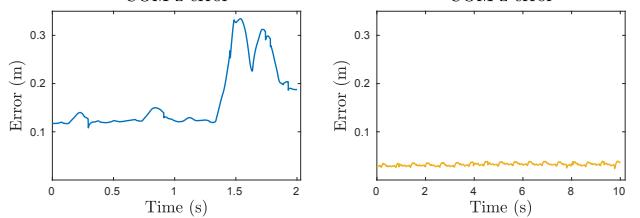


Fig. 8: **Walking experiment results.** a) Non-adaptive controller walking with 3-kg load; b) Adaptive controller walking with a 6-kg load.

added the load to the robot throughout the experiment until it fails or meets the expectation from our simulation. For standing, we added loads gradually from 1 kg for the robot to stand up to a height of 30cm. As presented in Fig. 7, with the baseline non-adaptive controller, the robot could barely stand up when the load is added to 6kg, resulting in a large tracking error of in z-direction of approximately 20 cm. With the adaptive controller, the robot can stand up with 6kg load on its back with a small tracking error (about 2 cm), plus it can handle a time-varying load of up to 11 kg. With the adaptive control, the tracking error in the z-axis is still less than 5 cm even with an unknown 11-kg load applied to the robot.

In the walking experiment, as presented in Fig. 8, the robot fails with the non-adaptive controller when a load of only 3kg is applied to the robot. The robot cannot keep balance due to model uncertainty and failed to the right within 2 seconds of walking. In comparison, our proposed adaptive force control can allow the robot to walk stably in all directions while carrying an unknown load of 6 kg,

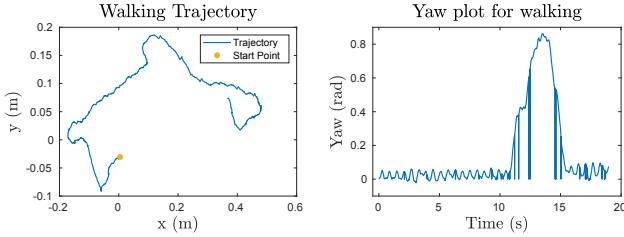


Fig. 9: **Walking experiment plot** a) robot trajectory plot; b) yaw plot for walking.

which is 50% of the robot's weight. Fig. 9, shows the path of the robot walking forward, backward, side-way, and turning while carrying the heavy load. The spikes shown in Fig. 7 and Fig. 9 result from the noise of the IMU during the experiment. More details of the walking experiment are presented in the supplemental video. Although there is a small constant error for the adaptive controller, it does not contradict the algorithm we have developed because the controller system we have designed guarantees input-to-state stability. The experiment results have clearly demonstrated the advancements of our proposed approach.

## VII. CONCLUSION

In summary, we have presented a novel control system that incorporates adaptive control into force control for dynamic legged robots walking under uncertainties. We have demonstrated the effectiveness of our proposed approach using both numerical and experimental validations. In simulation, we have shown that our proposed approach can allow the robot to climb up an uneven slope while carrying an unknown load of up to 6 kg (50% of the robot weight). In experiment, we have successfully implemented our proposed adaptive force control for quadruped robots standing and walking while carrying an unknown heavy load on the trunk. The experiment has presented impressive results when the robot can carry up to 11 kg of unknown load (92% of the robot weight) while standing with 4 legs while maintaining a tracking error of less than 5 cm in the robot height. In addition, with the adaptive controller, the robot can also walk stably in all directions with 6 kg of load on its trunk. In contrast, the baseline non-adaptive controller fails within 2 seconds of walking with only 3 kg of load. Our work has shown that the proposed adaptive force control not only can adapt to large model uncertainty but can also leverage the advantages of force control in navigating rough terrains for legged robots. In the future, we will extend our framework to achieve more dynamic gaits.

## REFERENCES

- [1] A. D. Ames, K. Galloway, J. W. Grizzle, and K. Sreenath, "Rapidly Exponentially Stabilizing Control Lyapunov Functions and Hybrid Zero Dynamics," *IEEE Trans. Automatic Control*, 2014.
- [2] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2245–2252.
- [3] F. Bullo and R. M. Murray, "Proportional derivative (PD) control on the Euclidean group," in *Proc. of the European Control Conference*, Rome, Italy, Jun. 1995, pp. 1091–1097.
- [4] C. C. and N. Hovakimyan, "Design and analysis of a novel l1 adaptive controller with guaranteed transient performance," *IEEE Transactions on Automatic Control*, vol. 53, no.2, pp. 586–591, March, 2008.
- [5] C. Cao and N. Hovakimyan, "Stability margins of l1 adaptive controller: Part ii," *Proceedings of American Control Conference*, pp. 3931–3936, 2007.
- [6] C. Cao and N. Hovakimyan, "L1 adaptive controller for a class of systems with unknown nonlinearities: Part i," *American Control Conference, Seattle, WA*, pp. 4093–4098, 2008.
- [7] Y. Chen, A. Pandey, Z. Deng, A. Nguyen, R. Wang, P. Thorrapalin, Q. Nguyen, and S. K. Gupta, "A semi-autonomous quadruped robot for performing disinfection in cluttered environments," in *2021 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC-CIE)*, Online, 2021.
- [8] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.
- [9] M. Focchi, A. del Prete, I. Havoutis, R. Featherstone, D. G. Caldwell, and C. Semini, "High-slope terrain locomotion for torque-controlled quadruped robots," *Autonomous Robots*, vol. 41, no. 1, pp. 259–272, Jan 2017. [Online]. Available: <https://doi.org/10.1007/s10514-016-9573-1>
- [10] C. Gehring, S. Coros, M. Hutter, M. Bloesch, M. Hoepflinger, and R. Siegwart, "Control of dynamic gaits for a quadrupedal robot," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 3287–3292.
- [11] J. Grizzle, C. Chevallereau, and C. Shih, "Hzd-based control of a five-link underactuated 3D bipedal robot," in *IEEE Conf. on Decision and Control*, Cancun, Mexico, December 2008.
- [12] N. Hovakimyan and C. Cao, *L1 adaptive control theory: Guaranteed robustness with fast adaptation*. Siam, 2010, vol. 21.
- [13] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger, "ANYmal - a highly mobile and dynamic quadrupedal robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2016, pp. 38–44.
- [14] E. Lavretsky, T. E. Gibson, and A. M. Annaswamy, "Projection operator in adaptive systems," *arXiv:1112.4232v6*, 2012.
- [15] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.
- [16] Q. Nguyen, M. J. Powell, B. Katz, J. D. Carlo, and S. Kim, "Optimized jumping on the mit cheetah 3 robot," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 7448–7454.
- [17] Q. Nguyen, A. Agrawal, W. Martin, H. Geyer, and K. Sreenath, "Dynamic bipedal locomotion over stochastic discrete terrain," *The International Journal of Robotics Research*, vol. 37, no. 13–14, pp. 1537–1553, 2018.
- [18] Q. Nguyen and K. Sreenath, "L1 adaptive control for bipedal robots with control lyapunov function based quadratic programs," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 862–867.
- [19] H.-W. Park, P. Wensing, and S. Kim, "Online planning for autonomous running jumps over obstacles in high-speed quadrupeds," in *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.
- [20] H.-W. Park, P. M. Wensing, and S. Kim, "High-speed bounding with the mit cheetah 2: Control design and experiments," *The International Journal of Robotics Research*, vol. 36, no. 2, pp. 167–192, 2017.
- [21] K. Sreenath, H.-W. Park, I. Poulikakos, and J. W. Grizzle, "Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on MABEL," *The International Journal of Robotics Research (IJRR)*, vol. 32, no. 3, pp. 324–345, March 2013.
- [22] B. Stephens and C. Atkeson, "Push recovery by stepping for humanoid robots with force controlled joints," in *IEEE-RAS International Conference on Humanoid Robots*, Dec. 2010, pp. 52–59.
- [23] E. Westervelt, J. Grizzle, and D. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 42–56, January 2003.