An Optimization-Based Locomotion Controller for Quadruped Robots Leveraging Cartesian Impedance Control

Guiyang Xin [1,2]*, Wouter Wolfslag [1,2], Hsiu-Chin Lin [3], Carlo Tiseo [1,2] and Michael Mistry [1,2]

[1] School of Informatics, Institute of Perception, Action and Behaviour, The University of Edinburgh, Edinburgh, United Kingdom, [2] Edinburgh Centre for Robotics, The University of Edinburgh, Edinburgh, United Kingdom, [3] School of Computer Science, Department of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada

Quadruped robots require compliance to handle unexpected external forces, such as impulsive contact forces from rough terrain, or from physical human-robot interaction. This paper presents a locomotion controller using Cartesian impedance control to coordinate tracking performance and desired compliance, along with Quadratic Programming (QP) to satisfy friction cone constraints, unilateral constraints, and torque limits. First, we resort to projected inverse-dynamics to derive an analytical control law of Cartesian impedance control for constrained and underactuated systems (typically a quadruped robot). Second, we formulate a QP to compute the optimal torques that are as close as possible to the desired values resulting from Cartesian impedance control while satisfying all of the physical constraints. When the desired motion torques lead to violation of physical constraints, the QP will result in a trade-off solution that sacrifices motion performance to ensure physical constraints. The proposed algorithm gives us more insight into the system that benefits from an analytical derivation and more efficient computation compared to hierarchical QP (HQP) controllers that typically require a solution of three QPs or more. Experiments applied on the ANYmal robot with various challenging terrains show the efficiency and performance of our controller.

Keywords: locomotion controller, projected inverse-dynamics, quadratic programming, impedance control, quadruped robots

1. INTRODUCTION

Recent improvements in mobile robotics indicate that it is possible to achieve autonomous inspection and maintenance of critical industrial infrastructure in extreme environments (e.g., off-shore plants and nuclear sites) within the next decade. The automation of such processes will reduce their economic cost and increase the life quality of the operators that can be moved away from operational hazards. Legged robots are the most promising among ground robots for locomotion flexibility, being able to surpass a varied set of obstacles. However, this comes at the expense of reduced stability compared to wheeled robots. Quadruped robots offer a balance between efficiency and stability. Algorithms that can generate precise but compliant motions while satisfying physical

feasibility are required to control quadruped robots with a large number of degrees of freedom. On the other hand, a trade-off strategy is also required to coordinate multiple tasks and constraints.

A good example for legged locomotion is the management of the contact forces that need to provide sufficient support to the body while not violating the non-slipping conditions determined by the friction cones. An intuitive control technique that can deal with these problems is the Virtual Model Controller (VMC) (Pratt et al., 2001), which has been applied to state of the art legged prototypes (Boaventura et al., 2012; Gehring et al., 2013). The VMC determines the contact forces based on the centroidal momentum model according to desired motions, and then maps these forces into joint torques. Meanwhile, the swing legs are controlled using joint PD controllers tracking a desired reference trajectory. Therefore, VMC controls swing legs and supporting legs separately and is not a whole-body controller.

Another widely used control framework is the fully optimization-based control (De Lasa et al., 2010; Saab et al., 2013). This framework relies on solving an optimization problem (usually quadratic) to minimize a cost function. The dynamic model and desired end-effector motions are introduced as equality constraints, while friction cones, torques, and joint limitations are inequality constraints. The main benefit of the optimization-based control is the integration of the operational space tasks and the joint space dynamics in a single problem. Furthermore, its formulation can be extended to solve multiple tasks, which can be implemented using either weighted optimization or strictly hierarchical optimization (Hutter et al., 2014). Two popular implementations are the Hierarchical Quadratic Programming (HQP) and the Weighted Quadratic Programming (WQP). HQP uses a strictly prioritized control framework based on null-space projectors; meanwhile, WQP relies on tunable weight matrices. Despite the fact that HQP solves two or more QPs, it is normally preferred to WQP because it does not require tuning.

The QP-schemes outlined above optimize for joint accelerations, contact forces, and joint torques, but only joint torques are eventually used for control. Since only torques are needed to control the robot, the original optimization formulation can be simplified by reducing optimization variables by considering reducing computation complexity as well. In Bellicoso et al. (2016), the QR decomposition proposed in Mistry et al. (2010) is employed to eliminate contact forces from dynamic equations so that optimization variables are reduced to two vectors. In Mansard (2012), a projector considering the consistency of the inertia matrix is proposed to separate the dynamic equations, and the optimization variables are finally reduced to two vectors. A straightforward decomposition is presented in Herzog et al. (2016) that directly splits six rows corresponding to six dimensional floating bases from the whole dynamic equation, thus removing actuation torques from optimization variables. By contrast, joint torques are the decision variables for the proposed control technique in this paper, and only one QP needs to be solved.

For optimization-based approaches, operational tasks are bounded to joint space dynamics by optimization constraints. Consequently, tuning parameters, such as impedance gains require several trials. A seminal control technique called operational space control (OSC), presented first in Khatib (1987), provides us with the analytical solution to derive joint torques from desired operational tasks. OSC has been extended to hierarchical OSC by iterative null-space projection for legged robots (Sentis and Khatib, 2006; Sentis et al., 2010; Lee et al., 2016). In Mistry and Righetti (2012), the orthogonal projector proposed by Aghili (2005) is applied to OSC with the consideration of reducing complexity. Subsequently, in Lin et al. (2018) and Xin et al. (2018), we extend that approach to involve the analytical Cartesian impedance controller proposed in Albu-Schäffer et al. (2003) and QP optimization. This paper is a further extension of our previous works under the framework of projected inverse dynamics.

The formulation proposed in this manuscript is a synergistic integration between the analytical Cartesian impedance controller and the QP. The method enables compensation of model errors while operating a trade-off strategy between multiple constraints, thus improving the locomotion capability traversing steep and slippery terrains. Specifically, the contributions of this paper come from combining the following components in a single control pipeline:

- Analytical Cartesian impedance control, based on the orthogonal projected inverse-dynamics, which allows us to analyze a legged robot using a mass-spring-damper model against disturbances.
- Perturbation estimation using the impedance behavior to evaluate disturbances including external forces and model errors.
- QP optimization that performs a trade-off strategy to relax trajectory tracking when feasibility constraints are close to violation, which is critical when walking in extreme environments.

Finally, the proposed pipeline and controller are extensively validated on the torque controllable quadruped robot ANYmal (see **Figure 1**) in various scenarios.

2. PROBLEM FORMULATION

In this section we consider the general problem of constrained inverse-dynamics for quadruped robots, including a discussion of the controllability of quadruped robots. Then we present the projected inverse-dynamics to deal with constrained dynamics, which we will use to derive our locomotion controller for quadruped robots.

2.1. Dynamic Model of Quadruped Robots

The rigid-body dynamics equation of a floating base system under contact constraints is

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{h} = \mathbf{B}\boldsymbol{\tau} + \mathbf{J}_c^\top \boldsymbol{\lambda} \qquad (1)$$
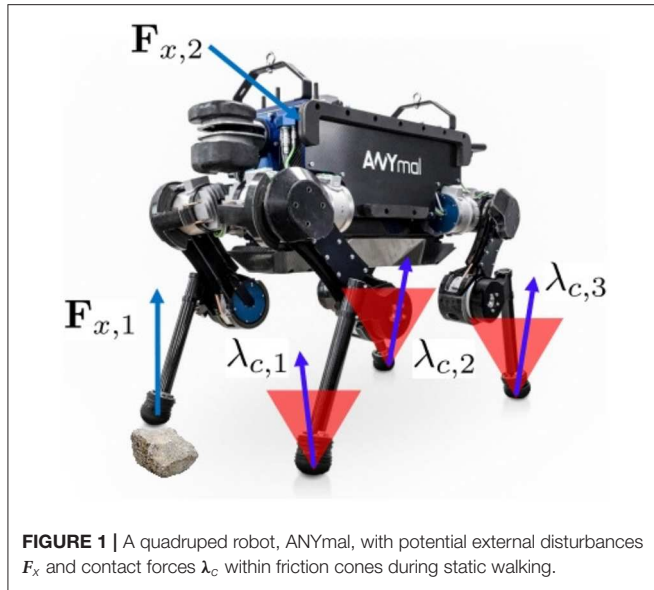
FIGURE 1 | A quadruped robot, ANYmal, with potential external disturbances F_x and contact forces λ_c within friction cones during static walking.

where $\mathbf{q} = [\mathbf{x}_b^\top \; \boldsymbol{\theta}^\top]^\top$ denotes the vector of actuated joint positions ($\boldsymbol{\theta} \in \mathbb{R}^n$) and floating base position and orientation ($\mathbf{x}_b \in SE(3)$), $\mathbf{M} \in \mathbb{R}^{(n+6)\times(n+6)}$ is the inertia matrix, $\mathbf{h} \in \mathbb{R}^{n+6}$ is the generalized vector containing Coriolis, centrifugal and gravitational effects, $\boldsymbol{\tau} \in \mathbb{R}^{n+6}$ is the vector of torques, $\mathbf{J}_c \in \mathbb{R}^{k\times(n+6)}$ is the constraint Jacobian that describes k linearly independent constraints, $\boldsymbol{\lambda} \in \mathbb{R}^k$ is the generalized constraint force vector that enforces the condition of $\mathbf{J}_c\ddot{\mathbf{q}} + \dot{\mathbf{J}}_c\dot{\mathbf{q}} = \mathbf{0}$ (or $\mathbf{J}_c\dot{\mathbf{q}} = \mathbf{0}$), and

$$\mathbf{B} = \begin{bmatrix} \mathbf{0}_6 & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_n \end{bmatrix} \qquad (2)$$

is the projector into actuated joint space with n dimensional identity matrix $\mathbf{I}_n$.

It should be noted that $\mathbf{J}_c = [\mathbf{J}_{c_b} \; \mathbf{J}_{c_s}]$, whereby $\mathbf{J}_{c_b} \in \mathbb{R}^{k\times 6}$ and $\mathbf{J}_{c_s} \in \mathbb{R}^{k\times n}$. $\mathbf{J}_{c_b}$ encodes the effect of contact constraints on the floating base, with its rank providing direct insights into the controllability properties of the robots (Mistry et al., 2008). For quadruped robots, with the assumption of point contacts, there is $k = 3 \times n_c$ where n_c denotes the number of legs in stance. The case of $rank(\mathbf{J}_{c_b}) < 6$ (such as during a trotting gait with only two stance legs, and static walking with three stance feet in a line) is called truly underactuated (Hutter et al., 2014) which means there are not enough contact constraints to independently control the six unactuated base coordinates $\mathbf{x}_b$. The case of $rank(\mathbf{J}_c) - rank(\mathbf{J}_{c_b}) > 0$ is called overconstrained which means we can use redundancy to control the constraint/internal forces. Static walking (except with co-linear feet) is a typical overconstrained case, as $rank(\mathbf{J}_c) - rank(\mathbf{J}_{c_b}) = 9 - 6 = 3$ with one swing leg. The proposed controller incorporating QP aims to control the internal forces to satisfy contact constraints. Moreover, one should note that we cannot independently generate the contact forces for both static walking and trotting gait because of $rank(\mathbf{J}_c) - rank(\mathbf{J}_{c_b}) < rank(\mathbf{J}_c)$.

2.2. Projected Inverse Dynamics

The dynamics Equation (1) can be divided into two subspaces: the motion space where the acceleration can be chosen freely, and the constraint space where the acceleration is fixed and constraint forces can be chosen. Working with the two subspaces allows us to satisfy contact constraints independently of joint motion. Paper (Aghili, 2005) proposes an orthogonal projector derived only from kinematic parameters without consideration of constraint dynamic consistency, leading to a much simplified solution compared to Sentis et al. (2010). The employment of this orthogonal projector is the key for analytical Cartesian impedance control and optimization variable reduction.

The constraint equation $\mathbf{J}_c\dot{\mathbf{q}} = \mathbf{0}$ implies that any admissible generalized velocity must lie in the null space of matrix $\mathbf{J}_c$, i.e., $\dot{\mathbf{q}} \in \mathcal{N}(\mathbf{J}_c)$. We can use the pseudoinverse of $\mathbf{J}_c$ to compute the orthogonal null space projector $\mathbf{P}$ as $\mathbf{P} = \mathbf{I} - \mathbf{J}_c^+\mathbf{J}_c$, such that $\mathcal{R}(\mathbf{P}) = \mathcal{N}(\mathbf{J}_c)$, $\mathbf{P}\dot{\mathbf{q}} = \dot{\mathbf{q}}$, $\mathbf{P}\mathbf{J}_c^\top = \mathbf{0}$, and $\mathbf{P}^2 = \mathbf{P}^\top = \mathbf{P}$. Since the vectors lying in $\mathcal{N}(\mathbf{J}_c)$ denote the motions of torso and any end-effector in the air, we call $\mathcal{N}(\mathbf{J}_c)$ the motion space. It should noted that any vector could be projected into motion space by pre-multiplying with the projection matrix $\mathbf{P}$. Then the complimentary projection $\mathbf{I} - \mathbf{P}$ projects vectors into the constraint space $\mathcal{N}(\mathbf{J}_c)^\perp = \mathcal{R}(\mathbf{J}_c^\top)$. Subsequently, we can split Equation (1) into two equations of two orthogonal spaces (motion space and constraint space) using the above two projectors:

$$\mathbf{P}\mathbf{M}\ddot{\mathbf{q}} + \mathbf{P}\mathbf{h} = \mathbf{P}\mathbf{B}\boldsymbol{\tau} = \boldsymbol{\tau}_m \qquad (3)$$

$$(\mathbf{I} - \mathbf{P})(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{h}) = (\mathbf{I} - \mathbf{P})\mathbf{B}\boldsymbol{\tau} + \mathbf{J}_c^\top\boldsymbol{\lambda} = \boldsymbol{\tau}_c + \mathbf{J}_c^\top\boldsymbol{\lambda} \qquad (4)$$

and then torque command $\boldsymbol{\tau} = \boldsymbol{\tau}_m + \boldsymbol{\tau}_c$. Constraint torques $\boldsymbol{\tau}_c$ in Equation (4) do not contribute to the motion of system, but system motion will affect contact forces due to $\ddot{\mathbf{q}}$ in Equation (4). We wish to invert $\mathbf{P}\mathbf{M}$ in Equation (3) to solve $\ddot{\mathbf{q}}$, and then substitute that result into (4) to determine contact forces. However, $\mathbf{P}\mathbf{M}$ is not invertible due to the rank deficiency of $\mathbf{P}$. The contact constraint dictates that $(\mathbf{I} - \mathbf{P})\dot{\mathbf{q}} = \mathbf{0}$, and its derivative

$$(\mathbf{I} - \mathbf{P})\ddot{\mathbf{q}} = \dot{\mathbf{P}}\dot{\mathbf{q}} \qquad (5)$$

Substituting Equation (5) into Equation (3) yields

$$\mathbf{M}_c\ddot{\mathbf{q}} + \mathbf{P}\mathbf{h} - \dot{\mathbf{P}}\dot{\mathbf{q}} = \boldsymbol{\tau}_m \qquad (6)$$

where $\mathbf{M}_c = \mathbf{P}\mathbf{M} + \mathbf{I} - \mathbf{P}$ is invertible and named as constraint inertia matrix. Then we get

$$\ddot{\mathbf{q}} = \mathbf{M}_c^{-1}(\boldsymbol{\tau}_m - \mathbf{P}\mathbf{h} + \dot{\mathbf{P}}\dot{\mathbf{q}}) \qquad (7)$$

Substituting Equation (7) into Equation (4) yields the equation that can determine the contact forces

$$\mathbf{J}_c^\top\boldsymbol{\lambda} = (\mathbf{I} - \mathbf{P})[\mathbf{M}\mathbf{M}_c^{-1}(\boldsymbol{\tau}_m - \mathbf{P}\mathbf{h} + \dot{\mathbf{P}}\dot{\mathbf{q}}) + \mathbf{h}] - \boldsymbol{\tau}_c \qquad (8)$$

By replacing $\boldsymbol{\tau}_m$ and $\boldsymbol{\tau}_c$ with their original definitions in terms of $\boldsymbol{\tau}$, Equation (8) becomes

$$\mathbf{J}_c^\top \boldsymbol{\lambda} = (\mathbf{I} - \mathbf{P})[-(\mathbf{I} - \mathbf{MM}_c^{-1}\mathbf{P})(\mathbf{B}\boldsymbol{\tau} - \mathbf{h}) + \mathbf{MM}_c^{-1}\dot{\mathbf{P}}\dot{\mathbf{q}}] \quad (9)$$

It should be noted that the utility of Equation (7) is key for the derivation of the analytical impedance control law in section 3 and the reduction of optimization parameters in section 4 to $\boldsymbol{\tau}$ instead of $[\ddot{\mathbf{q}}^\top \ \boldsymbol{\tau}^\top]^\top$.

3. CARTESIAN IMPEDANCE CONTROL IN MOTION SPACE

3.1. Applying Cartesian Impedance Control to Underactuated Systems

Since contact forces have been removed from the projected inverse dynamics of Equation (3), we can design a Cartesian impedance controller to achieve a desired motion performance in motion space without considering contact constraints. First, for the four stance legs case, only the torso requires desired motion, the torso is therefore considered an end-effector. The objective of Cartesian impedance control is to dictate the disturbance response of the robot. If a given Cartesian location $\mathbf{x}$ of the end-effector is subject to an external disturbance $\mathbf{F}_x$ we would like to prescribe the motion at $\mathbf{x}$ as

$$\boldsymbol{\Lambda}_d \ddot{\mathbf{e}} + \mathbf{D}_d \dot{\mathbf{e}} + \mathbf{K}_d \mathbf{e} = \mathbf{F}_x \quad (10)$$

where $\mathbf{e} = \mathbf{x} - \mathbf{x}_d$ is the motion error between the current pose and the desired pose, $\boldsymbol{\Lambda}_d$, $\mathbf{D}_d$ and $\mathbf{K}_d$ are the desired Cartesian inertia, damping, and stiffness matrices that define the Cartesian impedance behavior. On the other hand, adding the external disturbance into the general dynamic equation Results in

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{h} = \mathbf{B}\boldsymbol{\tau} + \mathbf{J}_c^\top \boldsymbol{\lambda} + \mathbf{J}_x^\top \mathbf{F}_x \quad (11)$$

where $\mathbf{J}_x$ is the Jacobian matrix mapping external forces/torques to joint space. Project Equation (11) into the motion space

$$\mathbf{PM}\ddot{\mathbf{q}} + \mathbf{Ph} = \mathbf{PB}\boldsymbol{\tau} + \mathbf{PJ}_x^\top \mathbf{F}_x \quad (12)$$

Next, we employ $\mathbf{M}_c$ to transform Equation (12) to operational space. Equation (12) can be rewritten as

$$\mathbf{M}_c \ddot{\mathbf{q}} + \mathbf{Ph} - \dot{\mathbf{P}}\dot{\mathbf{q}} = \mathbf{PB}\boldsymbol{\tau} + \mathbf{PJ}_x^\top \mathbf{F}_x \quad (13)$$

Pre-multiplying by $\mathbf{J}_x \mathbf{M}_c^{-1}$ gives

$$\mathbf{J}_x \ddot{\mathbf{q}} + \mathbf{J}_x \mathbf{M}_c^{-1}(\mathbf{Ph} - \dot{\mathbf{P}}\dot{\mathbf{q}}) = \mathbf{J}_x \mathbf{M}_c^{-1}\mathbf{PB}\boldsymbol{\tau} + \mathbf{J}_x \mathbf{M}_c^{-1}\mathbf{PJ}_x^\top \mathbf{F}_x \quad (14)$$

We replace $\mathbf{J}_x \ddot{\mathbf{q}}$ with $\ddot{\mathbf{x}} - \dot{\mathbf{J}}_x \dot{\mathbf{q}}$ and multiply by $\boldsymbol{\Lambda}_c = (\mathbf{J}_x \mathbf{M}_c^{-1}\mathbf{PJ}_x^\top)^{-1}$

$$\boldsymbol{\Lambda}_c \ddot{\mathbf{x}} + \boldsymbol{\Lambda}_c \mathbf{J}_x \mathbf{M}_c^{-1}(\mathbf{Ph} - \dot{\mathbf{P}}\dot{\mathbf{q}}) - \boldsymbol{\Lambda}_c \dot{\mathbf{J}}_x \dot{\mathbf{q}} = \boldsymbol{\Lambda}_c \mathbf{J}_x \mathbf{M}_c^{-1}\mathbf{PB}\boldsymbol{\tau} + \mathbf{F}_x \quad (15)$$

where $\boldsymbol{\Lambda}_c$ is named as the constraint-consistent operational space inertia matrix. For the moment, we will assume full actuation $\mathbf{B} = \mathbf{I}$. In the latter, we will deal with the underactuation constraint separately. Then, if we replace all non-linear terms

with $\mathbf{h}_c$ and $\boldsymbol{\tau}$ with $\mathbf{J}_x^\top \mathbf{F}$, we can compactly write the above equation as

$$\boldsymbol{\Lambda}_c \ddot{\mathbf{x}} + \mathbf{h}_c = \mathbf{F} + \mathbf{F}_x \quad (16)$$

where $\mathbf{F}$ is the operational forces due to actuation. In order to achieve the impedance response of Equation (10), we can define the control law as

$$\mathbf{F} = \mathbf{h}_c + \boldsymbol{\Lambda}_c \ddot{\mathbf{x}}_d - \boldsymbol{\Lambda}_c \boldsymbol{\Lambda}_d^{-1}(\mathbf{D}_d \dot{\mathbf{e}} + \mathbf{K}_d \mathbf{e}) + (\boldsymbol{\Lambda}_c \boldsymbol{\Lambda}_d^{-1} - \mathbf{I})\mathbf{F}_x \quad (17)$$

We notice that if we sacrifice the ability to shape the inertia matrix by specifying $\boldsymbol{\Lambda}_d = \boldsymbol{\Lambda}_c$ our control law simplifies to

$$\mathbf{F} = \mathbf{h}_c + \boldsymbol{\Lambda}_c \ddot{\mathbf{x}}_d - \mathbf{D}_d \dot{\mathbf{e}} - \mathbf{K}_d \mathbf{e} \quad (18)$$

The advantage of the above control law is that we do not need to measure the external disturbances $\mathbf{F}_x$ (e.g., with a force/torque sensor) to realize the impedance response.

On the other hand, by assigning $\boldsymbol{\Lambda}_d = \boldsymbol{\Lambda}_c$, the impedance gains of Equation (10) can still be tuned by analytical techniques based on different configurations (see in Angelini et al., 2019), whereas HQP controllers rely on manual trial and error tuning.

Remark. *It should be noted that if $\mathbf{J}_x$ is rank deficient, $\mathbf{J}_x \mathbf{M}_c^{-1}\mathbf{PJ}_x^\top$ is not invertible, and we cannot determine $\boldsymbol{\Lambda}_c = (\mathbf{J}_x \mathbf{M}_c^{-1}\mathbf{PJ}_x^\top)^{-1}$. This may happen, for example, for the case when the robot is trotting (only two point feet on the ground) or for a swing leg when the leg is in singularity configuration. In these cases we can approximate $\boldsymbol{\Lambda}_c$ by computing a singularity robust inverse of $\mathbf{J}_x \mathbf{M}_c^{-1}\mathbf{PJ}_x^\top$, for example, by Singular Value Decomposition (SVD) with zero or near-zero eigenvalues removed.*

3.2. Imposing Underactuation Constraints

The previous derivation of the control law (Equation 18) for Cartesian impedance control is based on an assumption of $\mathbf{B} = \mathbf{I}$. In fact, a quadruped robot is a typical underactuated system, i.e., the definition of $\mathbf{B}$ in Equation (2) is not an identity matrix. We need to satisfy the additional underactuation constraint

$$\boldsymbol{\tau}_m = \mathbf{B}\boldsymbol{\tau}_m \quad (19)$$

Our control law of Equation (18) is defined in operational space. Although in general $\mathbf{J}_x^\top \mathbf{F} \neq \mathbf{BJ}_x^\top \mathbf{F}$, we may still be able to satisfy (19) by adding constraint forces in constraint space

$$\boldsymbol{\tau}_m = \mathbf{J}_x^\top \mathbf{F} + (\mathbf{I} - \mathbf{P})\boldsymbol{\tau}_0 = \mathbf{BJ}_x^\top \mathbf{F} + \mathbf{B}(\mathbf{I} - \mathbf{P})\boldsymbol{\tau}_0 \quad (20)$$

and we can solve for $\boldsymbol{\tau}_0$

$$(\mathbf{I} - \mathbf{B})\mathbf{J}_x^\top \mathbf{F} = -(\mathbf{I} - \mathbf{B})(\mathbf{I} - \mathbf{P})\boldsymbol{\tau}_0 \quad (21)$$

$$\boldsymbol{\tau}_0 = -[(\mathbf{I} - \mathbf{B})(\mathbf{I} - \mathbf{P})]^+ (\mathbf{I} - \mathbf{B})\mathbf{J}_x^\top \mathbf{F} \quad (22)$$

Subsequently, the Cartesian impedance control law in motion space with consideration of underactuation becomes

$$\boldsymbol{\tau}_m = \mathbf{J}_x^\top \mathbf{F} - (\mathbf{I} - \mathbf{P})[(\mathbf{I} - \mathbf{B})(\mathbf{I} - \mathbf{P})]^+ (\mathbf{I} - \mathbf{B})\mathbf{J}_x^\top \mathbf{F} \quad (23)$$
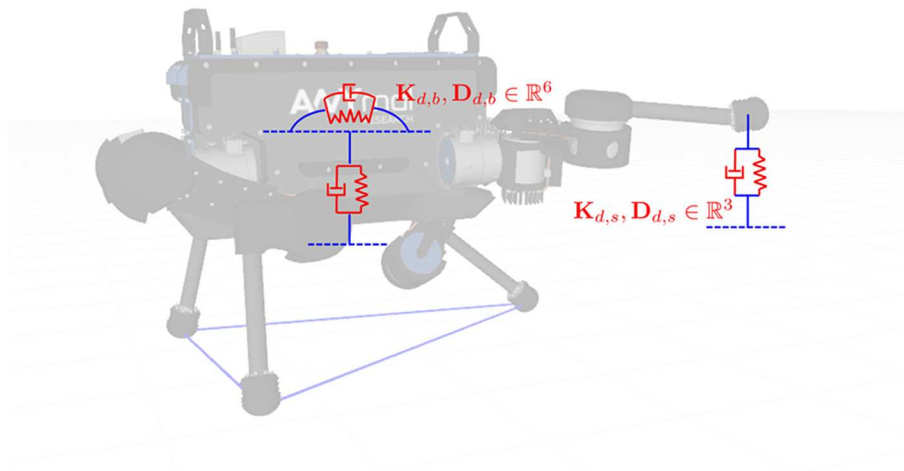
FIGURE 2 | The base pose and the swing foot position are controlled by two Cartesian impedance controllers.

Furthermore, as described in Mistry and Righetti (2012), it could be simplified to

$$\boldsymbol{\tau}_m = (\mathbf{PB})^+ \mathbf{J}_x^\top \mathbf{F} \qquad (24)$$

where $\mathbf{F}$ is defined in Equation (18).

The above Cartesian impedance controller could also be applied to swing legs by replacing $\mathbf{J}_x$ with corresponding Jacobian matrix of a swing foot. If we define the Jacobian matrix of torso and one swing leg as $\mathbf{J}_b$ and $\mathbf{J}_s$, respectively, the only difference is $\mathbf{J}_s \in \mathbb{R}^{3 \times (n+6)}$ due to the point foot while $\mathbf{J}_b \in \mathbb{R}^{6 \times (n+6)}$. Furthermore, let $\mathbf{F}_b \in \mathbb{R}^6$ and $\mathbf{F}_s \in \mathbb{R}^3$ represent the control law of Equation (18) for torso and swing feet, resulting in the torque command of motion space as follows

$$\boldsymbol{\tau}_m = (\mathbf{PB})^+ \left(\mathbf{J}_b^\top \mathbf{F}_b + \sum_{i=0}^{n_{legs}} \omega \mathbf{J}_{s,i}^\top \mathbf{F}_{s,i} \right) \qquad (25)$$

where weight ω is 1 when leg i is in swing, and 0 otherwise. **Figure 2** depicts the case of two Cartesian impedance controllers applied to base and one foot, respectively.

3.3. Model Error Estimation and Compensation

One of the benefits of Cartesian impedance control is that the impedance response of Equation (10) provides us with a way to estimate the disturbances using motion errors

$$\hat{\mathbf{F}}_x = \boldsymbol{\Lambda}_d \ddot{\mathbf{e}} + \mathbf{D}_d \dot{\mathbf{e}} + \mathbf{K}_d \mathbf{e} \qquad (26)$$

It should be noted that $\hat{\mathbf{F}}_x$ contains external disturbances as well as model errors. Considering the existing model error, the control law of Equation (18) becomes

$$\mathbf{F} = \hat{\mathbf{h}}_c + \hat{\boldsymbol{\Lambda}}_c \ddot{\mathbf{x}}_d - \mathbf{D}_d \dot{\mathbf{e}} - \mathbf{K}_d \mathbf{e} \qquad (27)$$

where $\hat{\mathbf{h}}_c$ and $\hat{\boldsymbol{\Lambda}}_c$ denote the approximate items in computation. Substituting Equation (27) into operational space dynamics of Equation (16) yields

$$\hat{\boldsymbol{\Lambda}}_c \ddot{\mathbf{e}} + \mathbf{D}_d \dot{\mathbf{e}} + \mathbf{K}_d \mathbf{e} = (\mathbf{h}_c - \hat{\mathbf{h}}_c) + (\boldsymbol{\Lambda}_c - \hat{\boldsymbol{\Lambda}}_c) \ddot{\mathbf{x}} + \mathbf{F}_x \qquad (28)$$

Further, the above equation implies $\hat{\mathbf{F}}_x = (\mathbf{h}_c - \hat{\mathbf{h}}_c) + (\boldsymbol{\Lambda}_c - \hat{\boldsymbol{\Lambda}}_c) \ddot{\mathbf{x}} + \mathbf{F}_x$ where not only external disturbances are involved, but also model errors. In the case where small external disturbances exist, Equation (28) can be used to estimate model errors.

This feature is useful for legged robots to carry unknown objects. If we feed the model error (e.g., caused by adding an object of unknown mass onto the robot) back into the control law of Equation (18), the position errors due to model errors will be removed. Therefore, the robot may carry unknown objects by compensating for estimated model errors. The control law in motion space with model-error compensation becomes

$$\mathbf{F} = \hat{\mathbf{h}}_c + \hat{\boldsymbol{\Lambda}}_c \ddot{\mathbf{x}}_d - \mathbf{D}_d \dot{\mathbf{e}} - \mathbf{K}_d \mathbf{e} + \hat{\mathbf{F}}_x \qquad (29)$$

where $\hat{\mathbf{F}}_x$ is constant by computing Equation (28) once at the beginning of switching on model error compensation.

By contrast, HQP controllers cannot use motion errors to estimate disturbances because the gains of HQP controllers have different scaling. We will verify this point in the experimental section.

4. QUADRATIC PROGRAMMING

The control scheme outlined above does not yet impose physical feasibility on the contact forces and joint torques. In this section, these constraints are introduced, and we present our quadratic programming approach to handle them.

4.1. Physical Constraints

The contact forces at the feet should be sufficient to prevent the separation or sliding of the contact. Additionally, motor torques typically have saturation values. To formalize the contact force constraint, the force at each foot is divided into tangential components λ_x and λ_y, and normal component λ_z. This leads to the following constraints:

- Unilateral constraint to avoid loss of contact

$$\lambda_z \geq 0 \tag{30}$$

- Friction cone constraint to avoid slipping

$$\mu \lambda_z \geq \sqrt{\lambda_x^2 + \lambda_y^2} \tag{31}$$

where μ is the friction coefficient at the contact point.

The saturation of torque command gives.

$$\boldsymbol{\tau}_{\min} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_{\max} \tag{32}$$

4.2. Optimization Formulation

Our previous work (Xin et al., 2018) minimized a quadratic cost function of the torque commands, thereby minimizing the torques.

$$\underset{\boldsymbol{\tau}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{\tau}\|_2^2 \tag{33}$$

where $\boldsymbol{\tau} = \boldsymbol{\tau}_m + \boldsymbol{\tau}_c$. To track trajectories and to deal with external disturbances, the Cartesian impedance controller (25) defines the motion space torque command $\boldsymbol{\tau}_m$. This partial command is included in the optimization using the linear equality constraint:

$$\mathbf{PB}\boldsymbol{\tau} = \boldsymbol{\tau}_m \tag{34}$$

The friction forces are computed from the optimization variable $\boldsymbol{\tau}$ based on (9) and simplified to be

$$\boldsymbol{\lambda} = \boldsymbol{\eta} + \boldsymbol{\rho} \tag{35}$$

where $\boldsymbol{\eta} = -(\mathbf{J}_c^\top)^+(\mathbf{I} - \mathbf{P})(\mathbf{I} - \mathbf{M}\mathbf{M}_c^{-1}\mathbf{P})\mathbf{B}\boldsymbol{\tau}$ and $\boldsymbol{\rho} = (\mathbf{J}_c^\top)^+(\mathbf{I} - \mathbf{P})[(\mathbf{I} - \mathbf{M}\mathbf{M}_c^{-1}\mathbf{P})\mathbf{h} + \mathbf{M}\mathbf{M}_c^{-1}\dot{\mathbf{P}}\dot{\mathbf{q}}]$. Then we can transform the unilateral constraint and friction cone constraint to be written in terms of decision variable $\boldsymbol{\tau}$. The quadratic programming problem is now formulated as:

$$\begin{aligned} \underset{\boldsymbol{\tau}}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{\tau}\|_2^2 \\ \text{subject to} \quad & \mathbf{PB}\boldsymbol{\tau} = \boldsymbol{\tau}_m \\ & \eta_z^i + \rho_z^i \geq 0 \\ & \mu(\eta_z^i + \rho_z^i) \geq \sqrt{(\eta_x^i + \rho_x^i)^2 + (\eta_y^i + \rho_y^i)^2} \\ & \boldsymbol{\tau}_{\min} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_{\max} \end{aligned} \tag{36}$$

where the index i signifies the ith stance leg.

4.3. Adding Trade-Off Between Multiple Constraints

In practice, there may not be a physically feasible solution that achieves both the desired motion and impedance behavior. For example, tangential contact forces may increase due to the desired acceleration for walking forward, causing contact forces to violate the friction cone. To avoid a need for replanning the desired motion, we propose a trade-off between the constraints. The inequality constraints cannot be traded-off because they enforce physical limitations and stability of the system. On the other hand, the equality constraint enforces the operational space tasks. We will sacrifice motion performance to ensure feasibility of the physical (inequality) constraints when there is no possibility of finding a solution that satisfies both types of constraints. Here, we employ a trade-off of the trajectory tracking control to seek a compromise in low level control. The trade-off is implemented by moving the equality constraint into the cost function to compose a least squares optimization.
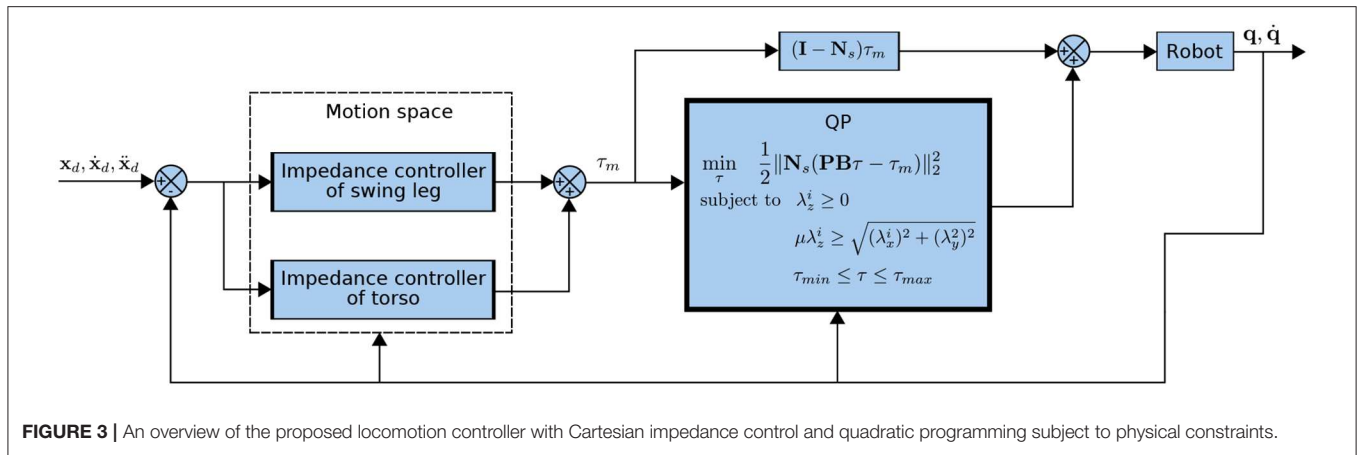
$$\begin{aligned} \underset{\boldsymbol{\tau}}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{PB}\boldsymbol{\tau} - \boldsymbol{\tau}_m\|_2^2 \\ \text{subject to} \quad & \eta_z^i + \rho_z^i \geq 0 \\ & \mu(\eta_z^i + \rho_z^i) \geq \sqrt{(\eta_x^i + \rho_x^i)^2 + (\eta_y^i + \rho_y^i)^2} \\ & \boldsymbol{\tau}_{\min} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_{\max} \end{aligned} \tag{37}$$

The above formulation means that the QP optimization will try to find the admissible torques that are as close as possible to the torque command of motion space required for the desired motion performance. It should be noted that if the QP needs to trade-off the motion performance it will change the swing trajectory more than the torso trajectory, as the torso contributes more to contact forces than the swing leg. Alternatively, we could add a weighting matrix to balance the modification of different parts if it is necessary. However, we would then need to tune the weighting matrix in practice. Therefore, we recommend using a null space projection of the swing legs to project the cost function, in order to enforce accurate swing leg performance. After the QP provides the optimized torque command based on the torso motion, we will add the dynamic consistent torques for the swing legs. The QP in the null space of the swing legs becomes

$$\begin{aligned} \underset{\boldsymbol{\tau}}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{N}_s(\mathbf{PB}\boldsymbol{\tau} - \boldsymbol{\tau}_m)\|_2^2 \\ \text{subject to} \quad & \eta_z^i + \rho_z^i \geq 0 \\ & \mu(\eta_z^i + \rho_z^i) \geq \sqrt{(\eta_x^i + \rho_x^i)^2 + (\eta_y^i + \rho_y^i)^2} \\ & \boldsymbol{\tau}_{\min} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_{\max} \end{aligned} \tag{38}$$

where $\mathbf{N}_s = \mathbf{I} - \mathbf{J}_s^+\mathbf{J}_s$, $\mathbf{J}_s = \mathbf{J}_{s,1}$ for static walking gait, $\mathbf{J}_s = \begin{pmatrix} \mathbf{J}_{s,1} \\ \mathbf{J}_{s,2} \end{pmatrix}$ for the gaits with only two supporting legs, such as trotting gait, and $\mathbf{J}_s = \mathbf{0}$ for four stance legs case. The final torque command will be

$$\boldsymbol{\tau} = \boldsymbol{\tau}^* + (\mathbf{I} - \mathbf{N}_s)\boldsymbol{\tau}_m \tag{39}$$

FIGURE 3 | An overview of the proposed locomotion controller with Cartesian impedance control and quadratic programming subject to physical constraints.

The controller proposed above is represented as a block diagram in **Figure 3**.

Remark. *One notices that the friction cone constraints are quadratic constraints. Therefore, the quadratic programming of Equation (38) is quadratically constrained quadratic programming. In order to use standard active set methods, we approximate the friction cone with a linearized 4-edge pyramid. This results in a linearly constrained quadratic programming problem, which is readily solved (Goldfarb and Idnani, 1983).*

5. EXPERIMENTS AND DISCUSSION

To validate our proposed controller, three sets of experiments were performed: a comparison the HQP controller, model error compensation, and walking on challenging terrains. This section treats each of the experiments in turn, and provides a setup description, results, and discussion for all of them.

The experiments have been performed using ANYmal, a torque-controlled quadruped robot made by ANYbotics. The robot weights ∼35 kg and has 12 joints actuated by the Series Elastic Actuator (SEA). The control cycle runs at 400Hz. The on-board computer running controllers has an Intel 4th generation (Haswell ULT) i7-4600U (1.4–2.1GHz) processor and two HX316LS9IBK2/16 DDR3L memory cards. Experiment videos can be found in the **Supplementary Video 1**.

5.1. Comparison With HQP Controller

To show the efficacy of our proposed controller, we compare it with an HQP-computer. This comparison considers both the computational efficiency and the capacity to estimate external disturbances. Our implementation of the HQP controller follows the original framework from De Lasa et al. (2010), adding a dimension-reduction to that framework in order to enhance computational efficiency. This dimension-reduction, proposed in Herzog et al. (2016) removes the motor-torque from the decision variables by using the equations of motion of the robot. The remaining decision

variables are the torso and joint accelerations and the contact forces. An active-set QP solver was employed to solve QPs in both controllers.
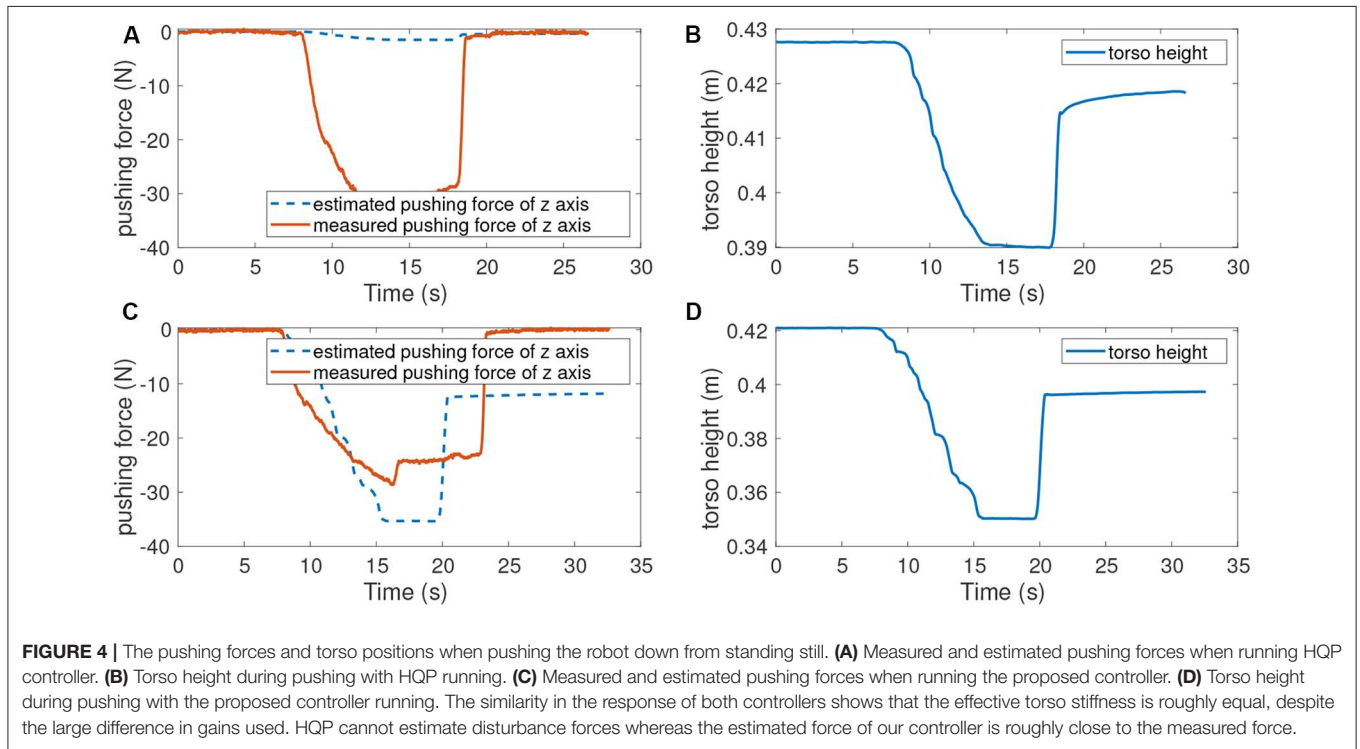
5.1.1. Computational Efficiency

To evaluate the computational efficiency of the proposed whole-body controller against the HQP-controller, we compare their execution time on the robot hardware. A comparison of average computation times while executing a fixed static walking gait is provided in **Table 1**.

The computation time analysis shows that our controller is faster than the HQP controller. This is expected since our controller only requires one QP although it computes a few pseudo-inverses of matrices. By contrast, HQP needs to compute more QPs and also needs to compute null-space projectors which include pseudo-inverses as well. In fact, HQP needs to solve at least two QPs for dynamic feasibility and trajectory tracking tasks. In order to save energy, a third QP has to be added to minimize torque commands. Usually, those are the three fundamental QPs required by HQP controllers (Bellicoso et al., 2019). For walking gaits, if we separate swing legs and torso into different prioritized tasks, one more QP has to be solved. In this experiment, the HQP controller solved three QPs for the four-legged stance phase and four QPs during a one leg swing. In Herzog et al. (2016), the first QP is solved by pseudo-inverse when moving friction cone constraints into a lower priority, which aims to save computation time. But, based on **Table 1**, the computation time of HQP is still greater than the proposed controller, even omitting the computation time of QP_1. Furthermore, Herzog et al. (2016) also suggests parallelizing the computations of null-space projectors and QPs. In that case, the total computation time of HQP will be the sum of solving QPs, which is close to our controller's according to **Table 1**. What's more, as pointed out in Escande et al. (2014) and Herzog et al. (2016), the size of decision variables can be reduced over the levels of hierarchies by additional SVD, which could potentially improve the computation efficiency of HQP even further. Readers should keep in mind that we did not employ that technique in the comparison of **Table 1**.

TABLE 1 | The computation efficiency comparison between proposed controller (P.C.) and HQP controller during static walking.

Controller	Phase	τ_m	QP_1	N_1	QP_2	N_2	QP_3	N_3	QP_4	Total (m/s)
P.C.	Stance	0.203	0.064							0.267
	Swing	0.147	0.095							0.242
HQP	Stance		0.09	0.124	0.1	0.024	0.09			0.428
	Swing		0.075	0.081	0.098	0.02	0.072	0.025	0.079	0.45

The results are the means of 2,000 samples, respectively. HQP controller needs to compute two null-space projectors N_i and solve three QPs during Stance phase, whereas one more QP is added for swing leg control during Swing phase. The proposed controller only needs to compute analytical Cartesian impedance control τ_m and solve one QP. The computation time of HQP is almost doubled compared to the proposed controller either in Stance phase or in Swing phase.

FIGURE 4 | The pushing forces and torso positions when pushing the robot down from standing still. **(A)** Measured and estimated pushing forces when running HQP controller. **(B)** Torso height during pushing with HQP running. **(C)** Measured and estimated pushing forces when running the proposed controller. **(D)** Torso height during pushing with the proposed controller running. The similarity in the response of both controllers shows that the effective torso stiffness is roughly equal, despite the large difference in gains used. HQP cannot estimate disturbance forces whereas the estimated force of our controller is roughly close to the measured force.

5.1.2. External Disturbance Estimation

The ability of the two controllers to estimate external disturbances is assessed on the robot by comparing their response to similar force perturbations while standing still. The operator pushed the torso along the vertical axis using about 30N force, and the recorded position displacements are used to estimate the external perturbation received by the robot. As would happen in practice, we only use the multiplication of $\mathbf{K}_d$ and displacement errors as the estimated disturbance instead of the entire Equation (28) since accelerations and velocities are noisy.

Figure 4 shows the measured pushing forces and torso heights, which have been obtained using the gains reported in **Table 2**. The position errors caused by pushing forces are similar between the two controllers, which means they produce approximately the same effective stiffness. The force estimated by the proposed controller is $\mathbf{K}_{d,z}\mathbf{e}_z \approx 35N$. This estimation is close to the measured pushing force 30N. However, the relationship between the effective stiffness and the controller gains is not straightforward for the HQP-controller. For this controller, the gains map from an error to a desired acceleration, not to a desired force, as is the case for our proposed controller. As a result, the $\mathbf{K}_d$ gains of the HQP controller are much smaller than the ones of the proposed controller, meaning the HQP estimated force is 1.6N which is much lower than the received external perturbation. A conversion factor could be found, either based on the equations of motion or using experiments, such as this one. However, the conversion factor would depend on the robot configuration, which means the HQP-controller would require extensive tuning to be able to perform disturbance estimation. Therefore, the proposed controller is more suitable for performing the forces estimation for the compensation of constant external perturbations, such as the additional load carried by the robot.

5.2. Model Error Compensation

Legged robots may be deployed to move small loads which may not always be known; thus, it is important to adapt the robot

behavior to maintain stable and efficient locomotion. Equation (29) describes how the proposed controller may be modified to use the external perturbation estimation to perform a model error compensation. For the experiment, the robot was loaded with a 10 kg mass and walked with and without activating the model compensation, which has been enabled after about 23 s. **Table 3** contains the gains of the proposed controller in the experiment. The recorded video for this experiment can be found in the **Supplementary Material**.

Figure 5 shows the snapshots from the experiments of adding weight to walking with it. **Figure 6** is the recorded position error along the z axis during the experiment. After enabling the compensation, the torso immediately rose as shown in **Figure 6** and continued its locomotion with a reduced position error. It should be noted that the estimated model error is about 65N. A bias of 35N has been added in this experiment to account for friction and other uncertainties of a real robot. The results demonstrate that the position error

can be reduced by model error compensation when carrying unknown objects, which is useful for deployment of legged robots in th real world. However, an important upgrade required to deploy this feature in a real scenario is continuous real-time model adaptation, which has currently been left as future work to develop.

5.3. Walking on Challenging Terrains

We tested the proposed controller for its ability to tackle challenging terrains. We tested walking on a low friction surface (PTFE Sheets, fiction coefficient of 0.22 as shown in **Figure 7**), and the feasibly of climbing a significant incline (30° slope in **Figure 8**). As perception is not the focus of this work, the friction coefficient and slope angle parameters, to be used by the controller in the optimization stage, are set manually. Locomotion performance for these tasks may be seen in the accompanying video.

We did not alter the step planning in our experiment, but rather used the default ANYbotics static gait planner. This means the planned locomotion trajectory was not optimized for

TABLE 2 | Gains for HQP controller and the proposed controller used in the stiffness comparison experiment.

Parameter	Symbol	HQP	Proposed controller
Stiffness gains for torso	K_d	diag(40,40,40,60,60,60)	diag(1000,800,500,200,200,200)
Damping gains for torso	D_d	diag(9,9,9,7,7,7)	diag(250,250,90,10,10,10)

Although the gains are so different in the two controllers especially for torso control, the torso showed similar stiffness performance.

TABLE 3 | Gains of the proposed controller during carrying weight experiment.

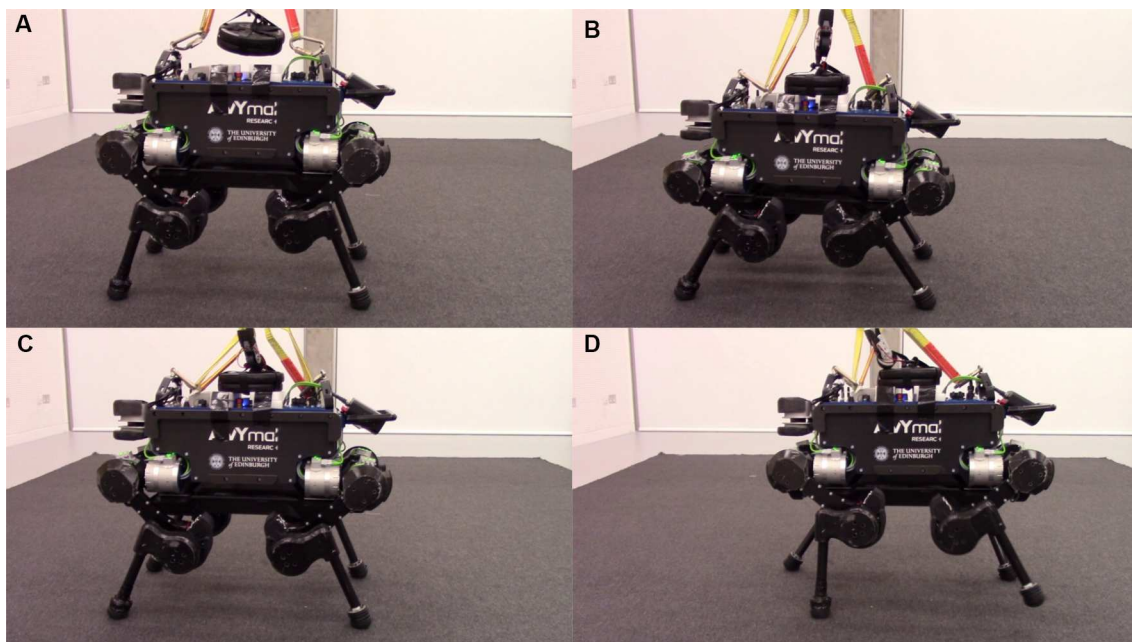| Parameter	Symbol	Proposed controller
Stiffness gains for torso	K_d	diag(1000,800,1000,200,200,200)
Damping gains for torso	D_d	diag(250,250,250,10,10,10)
Stiffness gains for swing foot	K_d	diag(250,250,250)
Damping gains for swing foot	D_d	diag(20,20,20)

FIGURE 5 | Snapshots of carrying 10 kg weight with model error compensation. **(A)** Before adding the weight. **(B)** After adding the weight, the torso went down. **(C)** The torso went up after enabling model error compensation. **(D)** Walking with the object with model error compensation.

our particular tasks or evaluated for feasibility. Thus, during execution, if the controller does not trade-off trajectory tracking and constraint satisfaction, the robot may slip due to the inability of the QP finding a feasible solution. We further evaluate this trade-off by analyzing trajectory tracking errors when statically walking on flat terrain, while varying the friction coefficient parameter used by the controller.

As shown in **Figure 9**, the trajectory tracking error along the z direction has an occasional sharp rise particularly when the friction coefficient μ is equal to 0.1 and 0.08. This indicates that trajectory tracking performance is sacrificed in order to maintain the friction cone constraint. We also see that the cost function value of QP increases, due to the increase of trajectory tracking error. In theory, the value of cost function should be zero if the trade-off is not triggered. The model error explains why the deviation between $\mathbf{PB\tau}$ and τ_m is not equal to zero even when

$\mu = 0.5$. The controller in our previous work (Xin et al., 2018) does not have this trade-off feature, and this is an improvement over our previous work.

6. CONCLUSIONS

A semi-analytical locomotion controller incorporating an analytical Cartesian impedance controller and a QP optimization for quadruped robots is presented in the paper. Cartesian impedance controllers can track desired end-effector trajectories and can also be used to estimate external disturbances. The disturbance estimation is applied to model error compensation in the case of making a robot carry unknown objects. The QP optimization guarantees physical feasibility by sacrificing trajectory tracking if the torque command required to track the desired trajectory does not satisfy physical constraints. The benefit of this trade-off strategy enables the robot to walk on synthetic ice and a 30°
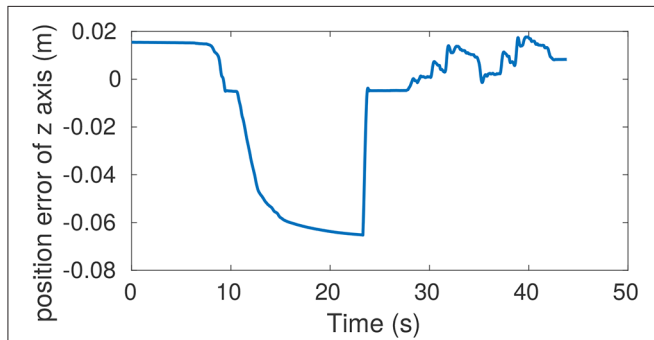

FIGURE 6 | Measured torso height during carrying 10 kg weight. At 10 s, the weight is slowly lowered onto the robot, resulting in an increasing position error. At 23 s the force compensation is enabled, after which the robot takes two steps. The small difference between the initial and final position error shows the efficacy of the disturbance compensation.
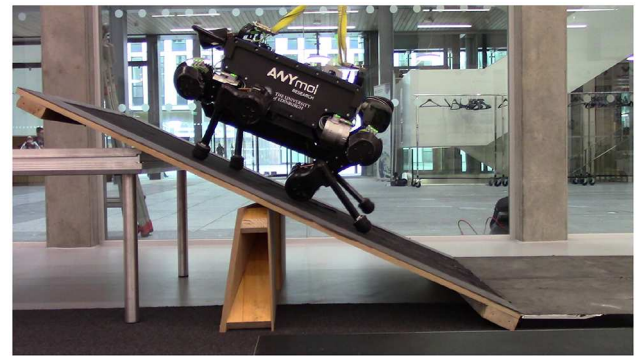

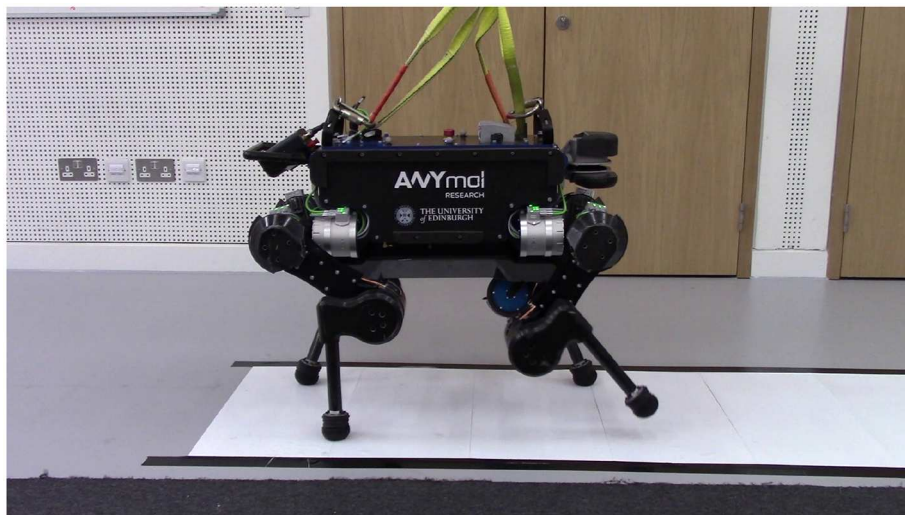**FIGURE 8 |** ANYmal climbing up on a 30° slope.

FIGURE 7 | ANYmal walking on slippery synthetic ice. The estimated friction coefficient between ANYmal's feet and the synthetic ice is 0.22. The friction coefficient in our controller needs to be ≤0.2 in order to avoid slipping on the synthetic ice.
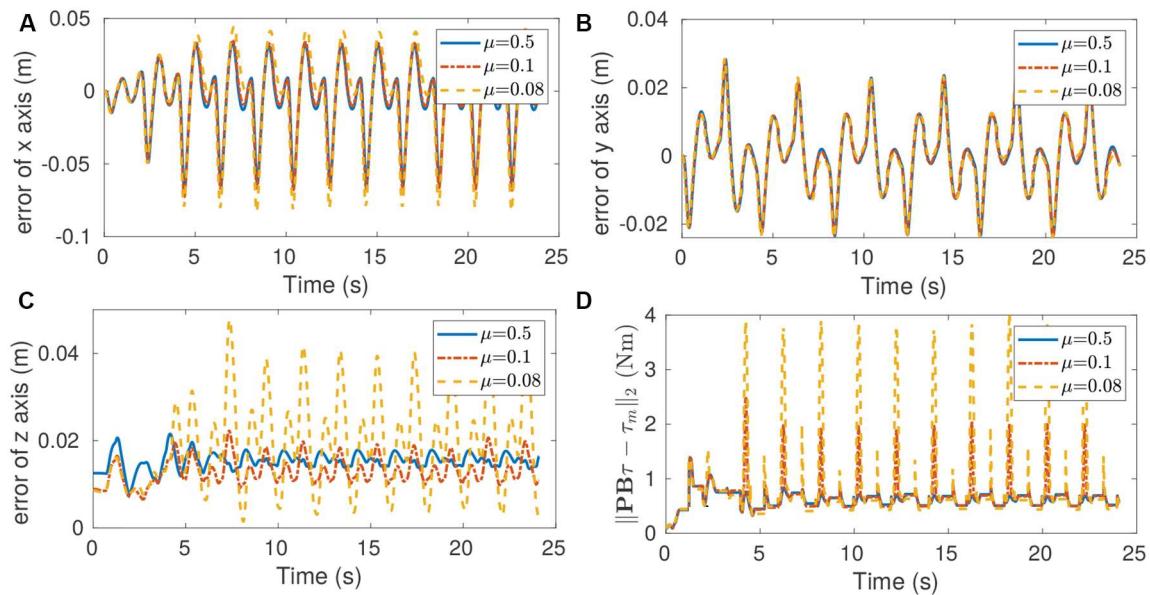
FIGURE 9 | Tracking performance when walking with a static gait on high-traction flat terrain for different values of the friction coefficient used by the controller. **(A)** Position tracking error of x direction. **(B)** Position tracking error of y direction. **(C)** Position tracking error of z direction. **(D)** Deviation between $\mathbf{PB}\tau$ and τ_m. The increasing errors for small friction coefficients show the trade-off in tracking performance made to guarantee feasibility of the friction cone constraint. In particular, a behavior emerges in which the controller deviates from the target motion in z direction, in order to increase normal contact forces when walking with a small friction coefficient. The trade-off between tracking error and constraint satisfaction is also visible in the increasing deviation between $\mathbf{PB}\tau$ and τ_m.

ramp. Besides, the computation time is verified to be less than other fully optimization-based whole-body controllers since we only need to solve one QP with less decision variables compared to other controllers. Future work will focus on applying this control framework to dynamic gaits, such as trotting.

DATA AVAILABILITY STATEMENT

All datasets generated for this study are included in the article/**Supplementary Material**.

AUTHOR CONTRIBUTIONS

GX, H-CL, and MM significantly contributed to the development of the presented approach. GX, WW, and CT added the least-square optimization to the framework. All authors have contributed to the writing and they have reviewed the submitted version.

FUNDING

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frobt.2020.00048/full#supplementary-material

Supplementary Video 1 | Model error compensation and walking on challenging terrains of a quadruped robot.

REFERENCES

Aghili, F. (2005). A unified approach for inverse and direct dynamics of constrained multibody systems based on linear projection operator: applications to control and simulation. *IEEE Trans. Robot.* 21, 834–849. doi: 10.1109/TRO.2005.851380

Albu-Schaffer, A., Ott, C., Frese, U., and Hirzinger, G. (2003). Cartesian impedance control of redundant robots: recent results with the DLR-light-weight-arms. *IEEE Int. Conf. Robot. Autom.* 3, 3704–3709. doi: 10.1109/ROBOT.2003.1242165

Angelini, F., Xin, G., Wolfslag, W. J., Tiseo, C., Mistry, M., Garabini, M., et al. (2019). "Online optimal impedance planning for legged robots," in *IEEE International Conference on Intelligent Robots and Systems* (Macau). doi: 10.1109/IROS40897.2019.8967696

Bellicoso, C. D., Gehring, C., Hwangbo, J., Fankhauser, P., and Hutter, M. (2016). "Perception-less terrain adaptation through whole body control and hierarchical optimization," in *IEEE-RAS International Conference on Humanoid Robots*, (Cancun) 558–564. doi: 10.1109/HUMANOIDS.2016.7803330

Bellicoso, C. D., Kramer, K., Stäuble, M., Sako, D., Jenelten, F., Bjelonic, F., et al. (2019). "Alma-articulated locomotion and manipulation for a torque-controllable robot," in *International Conference on Robotics and Automation (ICRA 2019)* (Montreal). doi: 10.1109/ICRA.2019.8794273

Boaventura, T., Semini, C., Buchli, J., Frigerio, M., Focchi, M., and Caldwell, D. G. (2012). "Dynamic torque control of a hydraulic quadruped robot," in *2012 IEEE*

International Conference on Robotics and Automation (Saint Paul, MN: IEEE), 1889–1894. doi: 10.1109/ICRA.2012.6224628

De Lasa, M., Mordatch, I., and Hertzmann, A. (2010). Feature-based locomotion controllers. *ACM Trans. Graph.* 29:131. doi: 10.1145/1778765.1781157

Escande, A., Mansard, N., and Wieber, P.-B. (2014). Hierarchical quadratic programming: fast online humanoid-robot motion generation. *Int. J. Robot. Res.* 33, 1006–1028. doi: 10.1177/0278364914521306

Gehring, C., Coros, S., Hutter, M., Bloesch, M., Hoepflinger, M. A., and Siegwart, R. (2013). "Control of dynamic gaits for a quadrupedal robot," in *2013 IEEE International Conference on Robotics and Automation* (Karlsruhe: IEEE), 3287–3292. doi: 10.1109/ICRA.2013.6631035

Goldfarb, D., and Idnani, A. (1983). A numerically stable dual method for solving strictly convex quadratic programs. *Math. Program.* 27, 1–33. doi: 10.1007/BF02591962

Herzog, A., Rotella, N., Mason, S., Grimminger, F., Schaal, S., and Righetti, L. (2016). Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid. *Auton. Robots* 40, 473–491. doi: 10.1007/s10514-015-9476-6

Hutter, M., Sommer, H., Gehring, C., Hoepflinger, M., Bloesch, M., and Siegwart, R. (2014). Quadrupedal locomotion using hierarchical operational space control. *Int. J. Robot. Res.* 33, 1047–1062. doi: 10.1177/0278364913519834

Khatib, O. (1987). A unified approach for motion and force control of robot manipulators: the operational space formulation. *IEEE J. Robot. Autom.* 3, 43–53. doi: 10.1109/JRA.1987.1087068

Lee, Y., Hwang, S., and Park, J. (2016). Balancing of humanoid robot using contact force/moment control by task-oriented whole body control framework. *Auton. Robots* 40, 457–472. doi: 10.1007/s10514-015-9509-1

Lin, H.-C., Smith, J., Babarahmati, K. K., Dehio, N., and Mistry, M. (2018). "A projected inverse dynamics approach for multi-arm cartesian impedance control," in *IEEE International Conference on Robotics and Automation* (Brisbane). doi: 10.1109/ICRA.2018.8461202

Mansard, N. (2012). "A dedicated solver for fast operational-space inverse dynamics," in *2012 IEEE International Conference on Robotics and Automation (ICRA)* (Saint Paul, MN: IEEE), 4943–4949. doi: 10.1109/ICRA.2012.6224851

Mistry, M., Buchli, J., and Schaal, S. (2010). "Inverse dynamics control of floating base systems using orthogonal decomposition," in *2010 IEEE International Conference on Robotics and Automation (ICRA)* (Anchorage, AK: Citeseer), 3406–3412. doi: 10.1109/ROBOT.2010.5509646

Mistry, M., Nakanishi, J., Cheng, G., and Schaal, S. (2008). "Inverse kinematics with floating base and constraints for full body humanoid robot control," in *Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots* (Daejeon: IEEE), 22–27. doi: 10.1109/ICHR.2008.4755926

Mistry, M., and Righetti, L. (2012). "Operational space control of constrained and underactuated systems," in *Robotics: Science and Systems VII* (Sydney), 225–232. doi: 10.15607/RSS.2011.VII.031

Pratt, J., Chew, C.-M., Torres, A., Dilworth, P., and Pratt, G. (2001). Virtual model control: an intuitive approach for bipedal locomotion. *Int. J. Robot. Res.* 20, 129–143. doi: 10.1177/02783640122067309

Saab, L., Ramos, O. E., Keith, F., Mansard, N., Soueres, P., and Fourquet, J. Y. (2013). Dynamic whole-body motion generation under rigid contacts and other unilateral constraints. *IEEE Trans. Robot.* 29, 346–362. doi: 10.1109/TRO.2012.2234351

Sentis, L., and Khatib, O. (2006). "A whole-body control framework for humanoids operating in human environments," in *ICRA 2006. Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006* (Orlando, FL: IEEE), 2641–2648. doi: 10.1109/ROBOT.2006.1642100

Sentis, L., Park, J., and Khatib, O. (2010). Compliant control of multicontact and center-of-mass behaviors in humanoid robots. *IEEE Trans. Robot.* 26, 483–501. doi: 10.1109/TRO.2010.2043757

Xin, G., Lin, H.-C., Smith, J., Cebe, O., and Mistry, M. (2018). "A model-based hierarchical controller for legged systems subject to external disturbances," in *IEEE/RSJ International Conference on Robotics and Automation* (Brisbane), 4375–4382. doi: 10.1109/ICRA.2018.8461172