Bounded haptic teleoperation of a quadruped robot's foot posture for sensing and manipulation

Guiyang Xin[1], Joshua Smith[1], David Rytz[2], Wouter Wolfslag[1], Hsiu-Chin Lin[3] and Michael Mistry[1]

Abstract— This paper presents a control framework to teleoperate a quadruped robot's foot for operator-guided haptic exploration of the environment. Since one leg of a quadruped robot typically only has 3 actuated degrees of freedom (DoFs), the torso is employed to assist foot posture control via a hierarchical whole-body controller. The foot and torso postures are controlled by two analytical Cartesian impedance controllers cascaded by a null space projector. The contact forces acting on supporting feet are optimized by quadratic programming (QP). The foot's Cartesian impedance controller may also estimate contact forces from trajectory tracking errors, and relay the force-feedback to the operator. A 7D haptic joystick, Sigma.7, transmits motion commands to the quadruped robot ANYmal, and renders the force feedback. Furthermore, the joystick's motion is bounded by mapping the foot's feasible force polytope constrained by the friction cones and torque limits in order to prevent the operator from driving the robot to slipping or falling over. Experimental results demonstrate the efficiency of the proposed framework.

I. INTRODUCTION

Haptic teleoperation plays an important role in robotic application cases, such as robot-assisted surgery and nuclear facility decommissioning. Vision-based teleoperation can be enhanced by haptic rendering., e.g., an user might want to remotely inspect a specific area through touching. Haptic feedback also helps visual perception when vision systems lose efficacy in some special cases, e.g., lidar can give wrong terrain maps when the terrain is covered by liquid or grass. Exploring the environment using tactile sensors could mitigate errors in vision system results. In the literature a number of haptic teleoperation works with single arm robots can be found, whereas teleoperating quadruped robots to do manipulation and exploration has been studied little.

Researchers have been working on taking full advantage of multi-legged robots to be more versatile. For humanoid robots it is natural to do manipulation using arms [1][2]. Equipping quadruped robots with an additional arm that is dedicated to manipulation tasks is a common way to enhance locomanipulation [3][4][5] due to the fact that the leg of a quadruped usually does not have enough DoFs nor a sufficiently large workspace to be competent at six dimensional manipulation in Cartesian space. Whole-body controllers can implement foot posture control by coordinating the leg and base, which greatly extends the versatility of quadruped robots in real world deployment.

A fully optimization-based whole-body controller called hierarchical quadratic programming (HQP) has been widely used on legged robots [6][7][8][9]. Multiple tasks are encoded by sequential strictly null-space prioritized QPs, which solve for the torque commands while taking into account joint space dynamics and physical constraints. Solving several QPs online is still time consuming, although decomposition methods are developed to reduce the number of decision variables. Usually three QPs have to be solved for dynamic feasibility, operational tasks and saving energy [5]. Alternatively a non-strictly prioritized weighted QP (WQP) is used to avoid solving sequential QPs by stacking task equations into a weighted quadratic cost function [10][11].

Alternatively, projected inverse dynamics is proposed by [12][13][14] to split up motion space and constraint space so that only one QP in constraint space needs to be solved. In [15], we derived operational space dynamics for floating base robots and then obtained analytical Cartesian impedance controllers in motion space. Although a QP optimization is still required to solve inequality constraints, the analytical Cartesian impedance controller gives us the ability to set gains based on operational space inertia [16] and to estimate external forces because of properly scaled gains. In this paper, we will apply that analytical method to haptic teleoperation for sensing and manipulation.

Since the hierarchical controller will sacrifice low level tasks to satisfy high level tasks, one may not be able to fulfill the constraints at the lower level. For example, if the foot controller has higher priority than the base controller, the base may move outside the support polygon and become unstable. To avoid such failures, we need to set boundaries to restrict teleoperation movement. The commanded end-effector forces should also be bounded based-on physical criteria, as high acting force may cause the supporting feet to slip and/or lose balance due to moment around support polygon edge generated by the acting force. Papers about motion planning have discussed balance maintenance for multiple contact cases [17][18]. This paper will present how to appropriately restrict operator commands by mapping the end-effector feasible wrench polytope (FWP) [19][20] to joystick boundaries.

The contribution of this paper is twofold. Firstly, we extend our hierarchical Cartesian impedance controller to adapt to foot posture teleoperation, and the impedance controller is experimentally proved to be accurate enough to estimate contact forces. Secondly, the range of the teleoperation

[1] G. Xin, J. Smith, W. Wolfslag, and M. Mistry are with Institute of Perception, Action and Behaviour, School of Informatics, The University of Edinburgh, 47 Potterrow, Edinburgh, UK `guiyang.xin@ed.ac.uk`
[2] D. Rytz is with the Oxford Robotics Institute at the University of Oxford, United Kingdom.
[3] H.-C. Lin is with the School of Computer Science/Department of Electrical and Computer Engineering at McGill University, Canada.

joystick is bounded with respect to the physical constraints to ensure safe teleoperation. To the best of authors' knowledge, this is the first time to achieve haptic teleoperation of foot posture for quadruped robots in the literature.

II. CONTROL FRAMEWORK

The joint space dynamics equation of a quadruped robot is

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{h} = \mathbf{B}\boldsymbol{\tau} + \mathbf{J}_c^\top \boldsymbol{\lambda}_c \quad (1)$$

where $\mathbf{q} = \begin{bmatrix} \mathbf{x}_b^\top & \mathbf{q}_j^\top \end{bmatrix}^\top$ denotes the vector of n actuated joint positions ($\mathbf{q}_j \in \mathbb{R}^n$) 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, $\mathbf{J}_c \in \mathbb{R}^{3k\times(n+6)}$ is the constraint Jacobian that describes $3k$ linearly independent constraints, k denotes the number of supporting feet, $\boldsymbol{\lambda}_c \in \mathbb{R}^{3k}$ are constraint forces acting on supporting feet, and

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

is the selection matrix with n dimensional identity matrix $\mathbf{I}_n$. Eq. (1) is subject to the following physical constraints,

$$\mathbf{J}_c\ddot{\mathbf{q}} + \dot{\mathbf{J}}_c\dot{\mathbf{q}} = \mathbf{0} \quad (3)$$

$$\tau_{min} \leq \tau \leq \tau_{max}, \quad i = 1, \dots, n \quad (4)$$

$$\mu\lambda_{z,i} \geq \sqrt{\lambda_{x,i}^2 + \lambda_{y,i}^2}, \quad i = 1, \dots, k \quad (5)$$

$$\lambda_{z,i} \geq 0, \quad i = 1, \dots, k \quad (6)$$

where Eq. (3) means supporting feet should not move relative to the ground, and Eqs. (4), (5) and (6) represent torque saturation, friction cone and unilateral constraints respectively.

A. Cartesian impedance controller

In paper [21], the authors suggested to use a null-space projector $\mathbf{P} = \mathbf{I} - \mathbf{J}_c^+\mathbf{J}_c$ to split up (1) into two orthogonal subspaces,

$$\mathbf{P}\mathbf{M}\ddot{\mathbf{q}} + \mathbf{P}\mathbf{h} = \mathbf{P}\mathbf{B}\boldsymbol{\tau} = \boldsymbol{\tau}_m \quad (7)$$

$$(\mathbf{I} - \mathbf{P})(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{h}) = (\mathbf{I} - \mathbf{P})\mathbf{B}\boldsymbol{\tau} + \mathbf{J}_c^\top \boldsymbol{\lambda}_c = \boldsymbol{\tau}_c + \mathbf{J}_c^\top \boldsymbol{\lambda}_c \quad (8)$$

Eq. (7) represents motion space that is not affected by contact forces of supporting feet; meanwhile, Eq. (8) represents dynamics in constraint space. The advantage of this separation is that we can derive an analytical Cartesian impedance controller with Eq. (7) and solve a QP optimization to generate an extra torque command satisfying constraints of Eqs. (4-6) with Eq. (8) as shown in subsection II-B. The final torque command is composed of two parts as follows,

$$\boldsymbol{\tau} = (\mathbf{P}\mathbf{B})^+\mathbf{P}(\mathbf{J}_s^\top \mathbf{F}_s + \mathbf{N}_s\mathbf{J}_b^\top \mathbf{F}_b) + \boldsymbol{\tau}_c \quad (9)$$

where the first right side term is equivalent to $\boldsymbol{\tau}_m$ resulting from Cartesian impedance control and $\boldsymbol{\tau}_c$ is the result of QP optimization. $\mathbf{J}_s$ denotes the Jacobian matrix of a swing foot, the dimension of which depends on the control task. Since we want to control both the position and orientation of the foot, $\mathbf{J}_s \in \mathbb{R}^{6\times(n+6)}$. $\mathbf{F}_s$ and $\mathbf{F}_b$, derived from operational space dynamics, are Cartesian impedance controllers for the swing foot and the base. $\mathbf{N}_s$ is the dynamic consistent null-space projector [21] of the swing foot, which enforces strictly hierarchical priorities. In the case of foot posture control, $\mathbf{N}_s$ will deal with the overlap between $\mathbf{J}_s$ and $\mathbf{J}_b$, leading to the convenience of leaving base Jacobian $\mathbf{J}_b$ to be always a $\mathbb{R}^{6\times(n+6)}$ matrix. As the torso is in the null-space of the swing foot, the torso is enforced to satisfy the swing foot's motion requirement, which results in automatic motion coordination and reachability extension of foot.

The generic equation of the Cartesian impedance control law for $\mathbf{F}_s$ and $\mathbf{F}_b$ is

$$\mathbf{F}_i = \mathbf{h}_{c,i} + \boldsymbol{\Lambda}_{c,i}\ddot{\mathbf{x}}_{d,i} - \mathbf{D}_{d,i}\dot{\mathbf{e}}_i - \mathbf{K}_{d,i}\mathbf{e}_i, \quad i = s \text{ or } b \quad (10)$$

where $\mathbf{h}_c$ represents the operational space gravity compensation vector, $\boldsymbol{\Lambda}_c$ denotes the operational space inertia matrix, $\mathbf{D}_d$ and $\mathbf{K}_d$ are diagonal matrices containing desired damping and stiffness gains, $\mathbf{e} = \mathbf{x} - \mathbf{x}_d$ is the pose error of end-effector (either the base or the swing foot), $\ddot{\mathbf{x}}_d$ denotes desired acceleration of end-effector. We refer to [15] for more details of $\mathbf{h}_c$ and $\boldsymbol{\Lambda}_c$.

The impedance control law Eq. (10) leads to the following impedance behavior under external disturbances

$$\boldsymbol{\Lambda}_c\ddot{\mathbf{e}} + \mathbf{D}_d\dot{\mathbf{e}} + \mathbf{K}_d\mathbf{e} = \mathbf{F}_x \quad (11)$$

where $\mathbf{F}_x = [\mathbf{f}_x^\top \quad \mathbf{m}_x^\top]^\top \in \mathbb{R}^6$ is the external wrench acting on the end-effector. Obviously if there is no external disturbance, the robot will track desired trajectory accurately with the assumption of using a perfect model. Conversely, we could measure motion error and then estimate external disturbances [14],

$$\hat{\mathbf{F}}_x = \boldsymbol{\Lambda}_c\ddot{\mathbf{e}} + \mathbf{D}_d\dot{\mathbf{e}} + \mathbf{K}_d\mathbf{e} \quad (12)$$

Model error always exists for real robot platforms, leading to a $\hat{\mathbf{F}}_x$ caused by both model error and disturbances. If the model error is much smaller than disturbances, $\hat{\mathbf{F}}_x$ is accurate enough to be a contact force estimator. Here in our experiments, we employ this estimation as haptic feedback for teleoperation, and thus do not require a force/torque sensor at the point of contact.

Note that the torso motion error doesn't affect the foot position error because the torso is controlled in the null-space of the manipulation foot. If the base were fixed, the same external force, $\mathbf{F}_x$, acting on the foot would result in the same motion error, $\mathbf{e}_s$, as when running our hierarchical whole-body controller with a floating base. The difference only exists in joint errors. We can always only measure the motion of the foot and then use Eq. (13) to estimate external force $\mathbf{F}_x$ without any torso motion error interfering.

$$\hat{\mathbf{F}}_x = \boldsymbol{\Lambda}_{c,s}\ddot{\mathbf{e}}_s + \mathbf{D}_{d,s}\dot{\mathbf{e}}_s + \mathbf{K}_{d,s}\mathbf{e}_s \quad (13)$$

B. QP optimization

The control structure is depicted in Fig. 1 where the inputs of QP optimization in constraint space are torques for the
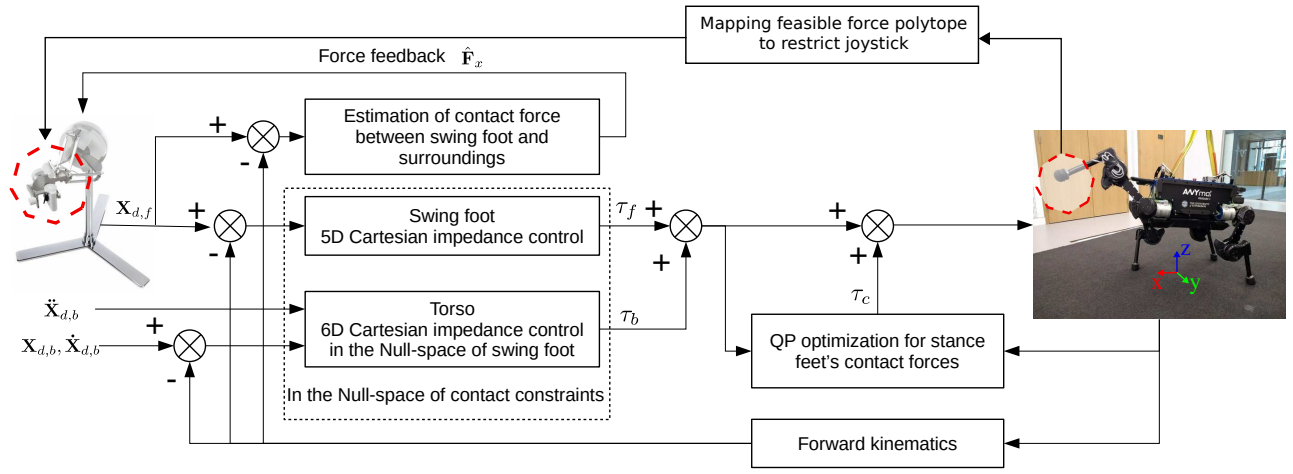
Fig. 1. Schematic of foot posture controller with haptic teleoperation.

motion and state feedback, which implies the motion can affect contact forces $\boldsymbol{\lambda}_c$ as $\ddot{\mathbf{q}}$ are involved in Eq. (8). We solve the forward dynamics of Eq. (7) to derive the equation of $\ddot{\mathbf{q}}$ with respect to $\boldsymbol{\tau}$, and then substitute it into Eq. (8) to eliminate $\ddot{\mathbf{q}}$. Since $\mathbf{PM}$ is not invertible, we resort to a trick (provided in [21][12]) of using $\mathbf{M}_c = \mathbf{PM} + \mathbf{I} - \mathbf{P}$ to replace $\mathbf{PM}$ in Eq. (7) as $(\mathbf{I} - \mathbf{P})\ddot{\mathbf{q}} = \dot{\mathbf{P}}\dot{\mathbf{q}}$ holds, leading to

$$\ddot{\mathbf{q}} = \mathbf{M}_c^{-1}(\boldsymbol{\tau}_m - \mathbf{Ph} + \dot{\mathbf{P}}\dot{\mathbf{q}}) \tag{14}$$

where $\boldsymbol{\tau}_m$ is the result from Cartesian impedance control in our case. Subsequently, substituting (14) into Eq. (8) yields

$$\boldsymbol{\lambda}_c = (\mathbf{J}_c)^+ \left[(\mathbf{I} - \mathbf{P})[\mathbf{MM}_c^{-1}(\boldsymbol{\tau}_m - \mathbf{Ph} + \dot{\mathbf{P}}\dot{\mathbf{q}}) + \mathbf{h}] - \boldsymbol{\tau}_c \right] \tag{15}$$

Therefore, the QP in constraint space with respect to $\boldsymbol{\tau}_c$ can be defined as

$$
\begin{aligned}
\underset{\boldsymbol{\tau}_c}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{\tau}_c\|_2^2 \\
\text{subject to} \quad & \boldsymbol{\tau}_{\min} - \boldsymbol{\tau}_m \leq \boldsymbol{\tau}_c \leq \boldsymbol{\tau}_{\max} - \boldsymbol{\tau}_m \\
& \mu(\boldsymbol{\eta} - \mathbf{J}_c^+ \boldsymbol{\tau}_c)_{z,i} \geq \sqrt{\lambda_{x,i}^2 + \lambda_{y,i}^2} \\
& (\boldsymbol{\eta} - \mathbf{J}_c^+ \boldsymbol{\tau}_c)_{z,i} \geq 0
\end{aligned}
\tag{16}
$$

where $\boldsymbol{\eta} = (\mathbf{J}_c)^+ (\mathbf{I} - \mathbf{P})[\mathbf{MM}_c^{-1}(\boldsymbol{\tau}_m - \mathbf{Ph} + \dot{\mathbf{P}}\dot{\mathbf{q}}) + \mathbf{h}]$ abbreviates the first right side term of Eq. (15). $\boldsymbol{\tau}_m$ in Eq. (16) is regarded as a constant vector. The QP may have no solution if the desired trajectory is not physically feasible, which can be solved by using a motion planner subject to physical constraints. In the case of teleoperation, this motion is generated by the operator. Particularly when the robot has interaction during a manipulation mission, the operator may push the joystick too much resulting in slipping. The next section will discuss how to set boundaries on the joystick in order to avoid the operator breaking physical constraints.

III. FEASIBILITY BOUNDARIES FOR TELEOPERATION

An user may operate the joystick to execute commands that are physically incapable for the robots. Instead of simply ignoring such infeasible commands, we propose to set

motion boundaries on the joystick in order to give operators early notifications of approaching dynamically infeasible areas.

A. Boundaries on CoM position

To limit CoM motion, we look for the maximum range in each direction from the current foot position that will keep the CoM within the support polygon. Theoretically, we can use 6 DoFs to control foot posture meanwhile using the remaining 3 DoFs to control the torso. The null-space projector will automatically use the 3 DoFs for the torso to guarantee the foot trajectory tracking if the 6 DoFs assigned to foot are not enough to achieve desired motion. Solving forward dynamics and simulating the system for a time horizon can give us predicted robot state. If we want to calculate the boundary in Cartesian space, that will be time consuming to predict. Instead of directly finding the boundaries of the foot, we shrink the supporting polygon to be the boundary of CoM. When the CoM is close to the boundary, joystick stiffness will increase to stop joystick moving along that direction.

B. Boundaries for feasible acting forces

Since the joystick sends position commands to the impedance controller, the impedance controller generates acting forces based on the position errors. We can restrict the workspace of joystick to saturate the acting force. Firstly, we need to compute the set of forces at the end-effector respecting joint torque limits and the contact force constraints. Such a set has been used to enhance motion planning of robot manipulators [19]. The operational-space impedance controller gains from (11) are used to transform the set of feasible end-effector forces to feasible displacements, and are subsequently mapped to feasible joystick commands. The joystick will run a varying impedance controller to sharply increase the stiffness when the operator is moving the joystick into a infeasible position.

To compute the set of feasible forces at the end-effector, we extend the computational framework from [20], which

1433

computes the set of feasible wrenches applied at the torso of the robot. As in that framework, we note the structure of the equations of motion (1):

$$
\begin{bmatrix} \mathbf{d}_b \\ \mathbf{d}_0 \\ \mathbf{d}_1 \\ \mathbf{d}_2 \\ \mathbf{d}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{I} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{I} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} 0 \\ \boldsymbol{\tau}_0 \\ \boldsymbol{\tau}_1 \\ \boldsymbol{\tau}_2 \\ \boldsymbol{\tau}_3 \end{bmatrix} + \begin{bmatrix} \mathbf{J}_{0,b}^\top & \mathbf{J}_{1,b}^\top & \mathbf{J}_{2,b}^\top & \mathbf{J}_{3,b}^\top \\ \mathbf{J}_{0,0}^\top & 0 & 0 & 0 \\ 0 & \mathbf{J}_{1,1}^\top & 0 & 0 \\ 0 & 0 & \mathbf{J}_{2,2}^\top & 0 \\ 0 & 0 & 0 & \mathbf{J}_{3,3}^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_0 \\ \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \\ \boldsymbol{\lambda}_3 \end{bmatrix}
$$
(17)

where we have split the inertial forces ($\mathbf{d} = \mathbf{M}\ddot{\mathbf{q}} + \mathbf{h}$), the joint-torques, and the contact forces into parts associated with the base and the separate legs and feet. Leg number 0 performs the manipulation, meaning the associated λ_0 will be 0.

The bottom four rows provide a constraint between joint torques and contact forces for each leg:

$$
\boldsymbol{\lambda}_i = \mathbf{J}_{i,i}^{-\top}(\mathbf{d}_i - \boldsymbol{\tau}_i), \quad i = 0, 1, 2, 3
$$
(18)

These equality constraints map the hypercube of allowed joint torques to a polytope of contact forces. For the stance-legs these polytopes are intersected with the friction cone. Any constraints on the contact force in manipulation, for instance when interacting with fragile environments, can be added at this stage as well. This results in a polytope (FCP_i) of feasible contact forces for each separate leg.

Finally the top row of (17) combines these polytopes with the dynamics of the robot torso, to form the feasible force polytope FFP for the manipulating leg:

$$
FFP = \{\mathbf{f} \in FCP_0 \mid \mathbf{J}_{0,b}^\top \mathbf{f} \in (\mathbf{d}_b - \sum_{i=1}^{3} \mathbf{J}_{i,b}^\top FCP_i)\}
$$
(19)

where a sum of polytopes is taken to be their Minkowski sum. The polytope-operations are handled using the CD-DLib[1] library. Unfortunately the various transformations require multiple conversions between vertex and halfspace representations, making the computation of the feasible force polytope expensive. As a result the polytopes are computed offline with a simple look-up used for online access.

Figure 2 shows the feasible force polytope for a configuration in simulation. In this configuration the robot can exert little force upwards, as larger forces would shift the centre of pressure beyond the support polygon. The posture of each surface of the polytope in world frame is defined by the normal vector and the maximum feasible force along this direction. In theory we need to project the contact force on all these directions (37 for this configuration) and then compare the projection with the maximum feasible force along these directions. If the projected force is greater than the maximum feasible force, that means the contact force will break physical constraints. In practice we will shrink the FFP corresponding to a certain configuration to be the boundary on the joystick. When any direction is close to the boundary we will generate a force to prevent the joystick from entering the area and to inform the operator.

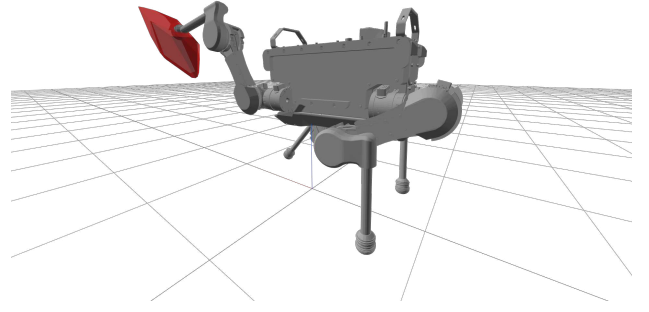[1]See https://github.com/JuliaPolyhedra/CDDLib.jl

Fig. 2. The feasible force polytope during the experiment. Shown in red are the forces that can be applied to the environment. The force polytope is shown with the origin shifted to the centre of the foot and using a scaling of $500\,\mathrm{N\,m^{-1}}$.

IV. EXPERIMENTS

The proposed algorithms are implemented on the quadruped robot ANYmal[2] and a Sigma.7[3] haptic joystick. In practical manipulation cases the yaw of the foot with respect to foot frame (z axis of foot frame coincides with the central axis of shank) has little to no effect on the task, so we chose not to control it. The remaining 5 dimensions are controlled by the foot impedance controller as shown in Fig. 1. Not controlling the yaw helps in giving the torso controller more authority in the null-space, otherwise the torso is quite sensitive to disturbance and non-smooth motion commands. The experiment videos can be found at: https://www.youtube.com/watch?v=htI8202vfec.

A. Haptic joystick setup

The Sigma.7 haptic device used in Fig. 1 is connected to the quadruped using a WiFi connection and custom ROS node running on a different computer than the ANYmal with Ubuntu 18.04 LTS. The sensing and control cycle of Sigma.7 is 100 Hz, meanwhile the control loop of locomotion controller on ANYmal's PC is 400 Hz. The commercial version from Force Dimension comes with a SDK that allows to easily access the device pose, twist and wrench of the device end-effector while also applying a desired wrench. For this work the device pose is read and added to the current quadruped end-effector pose to generate the target pose. As the workspace of the haptic device is significantly smaller than the ANYmal one, the Sigma.7 gripper is used to reset the end-effector command. To reduce the mental load on the operator and increase the stability of the quadruped the authors implemented the following measures:

- The haptic device stiffens when the quadruped is reaching feasible limits as described in section III.
- As the quadruped might experience high contact forces the reported contact forces are only applied when the operator presses the device gripper. In case of an open gripper state the Sigma.7 device goes into it's initial base pose in the center of the workspace.

[2]See http://www.anybotics.com/
[3]See http://www.forcedimension.com/

1434

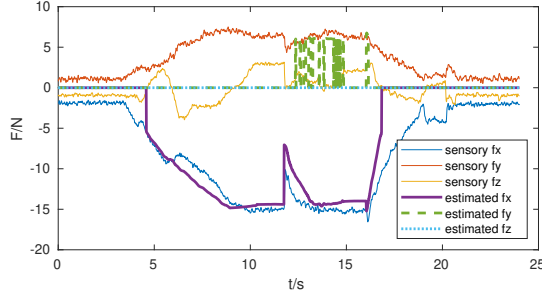Fig. 3. Teleoperating ANYmal to press the e-stop button of a crane.

Fig. 4. Force comparison between sensory results and estimated results during pressing button experiment. The button was pressed at around 5 s. The contact forces suddenly changed at around 12 s because the button started to move. After pressing down, contact forces increased again. The operator released the button at around 16 s. Compared to noisy sensory forces, the estimated forces are better to be used as haptic feedback.

B. Accuracy verification of estimated contact force

The Cartesian impedance controller can estimate the contact force as well as the contact wrench. Usually we do not need wrench feedback since the foot shape is a small sphere, designed as a point foot. The first experiment is to verify how accurate the force estimation based on impedance controller is. We teleoperated the robot to press an e-stop button of a crane as shown in 3. The force sensor inside the foot is enabled to record sensory contact force to compare with estimated force by impedance controller. The Eq. (13) is used to compute estimated forces. However, acceleration and velocity are noisy. If the estimated force contains high frequency components it will cause oscillation on joystick when rendering the feedback forces. In the end, we used only the position error to estimate the contact forces.

The recorded data is shown in Fig. 4. It should be noted that we set a threshold of 5 N to filter the estimated force, intending to discriminate non-contact error when swinging in the air. This threshold can explain the spikes of estimated f_y, and also leads to $f_z = 0$ because estimated f_z is always less than 5 N in this experiment. The force dropping at around the 12 s mark, is due to the button being fully pressed. When the button reaches the bottom, the contact force increases until the operator feels that the button was completely pressed down.

Based on Fig. 4 the estimated force is accurate enough to be used as haptic feedback for teleoperation. Additionally compared to noisy sensory forces (the sensory result shown in Fig. 4 has already been filtered by a second order low pass filter) the estimated force is easier to be projected onto the joystick.
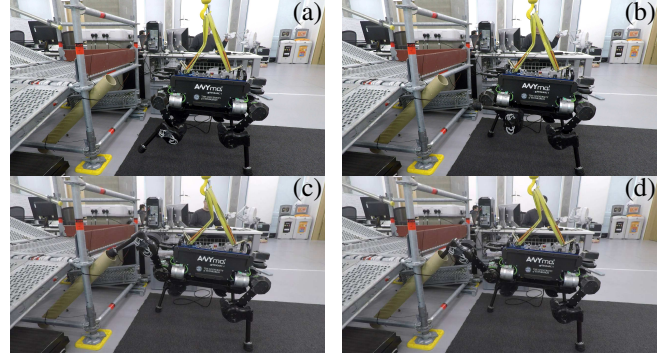

Fig. 5. Teleoperating the foot into a pipe

Fig. 6. Teleoperating the foot to reach the bottom covered by plastic peanuts and sliding on the surface to establish the depth of the reliable bottom surface.

C. Pipe manipulation

In this experiment we show-case the robot inserting its foot into a pipe, via teleoperation. This scenario is relevant for industrial use and inspection tasks. The aim of this experiment is to show the whole-body controller can control foot posture, since inserting the foot into the pipe demands accurately adjusting the foot orientation to be aligned with the pipe. Fig. 5 shows the snapshots of the process. We can see that the whole shank is finally inside the pipe. Moreover the entrance of the pipe is quite high for ANYmal. We rotated the second joint more than $\pi/2$ to reach the pipe. During that we moved through a singular configuration as shown in the second snapshot. The leg moved fast when trying to overcome singularity (as can be seen in the accompanying video), but this did not cause any issue on the robot, which shows the robustness of the our controller.

D. Exploring surface

In real world deployment, visual perception of solid surfaces may be obscured by liquid, smoke, grass etc. The robot's locomotion planner needs to know the accurate elevation map to plan foot placement. In this experiment the operator teleoperates the robot to explore the surface covered by some light packing peanuts. We place the robot on a 20 cm high stage as shown in Fig. 6, such that the robot needs to stretch its leg to reach the bottom. Benefiting from the hierarchical whole-body controller, the robot can automatically lower its torso to extend the reachability of the leg. The operator then explores the surface by sliding the foot back and forth. If we recorded the foot trajectory during sliding on the surface, we can establish the geometrical plane to update the elevation map.
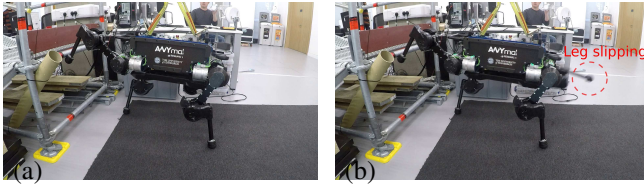
Fig. 7. Pushing against scaffold. (a) Starting to push, (b) Right hind leg slipping due to too much pushing force.
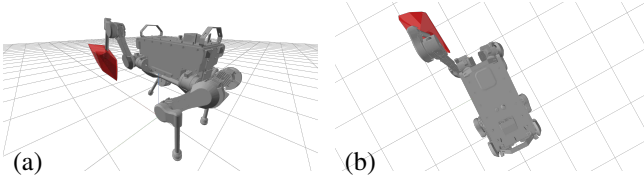

Fig. 8. *FFP* of the configuration in the pushing experiment. (a) Side view, (b) Top view.

E. Feasible force polytope boundary

In this experiment w e verify that setting a boundary on the joystick to restrict acting force is necessary. The robot was operated to push against a scaffolding structure as shown in Fig. 7. The *FFP* is calculated based on the measured configuration when the robot is pushing as shown in Fig. 8. The polytope for this configuration has 33 surfaces. The smallest infeasible force in this configuration is $25.6\,\mathrm{N}$ with direction $\begin{bmatrix} 0.08 & -0.99 & -0.11 \end{bmatrix}^{\top}$. To verify the correctness of *FFP*, the operator kept pushing upward and forward in order to trigger the *FFP* limitation without a boundary being set on joystick. The recorded contact forces are shown in Fig. 9. The right hind leg slipped quickly at around $25\,\mathrm{s}$ when the recorded forces jump up according to Fig. 9. The recorded data after $25\,\mathrm{s}$ is irrelevant because the system crashed after detecting large slipping. We use the recorded forces to compute the projection along each direction of *FFP*. This shows that the projection of the contact force along with direction $\begin{bmatrix} -0.39 & -0.04 & 0.92 \end{bmatrix}^{\top}$ is 30.69 at $23.5\,\mathrm{s}$. That is slightly greater than the model-predicted maximum feasible value 30.68 in that direction, implying that the computed *FFP* is close to the real limit considering model error and force estimation error. Subsequently, we repeated that experiment with enabled *FFP* boundary on joystick. We shrunk the boundary to 80% for safety. The oscillation of contact force shown in Fig. 10 happened when the boundary was triggered. The increased stiffness of joystick prevented the operator from sending infeasible commands.

V. CONCLUSIONS

A hierarchical whole-body controller is successfully applied to foot posture teleoperation of a quadruped robot ANYmal. The output torques of the whole-body controller consists of two orthogonal components: 1) motion torques which are dedicated to execute mission commands without considering physical constraints, and 2) constraint torques that are only for satisfying physical constraints without generating any movement. The motion torques are computed
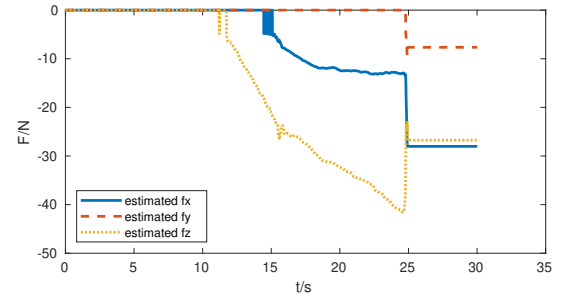

Fig. 9. Estimated contact forces during pushing against a scaffold structure. The right hind foot started to slip at around $25\,\mathrm{s}$ because the contact forces was greater than the limit.
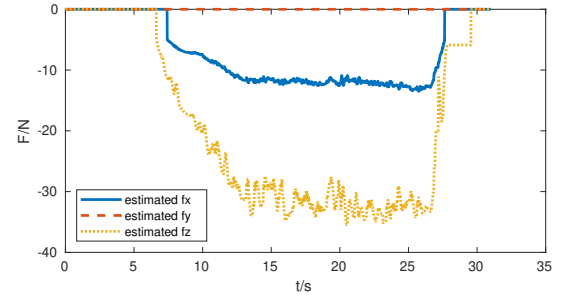

Fig. 10. Estimated contact forces during pushing against a scaffold structure with *FFP* boundary running. The oscillation caused by predefined pushing back forces around the boundary implies the boundary is triggered.

by two analytical Cartesian impedance controllers decoupled by a dynamically consistent null-space projector. Motion control is performed by an analytical controller, such that only one QP optimization needs to be solved to cope with the inequality constraints. Two impedance controllers for the foot posture control and the torso control potentially overlap, but the null-space projector imposes a strict priority, which benefits the foot reachability. The foot impedance controller is also used to estimate contact forces when the manipulation foot is in contact with the environment. Experimental results show the estimation accuracy of this method for haptic teleoperation. In order to avoid falling over due to infeasible commands sent by operator, the CoM and the acting forces are effectively bounded by limiting the joystick movement. These boundaries not only protect the robot, they also guide the operator to move the joystick in feasible directions. Future work will aim to provide predictive online computing algorithms for the CoM boundary.

REFERENCES

[1] O. C. Jenkins, R. Peters, and R. Bodenheimer, "Uncovering success in manipulation," in *Robotics: Science and Systems Workshop on Manipulation in Human Environments, Philadelphia, PA, USA*, 2006.

[2] C. G. Atkeson, B. P. W. Babu, N. Banerjee, D. Berenson, C. P. Bove, X. Cui, M. DeDonato, R. Du, S. Feng, P. Franklin *et al.*, "No falls, no resets: Reliable humanoid behavior in the darpa robotics challenge," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 623–630.

[3] B. U. Rehman, M. Focchi, J. Lee, H. Dallali, D. G. Caldwell, and C. Semini, "Towards a multi-legged mobile manipulator," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 3618–3624.

[4] B. Dynamics. Introducing spotmini. Youtube. [Online]. Available: https://www.youtube.com/watch?v=tf7IEVTDjng&t=59s

[5] C. D. Bellicoso, K. Kramer, M. Stäuble, D. Sako, F. Jenelten, F. Bjelonic, and M. Hutter, "Alma-articulated locomotion and manipulation for a torque-controllable robot," in *International Conference on Robotics and Automation (ICRA 2019)*, 2019.

[6] M. De Lasa, I. Mordatch, and A. Hertzmann, "Feature-based locomotion controllers," in *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4. ACM, 2010, p. 131.

[7] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014.

[8] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Autonomous Robots*, vol. 40, no. 3, pp. 473–491, 2016.

[9] C. Dario Bellicoso, C. Gehring, J. Hwangbo, P. Fankhauser, and M. Hutter, "Perception-less terrain adaptation through whole body control and hierarchical optimization," *IEEE-RAS International Conference on Humanoid Robots*, pp. 558–564, 2016.

[10] M. De Lasa and A. Hertzmann, "Prioritized optimization for task-space control," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 5755–5762.

[11] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson, "Optimization based full body control for the atlas robot," *IEEE-RAS International Conference on Humanoid Robots*, vol. 2015-Febru, pp. 120–127, 2015.

[12] F. Aghili, "A unified approach for inverse and direct dynamics of constrained multibody systems based on linear projection operator: Applications to control and simulation," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 834–849, 2005.

[13] F. Aghili and C.-Y. Su, "Control of constrained robots subject to unilateral contacts and friction cone constraints," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 2347–2352.

[14] H.-C. Lin, J. Smith, K. K. Babarahmati, N. Dehio, and M. Mistry, "A projected inverse dynamics approach for multi-arm cartesian impedance control," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–5.

[15] G. Xin, H.-C. Lin, J. Smith, O. Cebe, and M. Mistry, "A Model-based Hierarchical Controller for Legged Systems subject to External Disturbances," *IEEE/RSJ Int. Conf. on Robotics and Automation*, pp. 4375–4382, 2018.

[16] F. Angelini, G. Xin, W. J. Wolfslag, C. Tiseo, M. Mistry, M. Garabini, A. Bicchi, and S. Vijayakumar, "Online Optimal Impedance Planning for Legged Robots," *IEEE International Conference on Intelligent Robots and Systems*, 2019.

[17] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa, "Dynamics and balance of a humanoid robot during manipulation tasks," *IEEE Transactions on Robotics*, vol. 22, no. 3, pp. 568–575, 2006.

[18] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 295–302.

[19] H. Ferrolho, W. Merkt, C. Tiseo, and S. Vijayakumar, "Comparing metrics for robustness against external perturbations in dynamic trajectory optimization," *arXiv preprint arXiv:1908.05380*, 2019.

[20] R. Orsolino, M. Focchi, C. Mastalli, H. Dai, D. G. Caldwell, and C. Semini, "Application of wrench-based feasibility analysis to the online trajectory optimization of legged robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3363–3370, 2018.

[21] M. Mistry and L. Righetti, "Operational space control of constrained and underactuated systems," *Robotics: Science and systems VII*, pp. 225–232, 2012.