Optimisation of Body-ground Contact for Augmenting the Whole-Body Loco-manipulation of Quadruped Robots

Wouter J. Wolfslag, Christopher McGreavy, Guiyang Xin, Carlo Tiseo, Sethu Vijayakumar and Zhibin Li

***Abstract*— Legged robots have great potential to perform complex loco-manipulation tasks, yet it is challenging to keep the robot balanced while it interacts with the environment. In this paper we investigated the use of additional contact points for maximising the robustness of loco-manipulation motions. Specifically, body-ground contact was studied for its ability to enhance robustness and manipulation capabilities of quadrupedal robots. We proposed equipping the robot with prongs: small legs rigidly attached to the body which create body-ground contact at controllable point-contacts. The effect of these prongs on robustness was quantified by computing the Smallest Unrejectable Force (SUF), a measure of robustness related to Feasible Wrench Polytopes. We applied the SUF to evaluate the robustness of the system, and proposed an effective approximation of the SUF that can be computed at near-real-time speed. We developed a hierarchical quadratic programming based whole-body controller that can control stable interaction when the prongs are in contact with the ground. This novel prong concept and complementary control framework were implemented on hardware to validate their effectiveness by showing increased robustness and newly enabled loco-manipulation tasks, such as obstacle clearance and manipulation of a large object.**

I. Introduction

Combined locomotion and manipulation tasks are a key competence for legged robots in applications such as warehousing, search and rescue, and offshore inspection and maintenance. To manipulate objects, a robot must exert forces onto the environment. To locomote, the robot must remain balanced and stable under the load of the manipulation. The main challenge of loco-manipulation is performing these tasks simultaneously by managing the limited resources required to complete them: motor torques and tangential contact forces [1], [2]. Better management of these resources will improve the robot's workspace, payload, robustness and stability. This paper investigates how to improve that management by adding contact points to a quadruped robot.

Previous work has shown that extra contact points reduce resource consumption and improve the stability. Examples are found in humans or humanoid robots using their arms for balance and in multi-finger and arm manipulation, [3], and [4]–[6], and [7] respectively. Accurate force control at multiple contacts increases robustness of quadrupedal locomotion, especially during perturbations [8] and on various soft and slippery ground conditions [9].

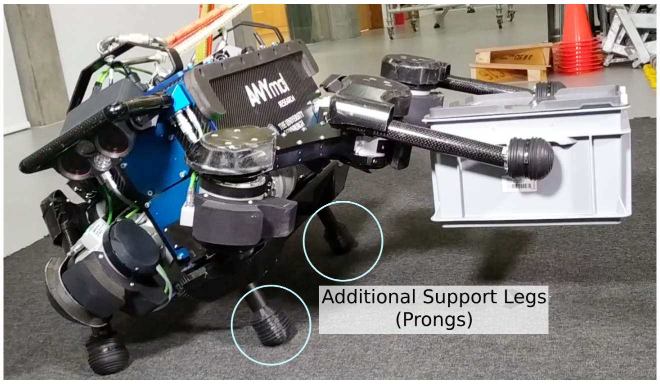

Fig. 1: The body-ground contact for enabling diverse loco-manipulation tasks: manoeuvring objects by legs.

Additional environmental contacts, however, also produce challenges in control, due to the uncertainty in estimating exact contact locations, and dealing with non-trivial surface geometries of the contacting body. Complex contacts do not fit well into multi-contact frameworks, which rely on simple contact geometry, and often on contacts only occurring at the end of the kinematic chain. This can limit the versatility of using extra contact points such as knee-ground contact [10], sliding [11] or rolling interactions in humanoid robots [12]. Recent machine learning approaches address more complex contact scenarios, such as in hand-manipulation and Jenga [13], [14] but also have limited versatility due to challenges of learning.

In contrast, humans and animals use various parts of their bodies to increase movement stability. Next to enhancing robustness, body-ground contact provides a manipulation benefit: by resting on the body, supporting limbs can be freed for performing manipulation tasks. We are motivated to investigate how quadrupedal robot movement might benefit from additional non-conventional body-ground contacts, which will be evaluated in this paper.

To enable versatile body-ground contact, we equip a quadruped robot with additional fixed limbs (see Figure 1), which we call *prongs*. These prongs are rigidly attached to the base of the robot and ensure point-contact at a known location. Such contact fits into the whole-body force control pipeline shown to be versatile in other multi-contact scenarios [15]. Rigidly connecting the prongs to the robot's torso means they will reduce actuator loads by supporting the robot's weight when they are in contact with the ground, thereby allowing the robot to perform additional tasks. Note the contrast between prong-ground and belly-ground contact: by using prongs, we know the exact contact locations, which

This work was supported by EPSRC UK RAI Hub for Offshore Robotics for Certification of Assets (ORCA, grant reference EP/R026173/1), the EPSRC CDT in Robotics and Autonomous Systems (EP/L016834/1) and EPSRC UK RAI Hub NCNR (EPR02572X/1)

Authors are with the School of Informatics, University of Edinburgh, UK.

would be difficult to estimate when using the belly, especially on non-level ground. A second key issue is manipulability: the height of the prongs allows the body to be mobile while maintaining contact with the ground, which would be more difficult with belly-ground contact.

Prong-like concepts are seen in wheeled platforms. Using outriggers, a wheeled robot can resist more disturbance force [16] [17], with an estimate of the benefits found in [18]. Legged robots can be augmented with wheels or skates at the feet to speed up locomotion in easy terrain [19], [20], or with a tail to counteract inertial shifts during fast locomotion [21]. Augmentations proposed in this paper can be used in parallel with those mentioned above.

While the prongs provide controllable ground contact, there are three related open questions: how to design prongs so they provide maximal benefit, how to deal with the control challenges posed by body-ground even with the simplified point contacts, and how to plan motions while deciding if and how to make contact with the prongs. This paper focuses on the proof-of-concept and the systematic analysis of enabling body-ground contact, therefore aspects such as planning or further mechanical enhancement such as retractable prongs are not within the scope.

We first deal with prong design, which must consider placement, ground-clearance (length), manipulability and disturbance rejection capabilities. We define the Smallest Unrejectable Force (SUF) as a metric to quantify disturbance rejection ability in loco-manipulation tasks within actuation limits and interaction constraints, and provide a fast-to-compute approximation, which are used optimise the design of the prongs. Our approach extends previous work on similar metrics (e.g., [22], [23]) and works directly showing the importance of optimising posture for robustness (e.g., [2], [24]).

To control the robot with prongs, we use the established framework for modern quadruped robots: Quadratic Programming (QP) based inverse dynamic controllers [8], [15], [25], [26]. However, these controllers were designed for contacts at the ends of the kinematic chain, not at the torso. The control of non-end-effector limb contact has been studied in manipulation [24] considering contacts with moving obstacles which do not kinematically constrain the contact limbs. When using prongs, the torso will be constrained, and the above controllers might become unstable due to rank deficiency of the constraint Jacobian. We propose a hierarchical QP controller that uses a reorganised QP hierarchy and a singular-value-decomposition-based pseudo-inverse of the constraint Jacobian to handle provide stable and numerically reliable control.

A. Contributions

Our paper studies the design of prongs for body-ground contact in quadrupedal robots. We validate their performance in three hardware experiments: push-rejection, obstacle clearance and object manipulation. The last two experiments use two conventional legs freed for manipulations by the support of the prongs. This provides the following contributions:

1) A proof-of-concept prong design for the ANYmal robot which enables effective body-ground contact (Section III).
2) A novel method to quickly compute an approximation of the Smallest Unrejectable Force, a measure for the robustness of the robot (Section IV).
3) Metrics for benchmarking the robustness and stability of a robot with and without prongs (Section VII).
4) An hierarchical QP controller that enables the robot to be operated with prongs by including contact constraints on base movement (Section VI).

Section II explains our notation for the robot dynamics. Section III discusses the optimal prong design. Our robustness measure, the SUF, and its novel approximations are explained in Section IV. Section V shows results from simulations and optimisations. Section VI explains the controller for the hardware experiments. Hardware experiments highlighting the efficacy of the prongs are in Section VII. The discussion and conclusion are in Section VIII and IX.

II. Preliminaries and Robot Dynamics

The dynamics of a quadrupedal robot with a manipulator, and prongs attached, as shown in Fig. 2, are given by:

$$M(q)\ddot{q}+h(q,\dot{q})=d(\ddot{q},\dot{q},q)=B\tau+J_{\mathrm{f}}^{\top}\lambda_{\mathrm{f}}+J_{\mathrm{p}}^{\top}\lambda_{\mathrm{p}}+J_{\mathrm{e}}^{\top}F, \quad (1)$$

where q are the generalised coordinates of the robot describing the position and orientation of the body, and the position of each joint, $M(q)$ is a positive definite mass matrix, $h(q,\dot{q})$ is the dynamic bias containing of centrifugal, Coriolis and gravitational effects, τ are the joint torques, B is a selection matrix, J_{f}, J_{p} and J_{e} are the Jacobians of the feet, prongs and end-effector of the arm respectively, and λ_{f}, λ_{p} and F are external (reaction) forces at those points.

These equation of motions are subject to further constraints to ensure physically feasible ground interaction and joint/motor torques. Ground interaction constraints ensure the robot does not slip or penetrate the ground, and are only considered when the associated body part is in ground contact. For computational efficiency, these conditions are approximated as linear constraints for each contact point i:

$$\begin{bmatrix}0 & 0 -1\end{bmatrix}\lambda_i \leq 0 \quad (2)$$

$$\begin{bmatrix}1 & 0 & -\tfrac{1}{2}\sqrt{2}\mu\end{bmatrix}\mathrm{abs}(\lambda_i) \leq 0 \quad (3)$$

$$\begin{bmatrix}0 & 1 & -\tfrac{1}{2}\sqrt{2}\mu\end{bmatrix}\mathrm{abs}(\lambda_i) \leq 0 \quad (4)$$

$$J_i\ddot{q} + \dot{J}_i\dot{q} = 0 \quad (5)$$

where μ is a friction coefficient, and the abs-operator returns the piece-wise absolute value. Additionally, the motor capabilities are reflected in bounds on the joint torque for each joint index i and torque limit $\bar{\tau}_i$:

$$-\bar{\tau}_i \leq \tau_i \leq \bar{\tau}_i. \quad (6)$$

III. Optimal Design of the Prongs

The prongs enlarge the buffer between the motor torque limits and the torques required to stand. This buffer can then be used to reject disturbances or perform secondary tasks.

However, the magnitude of the benefits depend on how the prongs are placed and sized. To investigate the effects of the prong design we focus on a scenario in which a force is applied at the end-effector of an arm attached to the torso of a robot (Fig. 2). We find the effect of prong placement on the size of the disturbance the end-effector can sustain without moving the robot. We then optimise the prong placement and the robot configuration for this robustness measure.

By using a two prong configuration, the robot can either fix torso pitch (by grounding both prongs) or allow pitching (by grounding one prong) depending on task requirements. For simplicity, we enforce symmetry by placing two prongs of equal length on the x-axis of the robot frame. Furthermore, we enforce a symmetric position of the feet. In this configuration, maximum robustness is achieved when the prongs are furthest apart, so prongs are placed as far apart as possible without interfering with the leg motion. Alternate placements might be beneficial for tasks with different loading conditions, and can be optimised using the methods presented. For the current configuration, the optimisation only requires three parameters:

$$\max_{\{x_f, y_f, b_z\}} F_{\text{SUF}}(x_f, y_f, b_z), \tag{7}$$

where the SUF is a measure for the robustness (defined in the next section) and x_f, y_f and b_z are the feet x and y position and torso z position. These decision variables are shown in Fig. 2. When optimising prong position, the prong length matches the height of the torso. To solve the inverse kinematics, we use a standard iterative procedure with the transpose Jacobian, which requires no further regularisation.

IV. Smallest Unrejectable Force

A key element of the robustness of a robot is the amount of external force it can withstand whilst tracking a target motion. Computing such forces and associated robustness metrics can be done via Feasible Wrench Polytopes, as discussed for legged robots in [22], and for manipulation in [27]. The FWP is the set of wrenches applied to the robot, such that the ground reaction forces and joint torques required to execute the desired motion stay within the friction cone and motor limits respectively. Here, we are interested in a slight variation: the Rejectable Force Polytope (RFP), the set of forces that can be applied to the robot at the end-effector, such that it is able to perform the desired accelerations while satisfying the constraints in (1)-(6).

$$\mathcal{F}_{\text{RFP}}(q, \dot{q}, \ddot{q}_{\text{d}}) = \{F \in \mathcal{R}^3 | \text{(1)-(6) hold for some values of}(\lambda_f,\ \lambda_p \text{ and } \tau),\ \ddot{q} = \ddot{q}_d\}, \tag{8}$$

where $\ddot{q}_{\text{d}}$ is the desired acceleration. The RFP is a polytope, as the constraints are linear in the free variables.

In practice, it is desirable to summarise the RFP into a single robustness metric. For this we propose the Smallest Unrejectable Force: the smallest disturbance force the robot cannot withstand while performing its desired motion. This is the same as finding the the Chebyshev radius of the RFP, but with the centre of the circle fixed to the origin.

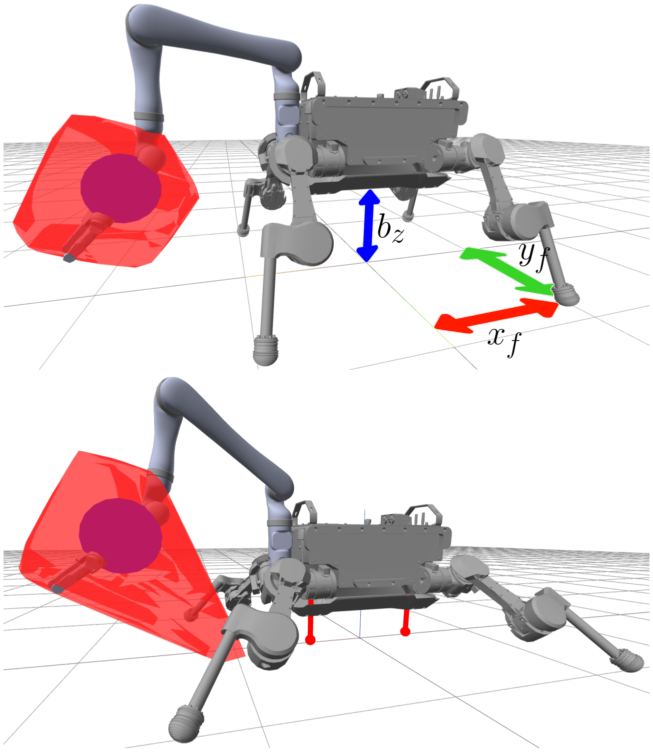

Fig. 2: Rejectable Force Polytope and maximal rejectable force for optimised robot configuration with prongs (bottom) and without prongs (top). The end-effector is set to a world frame position: {0.8m, 0.2m, 0.4m}.

The scheme from [28] can compute the exact RFP, but the computation time does not scale well with the number of contacts and joints, as it requires a transition between vertex-representation of the FWP to its halfspace-representation. The next step is to compute the SUF from the RFP, which is a one-dimensional linear optimisation problem (see (9) for details). To further simplify this step, a metric that finds the Smallest Unrejectable Force in a single predetermined direction was proposed in [22].

Here we propose and investigate three approximations of the SUF: Fibonacci, affine, and quadratic. The first approximates the RFP, the latter two directly approximate the SUF without finding the RFP.

The Fibonacci approximation is based on an inner (conservative) approximation for the RFP. Points on the boundary of the RFP are found by solving the optimisation problem:

$$\max_{f, \lambda_f, \lambda_p, \tau} f \quad \text{s.t.} \quad \text{(1)-(6)},\ F = f\hat{F}$$

where the resulting f is the maximum feasible scaling factor for force in the direction $\hat{F}$, considering the dynamic equations and leg joint torque limits $\bar{\tau}$.

The approximation of the RFP is the polytope spanned by vertices found by solving the above optimisation for a set of approximately uniformly distributed force directions (the Fibonacci-sphere [29]).

To determine the size, ρ of the Smallest Unrejectable

Force, we check for each halfspace that determines the polytope to check if the worst case force direction would violate the associated constraint at that value of ρ. Given halfspace representations $a_i F \leq b_i$, where i indexes the halfspaces of the RFP, we know the worst case force is in the direction of a_i (see [30]). Hence we solve the following optimisation problem using enumeration:

$$\max_{\rho} \boldsymbol{\rho} \qquad \text{s.t.} \quad \rho||a_i|| \leq b_i \quad \forall i = \max_{i}(b_i||a_i||^{-1}) \tag{9}$$

The Fibonacci approximation still requires a conversion from vertex to halfspace representation. The computation time depends on the number of vertices sampled. We use 1024 samples for a high quality approximation.

The affine and quadratic approximations of the SUF do not compute the RFP explicitly. Instead, they find the worst case disturbance force (similar to above), while simultaneously solving for an optimal control law determining how the joint-torques and ground reaction forces change with the disturbance force. As the true (nonlinear) optimal control law cannot be computed efficiently, the two approximations adapt the computation techniques from [31] to find optimal affine and quadratic control laws respectively. Detailed derivations of the techniques can be found in their work. Our formulations deviate slightly from those in [31] to simplify handling of the equality constraints for this specific scenario, and to search for the largest sphere centred around the origin, rather than around an arbitrary point.

First reparametrise the control equations:

$$\begin{bmatrix} F & \tau & \lambda \end{bmatrix}^\top = \begin{bmatrix} I & 0 \\ -W^+ J_e^\top & N \end{bmatrix} \begin{bmatrix} F \\ \delta Q \end{bmatrix} + \begin{bmatrix} 0 \\ W^+ d \end{bmatrix}$$

where δQ are combinations of joint-torques and ground reaction forces in the null-space of the dynamics equation, which are solved using the matrix $W = \begin{bmatrix} B & J_f^\top & J_p^\top \end{bmatrix}$, the matrix N is a basis for the nullspace of W and the Moore-Penrose pseudo-inverse is indicated by the $^+$ superscipt. The use of the pseudo-inverse in this computation requires that the desired acceleration satisfy the movement constraints and can be executed by the robot, given unlimited motor torques and contact forces.

The affine approximation optimises an affine control law from disturbance to reaction forces and torques:

$$\delta Q = \delta Q_0 + VF \tag{10}$$

where δQ_0 are nominal joint torques and ground reaction forces, and V is a gain matrix. These parameters are optimised along with the magnitude of the SUF (ρ), via the conical quadratic program:

$$\max_{\rho, \delta Q_0, V} \rho \quad \text{s.t.} \; \begin{bmatrix} 0 & \delta Q_0 \end{bmatrix} a_i + \left|\begin{bmatrix} \rho I & V^\top \end{bmatrix} a_i\right| \leq b_i \,\forall i \tag{11}$$

the constraint coefficients in a_i and b_i are taken from (3)-(6). Informally, the term inside the norm represents the effect of the worst case disturbance force on the original linear constraint. The quadratic approximation uses a quadratic control law. The resulting semi-definite program is included in the Appendix.

V. Simulations

This section first compares the proposed approximations on computational efficiency and accuracy. Then the prong optimisation problem from Section III is solved using the affine approximation of the SUF.

A. Comparing Approximation Methods

This section compares the computation time and accuracy of all computation methods: exact, single direction [22], Fibonacci, affine and quadratic. The simulations were implemented with `Julia` libraries for rigid body dynamics [32] and optimisation [33], [34]. Simulations were ran using an Intel Core i7-7830x processor and 32Gb of memory.

To compare the approximations the SUF was computed for random robot configurations from two scenarios: 1) tele-operation scenario, similar to [28], in which there is no arm attached to the robot, three legs are on the ground, and the remaining leg is used as end-effector 2) a scenario with an arm attached to the robot functioning as end-effector, and all four legs of the robot in contact with the ground (see Fig. 2).

The results, shown in Table I, affirm the slow computation of the exact method. The affine and quadratic approximations are faster than the Fibonacci approach. The quadratic approximation scales less well to the arm-attached scenario, due to the number of parameters in the quadratic term of the control law. The single direction approach is clearly fastest.

Fig. 4 conveys the quality of the approximations, by comparing the sizes of their SUFs relative to exact computation. Only the arm-attached scenario is shown to highlight the potential differences in quality, as those are larger in that scenario. The quadratic and Fibonacci approximations are very close to exact, with the affine approach resulting in slight under-approximations. The affine approximation failed to converge once due to numerical issues. Note these three approximations are conservative. The single-direction approximation is shown to have poor accuracy. Furthermore, this approach overestimates of the SUF, which is undesirable for robustness analysis. Due to the favourable trade-off between conservativity, accuracy and computation time, the affine approximation is used in the remainder of this paper.

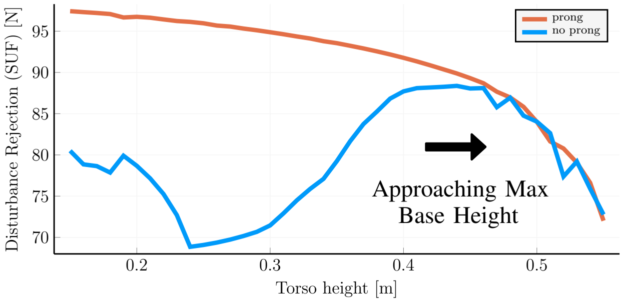

Fig. 3: The SUF for each height of the base. The foot locations are optimised for each height.

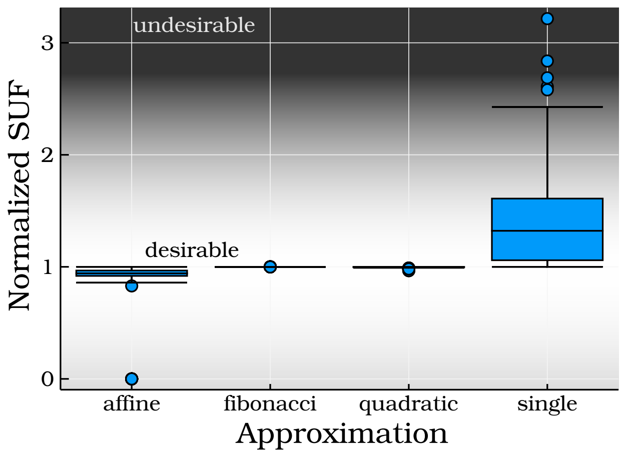

Fig. 4: Boxplot of relative SUF size in each approximation. The background color indicates the desirability. More accurate approximations are preferred, and errors that under-approximate the SUF are less undesirable, as the performance of the robot at the least matches the prediction, meaning that such errors do not lead to failures.

TABLE I: Computation time of SUF-approximations in ms

Task	Exact	Fibonacci	Affine	Quadratic	Single
teleoperation	4301	588.8	13.8	84.169	4.65
manipulation	37747	727.0	17.49	709.1	4.93

B. optimising Robust Body-Ground Contact

We optimised the prong length and placement, in order to maximise the SUF for a fixed end-effector position, see (7). As the F_{SUF} function is non-smooth, and gradients are difficult to obtain even where they exist, we use a gradient free optimisation method (COBYLA [35] in the Nlopt library [36]). A penalty method on the foot-placement error ensures the inverse-kinematics is successful at the solution. A multi-start with 20 trials is used.

The results of this optimisation are shown for scenarios with and without prong in Fig. 2. Shown are the optimal configuration of the robot, the resulting rejectable force polytope, and a sphere with the SUF as radius. The forces are scaled using a ‘stiffness’ of $1000\,\mathrm{N\,m^{-1}}$. The polytopes and spheres are translated, such that their origin (0 disturbance force) is at the end-effector. The minimal non-rejectable forces are $88\,\mathrm{N}$ and $96\,\mathrm{N}$ respectively.

To show the efficacy of prongs, we also found the SUF given a body height, optimising only the foot locations, see Fig. 3. Slight noise is caused by approximations in the IK algorithm. When the prongsare attached, the SUF is pointed upwards, and is limited by the unilaterality conditions. The prong’s length has little effect in this direction, so does not effect the SUF. Therefore, the prong length can be decided by other considerations: ground clearance and a minimum height from the base. Similar reasoning shows that adjusting the y-position of the prongs (or non-prong body-ground contact) would not benefit robustness for this scenario. The prongs also have little effect on the SUF when the torso height is larger, as the robot legs are then close to their singular position, which limits joint torques. However, as this singularity comes with mobility and control issues, such heights are undesirable. For more practical torso heights we see that the prongs provide a benefit of up to 35%.

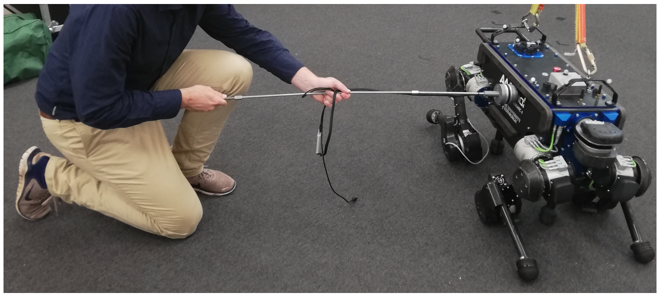

Fig. 5: The push recovery experimental setup. The robot is being pushed by a stick equipped with a force/torque sensor.

VI. Design of the QP Controller

The whole-body-control of ANYmal uses the well established hierarchical QP paradigm [37]. To ensure the robustness of this control paradigm, some additions have been proposed recently. For example, the techniques from [25] aim to improve robustness against joint tracking errors. In this paper, we follow the hierarchical QP framework by combining the foot contact constraints and the prong contact constraints into a single augmented contact Jacobian. As such, the prong and foot contacts are considered in the same way and their forces are optimised simultaneously.

At each time-step, as part of the QP, the controller minimises an error between the desired and actual task-space accelerations space: $||\ddot{x}-\ddot{x}_{\mathrm{d}}||^2_w$, with the desired accelerations based on the error e in the task space position. If for $\ddot{x} = \ddot{x}_d$ the kinematic constraints do not hold, i.e. the system is overconstrained with respect to its desired movement, minimising the acceleration error might lead to unstable behaviour. To ensure a solution that stabilises the robot, we use a whole-body controller consisting of the following five hierarchical layers, each solving a QP. Each layer ensures that the optimality conditions of the previous layers are satisfied, i.e., it optimises in the nullspace of the previous layers.

1) Dynamic feasibility: finds any feasible solution for the dynamic constraints, (1)-(6).
2) Torso angular acceleration tracking: minimises the error between the desired angular acceleration and the executed angular acceleration.
3) Torso translational acceleration tracking: when the prongs contact the ground, this layer has no effect on the outcome, as translational acceleration is not in the available nullspace. This prevents unstable behaviour.
4) Swing foot acceleration tracking
5) Torque minimisation: minimises the sum of squared motor torques, in order to reduce energy consumption.

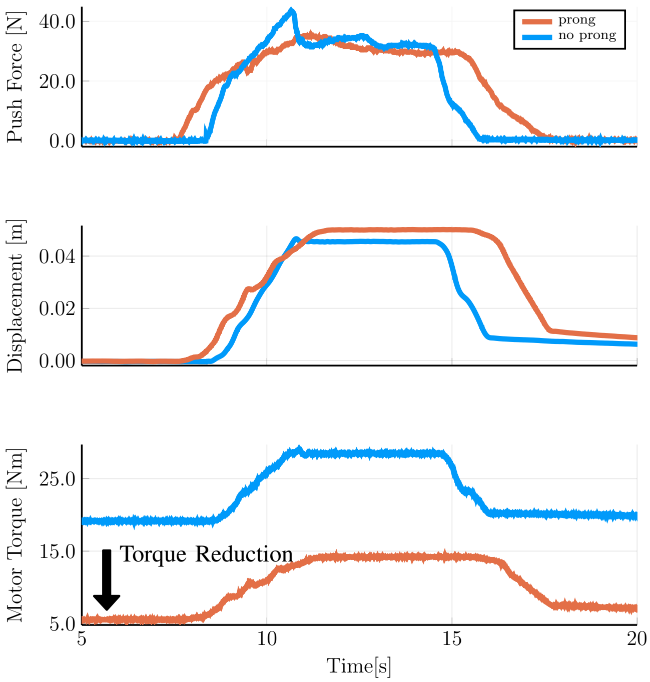

Fig. 6: The top two plots show the push force and resulting displacement during the experiment, and show that the behaviour of the robot is similar in both scenarios. The bottom plot shows a comparison of required motor torque when pushing the robot with (red) and without (blue) prongs.

The implementation also reduces computational load via the trick from [38] to avoid directly computing the torques. Due to the prongs, the first layer involves a *pseudo*-inverse of the constraint Jacobian, implemented using singular-value-decomposition. The hierarchical order ensures numerical stability of this pseudo-inverse, and ensures stable motions result in practice even if the desired accelerations cannot be tracked exactly. Note that the prongs increases the number of optimisation parameters, which potentially requires consideration on platforms with severe computational constraints.

VII. Hardware Experiments

To validate the use of prongs, we perform three experiments. 1) Pushing the robot to assess the joint-torques. 2) Clearing out an obstacle with the robots' conventional legs, freed for this task by the prongs. 3) Lifting a box with two side legs, to establish the versatility of the controller.

A. Push Recovery

To verify that our controller is able to use the prongs to enhance the robustness of the robot, we push the robot with a rod equipped with a force-torque sensor. The experiment is shown in Fig. 5. In the experiment, we push the robot horizontally on the middle of the base. The robot is in the same configuration for both experiments, which is the default standing configuration of the robot-platform with the torso height lowered to the height of the prongs. The force is gradually increased up to approximately 30 N. After holding this force for 5 seconds, the force is reduced to 0.

Fig. 6 shows the base displacement, disturbance force and motor torque during the experiment. The key result is found by comparing the effective compliance and the amplification factor between disturbance force and motor torque during the period of maximum push force. The motor torque is significantly lower with prong, despite a slightly stiffer torso behaviour. These reduced knee-joint torques result in a capacity to reject larger disturbance forces.

B. Obstacle Clearance

The second experiment shows the prongs ability to increase the robot's ability to perform basic manipulation. Standard quadrupeds would be unable to perform manipulation with more than one leg, as they are required for standing. The prongs take over responsibility for standing, freeing the legs for manipulation. Using the free-gait motion description library [39], we generate a sequence of body and end-effector targets, such that the robots pushes an obstacle away.

The resulting motion can be seen in Figs. 7 and 8. Note that, by necessity, the legs of a quadruped are equipped with relatively strong motors, which makes them well suited for obstacle clearing tasks such as this. Performing such a task is only possible when relying on body-ground contact. Enhancing the capabilities of the robot to allow obstacle clearing makes them more versatile in rough terrain.

C. Lifting A Box

Experiment three shows controlled torso mobility while the prongs contact the ground, and further manipulation with the legs. The robot lowers on its prongs, and leans to its right side, freeing the left side legs for manipulation. The legs are controlled in task-space with low-gains, allowing basic dual arm manipulation: picking up a box. Figures 9 and 10 shows snapshots of this type of experiments of lifting two different boxes. This and the previous experiment are available in the accompanying video.

When lifting the left-side legs, the desired ground-contact force changes quickly, which can result in jerky motions. To resolve such an issue, we enforced a smooth contact force in our optimisation framework.

VIII. Discussion

The improvement in robustness and manipulation capabilities introduced by the body-ground contact opens several applications, improving the intervention capabilities in robots deployed for exploration and monitoring. However, body-ground contact needs more research before quadrupeds are ready for such applications.

There are four main areas of improvement for future work. First is developing an approach to quantify and optimise the manipulability benefits the prongs. Second is extending the affine and quadratic approximations of the SUF. Knowledge about the distribution of contact forces can be used to bias the robustness measure towards more likely disturbances. With further improvements to the computational efficiency, these approximations could be incorporated in real-time planning and control. Third is incorporating body-ground contact in

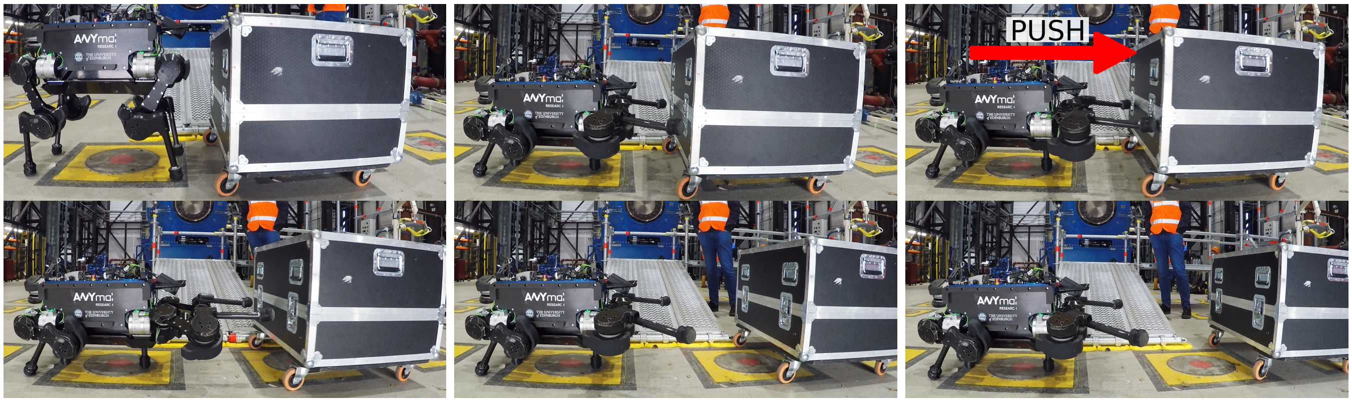

Fig. 7: The robot leaning on both prongs to push a box away. From top left: From a standing position (1), the robot lowers itself onto the prongs (2), which enables the front legs to be lifted from the ground. These front legs are used to push the box (3), which is pushed out of the way (4-6), clearing space for the robot to navigate onto the platform.

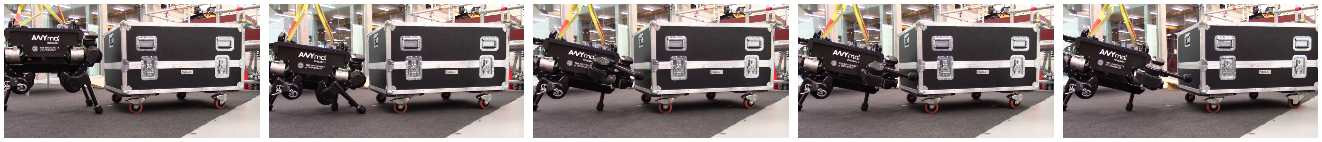

Fig. 8: The robot leaning on both prongs to push a box away. From top left: From a standing position (1), the robot lowers itself onto the prongs (2), which enables the front legs to be lifted from the ground. These front legs are used to push the box (3), which is pushed out of the way (4-5), creating a clear path.

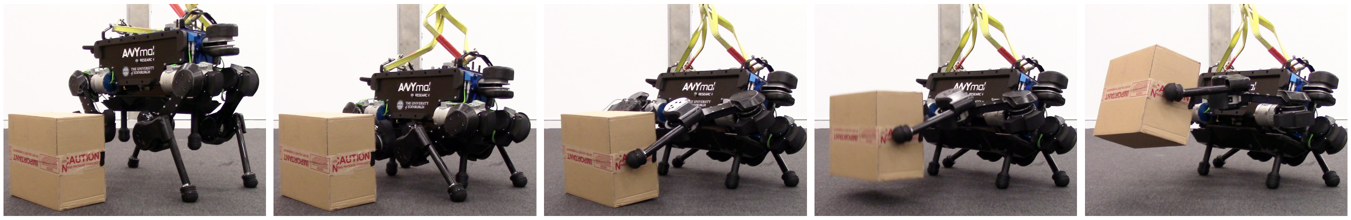

Fig. 9: Experimental snapshots of lifting a box: the support provided by the prongs is used to free two of the legs for a manipulation task.

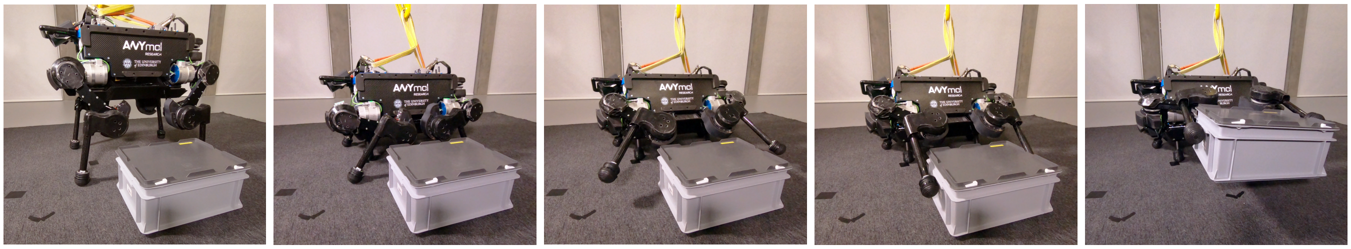

Fig. 10: Experimental snapshots of a second box lifting experiment, using a differently sized box.

motion planning. Making decisions about when to use the prongs (for robustness) and when not (for speed), maximising the utility of the prongs is a challenging problem due to the intermittent nature of the contact. Fourth is applying prongs to body-environment contact with other robot morphologies. Two examples are: a quadruped increasing its robustness by leaning sideways on a wall with its body, and a robot arm increasing its accuracy by contacting a table with a prong on its elbow.

IX. Conclusion

This paper studied the use of *prongs* to enable body-ground contact in quadrupedal robots. We showed that using prongsincreases the robustness of the robot, as measured by its ability to reject forces applied to the end-effector, by up to 35 %, largely independent on prong length. A method was developed for fast approximation of this smallest unrejectable force metric.

We applied an optimisation-based whole-body controller that handles the constrained body motion resulting from body-ground contact. On the hardware, we verified the increased robustness in the form of push resistance with limited motor torques. We also showed obstacle clearance and basic object manipulation, two capacities added by the prongs freeing the legs from their body-support task.

Appendix

To compute the SUF, it is also possible to use a quadratic inverse dynamics law. Optimising this law for maximal force rejection is a semi-definite program. The program is detailed below, by adapting from [31].

$$\max_{\rho,\delta Q_0,V,W,\zeta,\xi} \rho \quad \text{s.t.} \qquad \zeta \leq \boldsymbol{b} \tag{12}$$

$$\xi \geq 0 \tag{13}$$

$$\begin{pmatrix} \zeta_i - \xi_i - \boldsymbol{a}_i^\top \begin{pmatrix} 0 \\ \delta Q_0 \end{pmatrix} & -\frac{1}{2}\boldsymbol{a}_i^\top \begin{pmatrix} \rho I \\ V \end{pmatrix} \\ -\frac{1}{2}\begin{pmatrix} \rho I \\ V \end{pmatrix}^\top \boldsymbol{a}_i & \xi_i - \sum_j^{n_2} a_{ij+n_1} W_j \end{pmatrix} \succeq 0 \; \forall \, i \tag{14}$$

Here the quadratic term of the inverse dynamics law is W, and we introduced two intermediate variables ξ and ζ.

References

[1] E. Farnioli *et al.*, "Optimal contact force distribution for compliant humanoid robots in whole-body loco-manipulation tasks," in *Int. Conf. on Robotics and Automation*, 2015.

[2] F. Abi-Farraj *et al.*, "Torque-based balancing for a humanoid robot performing high-force interaction tasks," *Robotics and Automation Letters*, 2019.

[3] M. Kudruss *et al.*, "Optimal control for whole-body motion generation using center-of-mass dynamics for predefined multi-contact configurations," in *Int. Conf. on Humanoid Robots*, 2015.

[4] B. Henze *et al.*, "Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios," *Int. J. of Robotics Research*, 2016.

[5] J. Carpentier and N. Mansard, "Multicontact locomotion of legged robots," *T. on Robotics*, 2018.

[6] T. Koolen *et al.*, "Design of a momentum-based control framework and application to the humanoid robot atlas," *Int. J. of Humanoid Robotics*, 2016.

[7] C. Smith *et al.*, "Dual arm manipulationa survey," *Robotics and Autonomous systems*, 2012.

[8] G. Xin *et al.*, "A model-based hierarchical controller for legged systems subject to external disturbances," in *Int. Conf. on Robotics and Automation*, 2018.

[9] I. Chatzinikolaidis *et al.*, "Contact-implicit trajectory optimization using an analytically solvable contact model for locomotion on variable ground," in *IEEE Robotics and Automation Letters*, 2020.

[10] B. Henze *et al.*, "Multi-contact balancing of humanoid robots in confined spaces: Utilizing knee contacts," in *Int. Conf. on Intelligent Robots and Systems*, 2017.

[11] M. Trkov *et al.*, "Bipedal model and hybrid zero dynamics of human walking with foot slip," *J. of Computational and Nonlinear Dynamics*, 2019.

[12] A. Specian *et al.*, "Robotic edge-rolling manipulation: A grasp planning approach," *Robotics and Automation Letters*, 2018.

[13] H. Van Hoof *et al.*, "Learning robot in-hand manipulation with tactile features," in *Int. Conf. on Humanoid Robots*, 2015.

[14] N. Fazeli *et al.*, "See, feel, act: Hierarchical learning for complex manipulation skills with multisensory fusion," *Science Robotics*, 2019.

[15] M. Hutter *et al.*, "Quadrupedal locomotion using hierarchical operational space control," *Int. J. of Robotics Research*, 2014.

[16] S. Shirafuji *et al.*, "Mechanism allowing a mobile robot to apply a large force to the environment," in *Int. Conf. on Intelligent Autonomous Systems*, 2016.

[17] T. Kiyota *et al.*, "3d-free rescue robot system," in *Int. Conf. on Robotics and Automation*, 2006.

[18] S.-L. Jeng *et al.*, "Outrigger force measure for mobile crane safety based on linear programming optimization#," *Mechanics Based Design of Structures and Machines*, 2010.

[19] P. Oh *et al.*, "Technical overview of team drc-hubo@ unlv's approach to the 2015 darpa robotics challenge finals," *J. of Field Robotics*, 2017.

[20] M. Bjelonic *et al.*, "Skating with a force controlled quadrupedal robot," in *Int. Conf. on Intelligent Robots and Systems*, 2018.

[21] W. Saab *et al.*, "Robotic tails: a state-of-the-art review," *Robotica*, 2018.

[22] R. Orsolino *et al.*, "Application of wrench-based feasibility analysis to the online trajectory optimization of legged robots," *Robotics and Automation Letters*, 2018.

[23] Y. Zheng and K. Yamane, "Evaluation of grasp force efficiency considering hand configuration and using novel generalized penetration distance algorithm," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 1580–1587.

[24] B. J. Thibodeau *et al.*, "Static analysis of contact forces with a mobile manipulator," in *Int. Conf. on Robotics and Automation*, 2006.

[25] A. Del Prete and N. Mansard, "Robustness to joint-torque-tracking errors in task-space inverse dynamics," *T. on Robotics*, 2016.

[26] A. Escande *et al.*, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *Int. J. of Robotics Research*, 2014.

[27] H. Ferrolho *et al.*, "Comparing metrics for robustness against external perturbations in dynamic trajectory optimization," *arxiv*, 2019.

[28] G. Xin *et al.*, "Bounded haptic teleoperation of a quadruped robot's foot posture for sensing and manipulation," in *Int. Conf. on Robotics and Automation*, 2020.

[29] Á. González, "Measurement of areas on a sphere using fibonacci and latitude–longitude lattices," *Mathematical Geosciences*, vol. 42, no. 1, p. 49, 2010.

[30] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[31] J. Zhen and D. Den Hertog, "Computing the maximum volume inscribed ellipsoid of a polytopic projection," *INFORMS J. on Computing*, 2017.

[32] T. Koolen and R. Deits, "Julia for robotics: Simulation and real-time control in a high-level programming language," in *Int. Conf. on Robotics and Automation*, 2019.

[33] I. Dunning *et al.*, "Jump: A modeling language for mathematical optimization," *SIAM Review*, 2017.

[34] B. O'Donoghue *et al.*, "Conic optimization via operator splitting and homogeneous self-dual embedding," *J. of Optimization Theory and Applications*, June 2016.

[35] M. J. Powell, "Direct search algorithms for optimization calculations," *Acta numerica*, pp. 287–336, 1998.

[36] S. G. Johnson, "The nlopt nonlinear-optimization package," 2017. [Online]. Available: https://github.com/JuliaOpt/NLopt.jl

[37] M. De Lasa *et al.*, "Feature-based locomotion controllers," in *ACM Transactions on Graphics (TOG)*, 2010.

[38] A. Herzog *et al.*, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Autonomous Robots*, 2016.

[39] P. Fankhauser *et al.*, "Free gaitan architecture for the versatile control of legged robots," in *Int. Conf. on Humanoid Robots*, 2016.