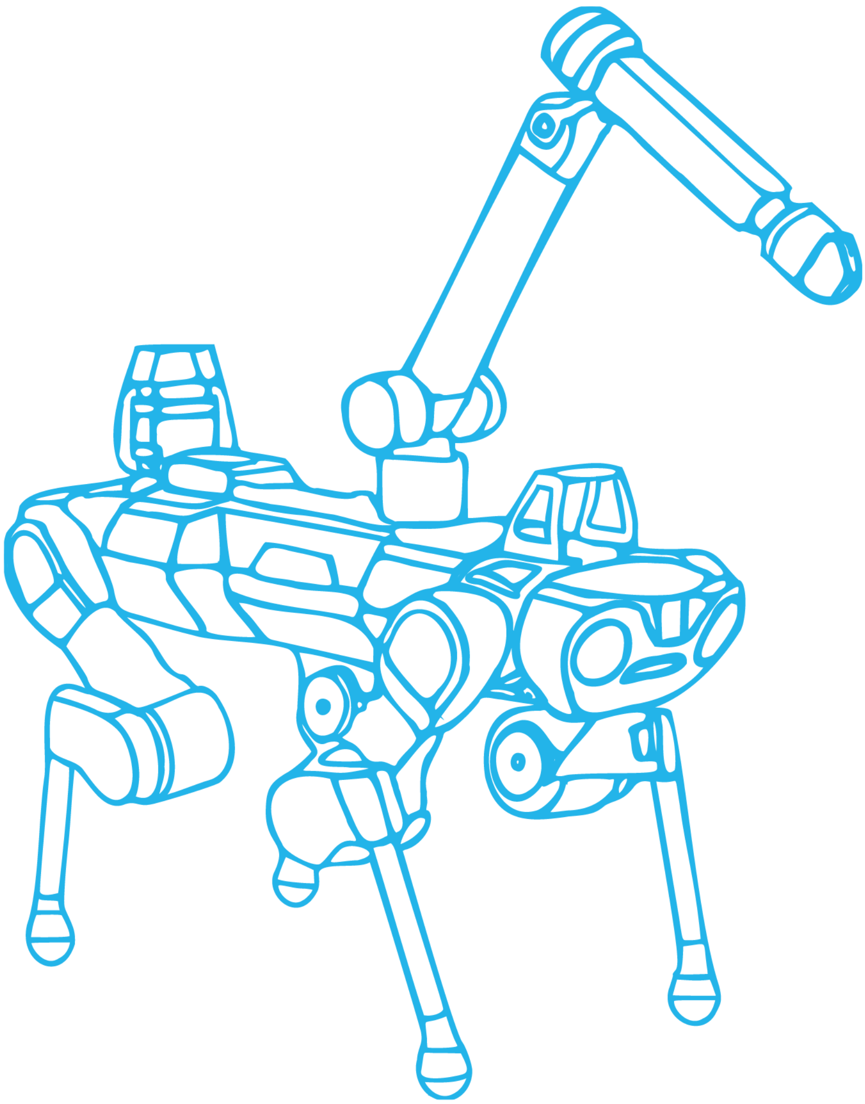
Collision detection, isolation and identification

Implemented on a four-legged robot with arm

J. van Dam

Master of Science Thesis



Delft University of Technology

Cognitive Robotics, BioMechanical Engineering and Delft Center for Systems and Control

Collision detection, isolation and identification

Implemented on a four-legged robot with arm

Master of Science Thesis

For the double degree of Master of Science in Systems and Control and BioMechanical Design at Delft University of Technology

J. van Dam

February 22, 2022

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis was done in the Robotic Systems Lab (RSL) at Eidgenössische Technische Hochschule (ETH) Zürich. Their cooperation is hereby gratefully acknowledged.

Table of Contents

List of Figures

List of Tables

Acknowledgements

First of all, I would like to thank my daily supervisors from ETH Zürich, Andreea Tulbure, Maria Vittoria Minniti, and Dr. Firas Abi Farraj, for their support. I enjoyed the collaboration with them. During the weekly meetings, they gave me new insights, encouraged interesting discussions, gave me useful feedback, and asked critical questions. I appreciate the time spent helping me with my experiments and guiding me through the complex code basis. Even though the experiments did not always go as planned, which is part of testing with hardware I learned, they made even more time for me to record data again. Also thank you to Prof. Dr. Marco Hutter for giving me the opportunity to do this project at the RSL. It allowed me to tackle a challenging project, and more importantly, enjoy a living abroad experience.

Next to that, I would like to thank my supervisors from the TU Delft, Prof. Dr. David Abbink and Dr. Javier Alonso-Mora, for guiding me pleasantly through the process of my thesis. David can motivate people in a way that they always leave a meeting with a positive feeling. Although Javier his field of expertise is not in collision detection, he gave me a much appreciated other perspective to my research.

Finally, I would like to thank my family, study buddies, roommates, and the new friends that I made during my time here in Zürich, for their unconditional support. They were always in for discussions on my work, they supported me if I was not doing well and made working the many hours on this project a lot more enjoyable.

Delft, University of Technology | J. van Dam
February 22, 2022

Chapter 1

Paper ‘A review of collision detection, isolation and identification methods for legged robots’

A review of collision detection, isolation and identification methods for legged robots

J.I. van Dam
ETH Zürich, Delft University of Technology
Zürich, Switzerland
j.i.vandam@student.tudelft.nl
Supervision: Prof. Dr. D.A. Abbink, Dr. J. Alonso-Mora, A. Tulbure, Dr. F. Abi-Farraj, M.V. Minniti

***Abstract*—Legged robots have recently become sufficiently advanced to be deployed in unknown environments, where they could operate alongside humans or other robots and execute complex tasks. To safely interact with the environment, collision handling is crucial. Literature has proposed a collision pipeline to respond to an external force but so far, most studies have focused on fixed-base manipulators. Certainly, legged robots impose additional challenges due to their high number of Degrees of Freedom and static instability during trotting. We aim to provide a systematic review, comparison, and extension of various torque estimation, collision detection, and isolation methods. We specifically point out the requirements for implementation on legged robots. Finally, we validate the presented methods on a legged manipulator using extensive hardware experiments in various scenarios.**

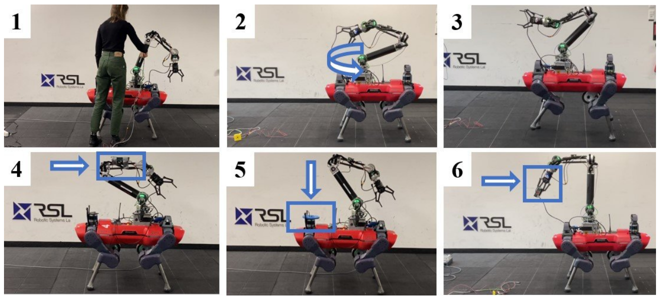


Fig. 1: The quadrupedal robot ANYmal with a 6-DoF manipulator mounted on top. The ground truth collision force of the contact applied at the arm is measured with a force/torque (F/T) sensor in multiple scenarios: (1) a collision applied in standstill; (2) arm motion; (3) trotting; (4) unmodeled $0.58\,\mathrm{kg}$ load on the forearm; (5) unmodeled $0.58\,\mathrm{kg}$ load on the base; and (6) $0.58\,\mathrm{kg}$ load in gripper, measured by a F/T sensor on the end-effector.

I. Introduction

A. Motivation

Over the past years, robot technology has developed fast and robots are finding a way into our daily lives. They start taking over jobs or collaborate with humans in industry [1], homes, offices, and hospitals [2]. To operate in these settings, robots have to safely interact with the environment. It is crucial to reliably detect and accurately estimate unexpected external forces. In this way, the robot can react appropriately to the unforeseen external contact and if desired, continue its motion or task.

Literature refers to the framework for responding to such external forces as the *collision event pipeline*, which separates a collision event into five phases [3]: detection, isolation, identification, classification, and reaction. In this work, we focus on the first three phases and additionally include the step prior to detection: torque estimation. *Torque estimation* is performed by observers that estimate the external torques on all Degrees Of Freedom (DoF) of the robot structure; *collision detection* aims to define when a collision occurs based on disturbances in the estimated torques; *collision isolation* localizes the external force's contact point; and finally, *collision identification* denotes the estimation of the external force with magnitude and direction. Model-free collision detection approaches compare velocity or motor torque measurements with their commanded values based on a predefined trajectory [4]. Its dependency on the trajectory and on sensor accuracy [5] is a limitation when working with robots that do not follow a predefined path. In this review, therefore, we assume the use of the robot dynamical model in detection, resulting in model-based methods. Furthermore, the use of solely *proprioceptive* (internal) sensors [3] is assumed. With external sensors, a collision cannot always be avoided, their implementation and maintenance costs are high and there is always the risk of sensor failure [6], [7].

The last two steps of the collision event pipeline are to classify the force and to react accordingly. Although we do not consider these steps in this work, we keep in mind that this is the final goal of the collision pipeline. To move around or comply with the contact forces, as a reaction to a collision, a legged robot, such as a humanoid or a quadruped, shows large advantages over fixed-base robots [8]. Legged systems can carry a high payload, are suitable to go across challenging terrains, and show redundancy, and are therefore capable of executing complex motions [9], [10]. Although research has focused on locomotion and stabilizing this complex robotic structure [11], [12], only few studies have looked into collision handling [10], [13]–[17].

B. Related work

Research in model-based collision detection often focuses on fixed-base manipulators [3] or wheel-based humanoids

[18], [19]. However, legged robots impose additional challenges, such as the static instability during trotting [10], multi-contact analysis during a collision because of the feet in contact with the ground [14], and the high number of DoFs [13]. The latter increases the computational time and tuning complexity.

In legged robot studies [10], [13]–[17], the focus is primarily on accurate force estimation, with some works additionally addressing the problem of isolation. For detecting the collision, all of these studies assume that an accurate robot model is available. Therefore, they do not provide a collision detection approach that is robust against model uncertainties.

On the contrary, works on fixed-base manipulators evaluate all five phases of the collision pipeline [3], and consider model inaccuracies in detection [20]–[23]. In [4], a detailed analysis of torque estimation and collision detection approaches is provided. The paper reviews some existing methods for manipulators and validates the torque estimation techniques on a 6-DoF robotic arm. However, the authors do not experimentally verify the discussed collision detection methods. Moreover, the torque estimation schemes are not compared to a ground truth, making it difficult to conclude on estimation accuracy. On the other hand, in [3], the torques estimated with various methods are compared to a ground truth and all five phases of the pipeline are experimentally verified. Its discussion on collision detection is limited, however, and no solution is provided for model inaccuracies. To the best of our knowledge, there is only one survey on legged robots comparing collision handling methods [24]. In [24], a summary is provided on external force estimation methods for humanoids specifically, but the schemes are not experimentally evaluated.

This work aims to extend the surveys in [3], [4]. In contrast to [3], [4], we will focus on implementation of the pipeline on legged robots. We will analyze approaches for the torque estimation, collision detection and collision isolation steps. To verify the torque estimation schemes, we will compare the estimations against a ground truth collision force. Next, we will evaluate the existing fixed-base manipulator collision detection methods that have been demonstrated to be robust against model inaccuracies, unmodeled payload, and dynamic motion. We will discuss if these approaches can be extended to a legged-robot implementation. For the collision isolation step, we will take a critical look at a technique whose performance has been demonstrated on humanoids. We will show that the approach is not always able to localize a colliding body. Finally, we will do extensive experimental validation of the torque estimation, collision detection, and isolation methods on a legged manipulator.

C. Outline

The pipeline for torque and force estimation, collision detection, isolation, and identification is depicted in Fig. 2, where the blocks of the scheme list the methods detailed in this work. The remainder of the paper is organized as follows. In Section II, the floating-base robot model is defined. Next, Section III introduces observers that estimate the external torques acting on the robot and the approach to calculate the force. Thereafter, Section IV outlines collision detection schemes that filter the estimated forces and set a threshold on these forces for detection. In Section V, we discuss the existing localization methods and the approach used for collision identification. Then, we compare the methods introduced in previous sections during experiments of which the results and analysis are presented in Section VI, before the work is concluded in Section VII.

II. Preliminaries

Throughout the paper, we use the symbol $\hat{a}$ to denote the estimate of a variable a. In addition, the 2-norm of a vector $\boldsymbol{a}$ is denoted as $|\boldsymbol{a}|$, while the absolute value of one of the components of the vector is denoted as $|a|$.

A *poly-articulated* robot is a robot with multiple rotary joints, like a legged or industrial robot [12]. The base of the robot is modelled as six virtual joints connected to one point, the *floating base*. These six joints are defined by three prismatic and three revolute joints [17]. Such a system can thus be seen as an unactuated 3D rigid body with multiple fully-actuated limbs attached to it. Generalized coordinates $\boldsymbol{q}$ are comprised by the base and joint variables, denoted with the subscripts b and j, respectively [14]

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_{\mathrm{b}} \\ \boldsymbol{q}_{\mathrm{j}} \end{bmatrix}, \text{ with } \quad \boldsymbol{q}_{\mathrm{b}} = \begin{bmatrix} \boldsymbol{r}_{\mathrm{b}} \\ \boldsymbol{\Phi}_{\mathrm{b}} \end{bmatrix} \in \mathbb{R}^{n_{\mathrm{b}}}, \quad \boldsymbol{q}_{\mathrm{j}} \in \mathbb{R}^{n_{\mathrm{j}}}, \qquad (1)$$

where $\boldsymbol{r}_{\mathrm{b}} \in \mathbb{R}^3$ is the Cartesian position of the base and $\boldsymbol{\Phi}_{\mathrm{b}}$ the base rotation. The size of n_{b} depends on the parameterization of the base orientation. Common parameterizations are quaternions, $\boldsymbol{\Phi}_{\mathrm{b}} \in \mathbb{R}^4$, or Euler angles, $\boldsymbol{\Phi}_{\mathrm{b}} \in \mathbb{R}^3$ [8]. The total number of robot DoFs is $6 + n_{\mathrm{j}}$ since the base velocity $\boldsymbol{v}_{\mathbf{b}}$ is defined with linear and angular velocity $\boldsymbol{u}_{\mathrm{b}}, \boldsymbol{\omega}_{\mathrm{b}} \in \mathbb{R}^3$.

The floating-base dynamic model is given by [12], [17]

$$\mathbf{M}(\boldsymbol{q})\dot{\boldsymbol{v}} + \mathbf{C}(\boldsymbol{q}, \boldsymbol{v})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \mathbf{S}^T \boldsymbol{\tau}_{\mathrm{m}} + \boldsymbol{\tau}_{\mathrm{ext}} + \boldsymbol{\tau}_{\mathrm{ft}}, \qquad (2)$$

where the matrix $\mathbf{M}(\boldsymbol{q}) \in \mathbb{R}^{(6+n_{\mathrm{j}})\times(6+n_{\mathrm{j}})}$ is the inertia matrix, symmetric and positive definite, $\mathbf{C}(\boldsymbol{q}, \boldsymbol{v}) \in \mathbb{R}^{(6+n_{\mathrm{j}})\times(6+n_{\mathrm{j}})}$ the matrix capturing centripetal and Coriolis effects, $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^{6+n_{\mathrm{j}}}$ the vector with torques resulting from gravity and $\boldsymbol{v} \in \mathbb{R}^{6+n_{\mathrm{j}}}$ the generalized velocities. The motor joint torques $\boldsymbol{\tau}_{\mathrm{m}} \in \mathbb{R}^{n_{\mathrm{j}}}$ in Eq. 2 are multiplied by the transpose of actuation matrix $\mathbf{S} = \begin{bmatrix} \mathbf{0}_{n_{\mathrm{j}}\times 6} & \mathbf{I}_{n_{\mathrm{j}}\times n_{\mathrm{j}}} \end{bmatrix}$, mapping the n_{j} motor torques into $6 + n_{\mathrm{j}}$ dimensional space. All external forces acting on the robot structure result in force/torque (F/T) sensor torques $\boldsymbol{\tau}_{\mathrm{ft}}$, which are measured, and external torques $\boldsymbol{\tau}_{\mathrm{ext}}$, which are estimated. A F/T sensor *wrench* $\boldsymbol{\mathcal{F}}_{\mathrm{ft}} \in \mathbb{R}^6$ consists of force $\boldsymbol{F}_{\mathrm{ft}} \in \mathbb{R}^3$ and moment $\boldsymbol{m}_{\mathrm{ft}} \in \mathbb{R}^3$. For multiple F/T sensors n_{ft} attached to the structure, these wrenches result in

$$\boldsymbol{\tau}_{\mathrm{ft}} = \begin{bmatrix} \mathbf{J}_{\mathrm{ft},1}^T(\boldsymbol{q}) & \dots & \mathbf{J}_{\mathrm{ft},n_{\mathrm{ft}}}^T(\boldsymbol{q}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{F}}_{\mathrm{ft},1} \\ \vdots \\ \boldsymbol{\mathcal{F}}_{\mathrm{ft},n_{\mathrm{ft}}} \end{bmatrix}, \qquad (3)$$

where the *geometric* contact Jacobian $\mathbf{J}_{\mathrm{ft},j}(\boldsymbol{q})$ depends on the robot configuration and is defined in joint space [8]. It

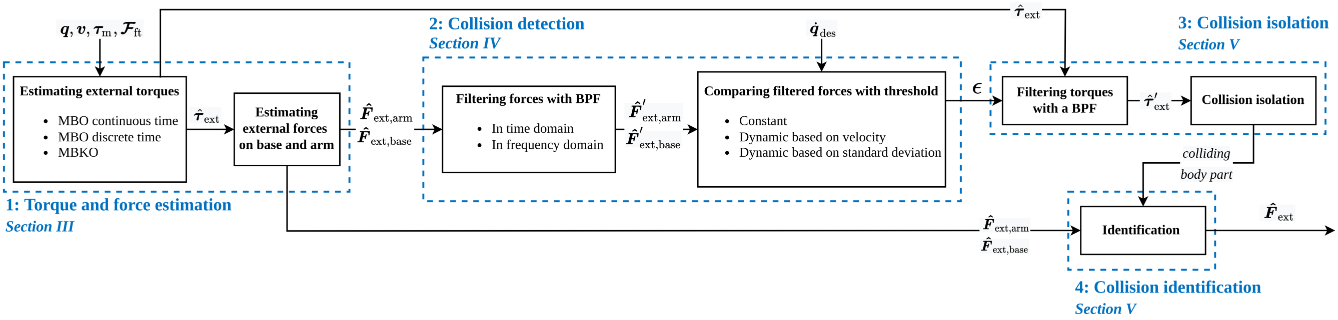


Fig. 2: The collision event pipeline consisting of four steps: (1) observing the external torque $\boldsymbol{\tau}_{\text{ext}}$; (2) deciding if a collision occurs ($\epsilon = 1$ or $\epsilon = 0$) by estimating the force, filtering it and comparing it to a threshold; (3) isolation of the colliding body part; and (4) identifying the magnitude and direction of the force. With final output force $\hat{\boldsymbol{F}}_{\text{ext}}$, a reaction strategy can be implemented.

determines the transformation from the local F/T sensor frame j, in which $\boldsymbol{\mathcal{F}}_{\text{ft}}$ is expressed, to the world frame.

Assuming proprioceptive sensing, a robot is equipped with torque and position sensors in the joints that measure motor torque, joint position and joint velocity. Sensor measurements of an Inertial Measurement Unit (IMU) attached to the base, contact states of the feet and joint positions are fused in a state estimator that calculates base orientation and twist. Acceleration is not measured but obtained via numerical differentiation. The inertial and gravitational terms, and the mass matrix in Eq. (2) can be computed using rigid body dynamics algorithms.

The robotic system used for our experimental validation is illustrated in Fig. 3. It has 18 actuated joints, $n_{\text{j}} = 18$: 3 rigid joints per leg and 6 rigid joints in the arm. The rotation of the base $\boldsymbol{\Phi}_{\text{b}}$ is given in quaternions. The task force, coming from the object the gripper is holding, is read by a F/T sensor, such that $n_{\text{ft}} = 1$.

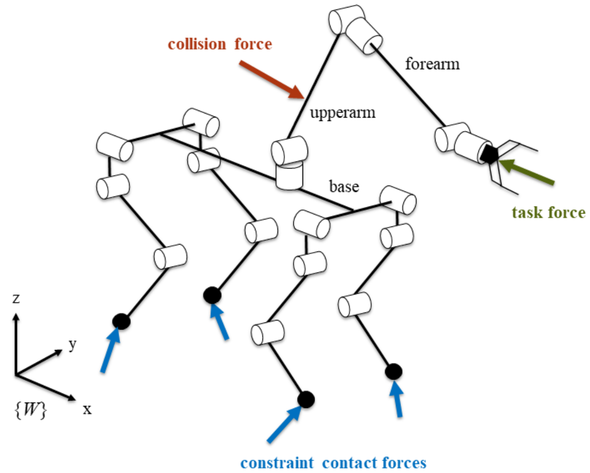


Fig. 3: Floating-base model of the robot used in this work. The robotic system has four legs, all consisting of three actuated revolute joints; an unactuated base; and a robotic arm with six actuated revolute joints placed on top of the base. The main reference frame is the world frame $\mathcal{W}$. In the figure, an illustrative scenario is represented where external contact forces result from the constraint contacts of the feet touching the ground, an external task force at the end-effector of the arm and a single collision force, which is in this case acting on the upperarm.

III. Torque and Force Estimation

This section highlights step 1 of the collision pipeline in Fig. 2. Previous works focusing on legged robots have proposed several methods to estimate the external torques. Direct estimation [13], [25] uses the acceleration terms, whose estimation amplifies measurement noise due to numerically differentiating the joint positions twice. On the other hand, [15] eliminates acceleration and velocity from the direct estimation expression, assuming that the robot is at standstill. This method, however, is limited to scenarios with static equilibrium. Another approach by [16] is a Kalman filter with the applied force as output and readings of the force-sensing resistors (FSR) on the feet and an IMU as an input. A limitation is that these sensors are not always available.

A momentum-based disturbance observer (MBO) that monitors the generalized momentum, first introduced in [26], is most commonly in use since it is relatively simple, allows for simultaneous collision detection, isolation and identification, and is not dependent on a specific type of robot, controller or trajectory. Moreover, no inertia matrix inversion is required and it eliminates the need of computing the joint accelerations. The linear observer generates a first-order filtered version of the estimated external joint torques.

The low-pass filter of the MBO can be extended to a higher-order filter [10], [27], [28], resulting in more accurate estimation due to the sharper filtering action. To the best of our knowledge, [11] is the only work that has verified its third-order MBO on a legged robot: the 12-DoF legs of a quadruped. To test its performance on the 24-DoF legged manipulator used in this paper, we tuned the diagonals of the three filter matrices. Since a higher-order filter causes more instability

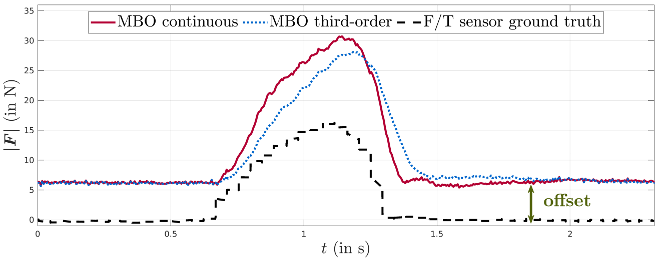


Fig. 4: A collision applied on the arm, where the force estimated with a first or a third-order MBO is compared against the ground truth collision force. Note that the offset before and after collision is due to model inaccuracies and sensor noise.

and oscillations, selecting the gains is challenging [29]. Additionally, the number of tuning parameters is high. As can be seen in Fig. 4, results show more accurate estimation but an increase in delay for the third-order MBO. Therefore, we choose not to include this observer in our final experimental comparison.

Another variation of the conventional MBO is a filtered dynamics observer [4], [20], which gives equal results to the classic MBO [4]. Other MBO alternatives are a sliding mode observer [4], and a nonlinear momentum observer [4], [5]. The latter models the disturbance as a nonlinear function and constructs a state-space representation of this system. In [5], it is shown that the nonlinear observer decreases the estimation error compared to the conventional MBO and that a peak in estimated torque that initially appears at the start of estimation is reduced. However, if the collision algorithm runs continuously, this peak at the start is irrelevant. Furthermore, the number of parameters to tune for the nonlinear observer in [5] is seven times larger than for the classical MBO. Another challenge regarding tuning is given by [4], who demonstrates the complexity of selecting the gain matrices such that the torque estimation error converges towards zero, both for the nonlinear and the sliding mode observer. Because of this complexity in selecting the gains, we do not implement the observers in this paper.

As an alternative, the Momentum Based Kalman Observer (MBKO) [30] estimates the momentum and external torques by applying a filter. Where the MBO uses the low-pass filtering property of the gain matrices, the MBKO filters the estimation with covariance matrices. Kalman filtering has proved to reduce phase lag and stability issues, and decrease noise compared to conventional first-order low-pass filters [31]. The additional DoF in Kalman filtering compared to the MBO arise from the three covariance matrices, instead of one observer gain matrix, and thus allow improvement in estimation accuracy. Although, similar to the nonlinear and higher-order MBOs, more tuning is required, [30] provides a simple calibration routine for the covariance matrices.

For this reason, in our experiments, we compare the conventional linear first-order MBO and MBKO to examine the theoretical improved estimation accuracy of the latter. In this section, the formulation of the MBO implemented in continuous and discrete-time, and the MBKO are given. Comparison between the continuous MBO and MBKO has been conducted before [30], [32], [33], but only for a 6- or 7-DoF manipulator, and no comparison with the discrete-time MBO has been presented.

A. Momentum Based Observer

a) Continuous time implementation: The observer was first defined for a robotic manipulator [26], and later extended to a floating base structure [10], [14], [17], [34]. The generalized momentum $\boldsymbol{p} \in \mathbb{R}^{6+n_{\text{j}}}$ is defined as $\boldsymbol{p} = \mathbf{M}(\boldsymbol{q})\boldsymbol{v}$ and let the nonlinear terms $\boldsymbol{n}(\boldsymbol{q}, \boldsymbol{v}) := \boldsymbol{g}(\boldsymbol{q}) + \mathbf{C}(\boldsymbol{q}, \boldsymbol{v})\boldsymbol{v} - \dot{\mathbf{M}}(\boldsymbol{q})\boldsymbol{v}$. The external torque estimate is given as follows

$$\hat{\boldsymbol{\tau}}_{\text{ext}} = \mathbf{K}_{\text{O}} \left(\boldsymbol{p}(t) - \int_0^t \left(\mathbf{S}^T \boldsymbol{\tau}_{\text{m}} + \boldsymbol{\tau}_{\text{ft}} - \hat{\boldsymbol{n}}(\boldsymbol{q}, \boldsymbol{v}) + \hat{\boldsymbol{\tau}}_{\text{ext}} \right) \mathrm{d}s \right), \tag{4}$$

where $\mathbf{K}_{\text{O}}$ is the positive diagonal observer gain matrix. By adjusting the elements of the gain matrix $\mathbf{K}_{\text{O}}$, the bandwidth of the external torque estimate changes. When choosing a smaller value, unwanted high frequency noise is reduced, although it does come with detection delay.

b) Discrete time implementation: A variation on the classical MBO is introduced in [35]. The authors state that issues arise when transforming the integral in Eq. (4) from continuous to discrete time. The discretization suffers from modeling errors in the residual that come in as fictional disturbances during fast motion. As a solution, the work formulates the algorithm directly in discrete time. In this paper, the formulation in [35] is extended to a floating-base expression. Moreover, the observer gain values β and γ are expanded to positive diagonal gain matrices $\mathbf{B}$ and $\boldsymbol{\Gamma}$, respectively, with dimension $6 + n_{\text{j}}$ such that the gains can be tuned for each DoF separately

$$\hat{\boldsymbol{\tau}}_{\text{ext}} = \mathbf{B}\boldsymbol{p}(k) - \frac{\mathbf{I} - \boldsymbol{\Gamma}}{\mathbf{I} - \boldsymbol{\Gamma} z^{-1}} \left(\mathbf{S}^T \boldsymbol{\tau}_{\text{m}} - \hat{\boldsymbol{n}}(\boldsymbol{q}, \boldsymbol{v}) + \mathbf{B}\boldsymbol{p} \right), \tag{5}$$

where z is the z-domain variable, the diagonal gain entries $0 < \gamma_i < 1$ of $\boldsymbol{\Gamma}$ are values monotonically related to the cut-off frequency and entries β_i in $\mathbf{B}$ are defined as $\beta_i = \frac{(1-\gamma_i)\gamma_i^{-1}}{T_{\text{s}}}$ with T_{s} the sampling time.

B. Momentum Based Kalman Observer

The first-order observer for generalized momentum as mentioned in the previous section is extended with the design of a Kalman filter that estimates the generalized momentum $\boldsymbol{p}$ and external torques $\boldsymbol{\tau}_{\text{ext}}$ simultaneously. The definitions below are taken from [30], [32], [33] and reformulated from a fixed-base manipulator to a floating-base robot expression.

Let $\bar{\boldsymbol{\tau}} = \mathbf{S}^T \boldsymbol{\tau}_{\text{m}} - \hat{\boldsymbol{n}}(\boldsymbol{q}, \boldsymbol{v})$. The continuous system dynamics are given as

$$\underbrace{\begin{bmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{\tau}}_{\text{ext}} \end{bmatrix}}_{\dot{\boldsymbol{x}}} = \underbrace{\begin{bmatrix} \mathbf{0}_{N \times N} & \mathbf{I}_N \\ \mathbf{0}_{N \times N} & \mathbf{A}_{\tau} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \boldsymbol{p} \\ \hat{\boldsymbol{\tau}}_{\text{ext}} \end{bmatrix}}_{\boldsymbol{x}} + \underbrace{\begin{bmatrix} \mathbf{I}_N \\ \mathbf{0}_{N \times N} \end{bmatrix}}_{\mathbf{B}} \underbrace{\bar{\boldsymbol{\tau}}}_{\boldsymbol{u}} + \boldsymbol{w}, \tag{6}$$

where for ease of notation the term $N = 6+n_\mathrm{j}$, $\boldsymbol{x} \in \mathbb{R}^{2N}$ is the state vector and $\boldsymbol{u} \in \mathbb{R}^N$ the input vector. Matrix $\mathbf{A}_\tau \in \mathbb{R}^{N\times N}$ is typically chosen to be $\mathbf{0}_{N\times N}$, however, defining a negative diagonal for the matrix allows to eliminate constant offsets in the external torques. Vector $\boldsymbol{w} = \begin{bmatrix}\boldsymbol{w}_p^T & \boldsymbol{w}_\tau^T\end{bmatrix}^T \in \mathbb{R}^{2N}$ describes the process noise in the generalized momentum dynamics $\dot{\boldsymbol{p}}$ and the external torques. The noise in the momentum results from modeling uncertainties. The term $\boldsymbol{w}$ is expressed with covariance matrices: $\boldsymbol{w}_\tau \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_\tau)$ and $\boldsymbol{w}_p \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_p)$ with $\mathbf{Q}_\tau, \mathbf{Q}_p \in \mathbb{R}^{N\times N}$. Besides process noise, measurement noise exists, which is visible in the generalized momentum term $\boldsymbol{p} = \mathbf{M}\boldsymbol{v}$ due to dependence on measurements $\boldsymbol{q}$ and $\boldsymbol{v}$. The noise is indicated with $\boldsymbol{v_p} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ with $\mathbf{R} \in \mathbb{R}^{N\times N}$ and appears in the output dynamics

$$\underbrace{\boldsymbol{p}}_{\boldsymbol{y}} = \underbrace{\begin{bmatrix} \mathbf{I}_N & \mathbf{0}_{N\times N} \end{bmatrix}}_{\boldsymbol{C}} \underbrace{\begin{bmatrix} \boldsymbol{p} \\ \hat{\boldsymbol{\tau}}_\mathrm{ext} \end{bmatrix}}_{\boldsymbol{x}} + \boldsymbol{v_p}. \tag{7}$$

Next, the continuous system Eq. (6), Eq. (7) is discretized to a linear time-invariant system and the Kalman filter update steps are followed. For the discretization process and Kalman filter update algorithm, the reader is referred to [30], [32], [33].

C. Force estimation

Once the external torques $\boldsymbol{\tau}_\mathrm{ext}$ are estimated, the stacked force vector is calculated. We aim to obtain *wrench* $\boldsymbol{\mathcal{F}}_{\mathrm{ext},i} = (\boldsymbol{F}_{\mathrm{ext},i}, \boldsymbol{m}_{\mathrm{ext},i}) \in \mathbb{R}^6$, which is applied to the colliding link i and consists of a force and torque component. We assume one external collision force only. The Moore-Penrose pseudoinverse operation, indicated with $^\#$, is applied to the matrix stacking the geometric Jacobians of the contact points [14], [17]

$$\begin{bmatrix} \hat{\boldsymbol{F}}_{\mathrm{f},1} \\ \vdots \\ \hat{\boldsymbol{F}}_{\mathrm{f},4} \\ \hat{\boldsymbol{\mathcal{F}}}_{\mathrm{ext},i} \end{bmatrix} = \begin{bmatrix}\mathbf{J}_{\mathrm{f},1}^T(\boldsymbol{q}) & \dots & \mathbf{J}_{\mathrm{f},4}^T(\boldsymbol{q}) & \mathbf{J}_i^T(\boldsymbol{q})\end{bmatrix}^\# \hat{\boldsymbol{\tau}}_\mathrm{ext}, \tag{8}$$

where $\boldsymbol{F}_{\mathrm{f},j}$ is the force on contact foot j with $\boldsymbol{J}_{\mathrm{f},j} \in \mathbb{R}^{3\times(6+n)}$ its translational Jacobian, and $\boldsymbol{J}_i \in \mathbb{R}^{6\times(6+n)}$ the spatial Jacobian of the colliding point i [14], [17]. Here, the contact feet and the collision links are modeled as point contacts and thus are only subject to linear forces. However, since the exact contact location of the collision force is unknown, the spatial Jacobian of an arbitrary robot link is used in Eq. (8), which results in the additional torque $\boldsymbol{m}_{\mathrm{ext},i}$ caused by the impact force.

Note that to find a solution to Eq. (8), first of all, the stacked Jacobian has to be full rank [3]. Structural lack of Jacobian rank occurs often for impact forces on links proximal to the base. These link Jacobians have larger null spaces and the force is undetectable if it is in the stacked Jacobian's null space, resulting in a non-unique mapping of the torques [36]. Secondly, the robot should stay away from singularities [3], [15]. In both cases, internal reaction forces and moments compensate for the wrench applied on the robot, which causes some of the components of $\boldsymbol{\mathcal{F}}_\mathrm{ext}$ to be transferred into the robot's mechanical structure. Therefore, we avoid singular configurations during the experiments. Other situations in which the estimation of the wrench is affected include cases when the external force does not perform virtual work on the robot motion, e.g. when the contact force is parallel to the joint axis connected to the contact link i.

In this work, we are interested in monitoring collisions that could happen during manipulation or human-robot collaboration tasks. In such scenarios, it is reasonable to assume that collisions with the legs are less likely to occur than collisions with the base or the arm of the robot. Thus, we exclude the former type of collisions from our analysis. Note that, at this stage of the pipeline (see Fig. 2), it is not possible to know whether a collision is occurring at the base or the arm of the robot. To resolve this ambiguity, we solve Eq. (8) for each of the two cases, resulting in $\hat{\boldsymbol{F}}_\mathrm{ext,arm}, \hat{\boldsymbol{F}}_\mathrm{ext,base}$. For the arm, the Jacobian of an arbitrary manipulator link can be chosen (e.g., the end-effector). In the following sections, we neglect the moment component of the wrench $\hat{\boldsymbol{\mathcal{F}}}_{\mathrm{ext},i}$ and only consider $\hat{\boldsymbol{F}}_{\mathrm{ext},i}$. This force serves as an input for collision detection.

IV. Collision detection

The largest limitation of model-based collision detection methods is that these approaches rely on an accurate robot dynamical model. The main challenge is therefore to separate external disturbances from model inaccuracies to prevent undesired false positives (FP). It is common to reduce the modeling errors by going to the root of the problem and improve the model accuracy by (offline) identification [4]. This often requires the use of noisy acceleration estimations, it is a time consuming process and the identification accuracy is dependant on the excitation level of the reference trajectory [4]. Furthermore, complete elimination of modeling errors can never be achieved. Other solutions include filtering of the estimated torques or forces before detecting it, setting a time- or velocity-based dynamic threshold, or using a model-adaptive technique for setting a threshold.

We will consider these solutions in this section, where we describe step 2 of Fig. 2, collision detection. The estimated external forces $\hat{\boldsymbol{F}}_\mathrm{ext,arm}, \hat{\boldsymbol{F}}_\mathrm{ext,base}$ from step 1 are filtered as a measure of detection, either in time or in frequency domain, which we discuss in Section IV-A. Next, constant threshold and dynamic thresholds, are introduced in Section IV-B.

A. Filtering the force

A low-pass filter (LPF) decreases the undesired high frequency (measurement) noise, but increases detection delay [3]. A high-pass filter (HPF) attenuates the lower frequencies. Low frequencies result either from the nominal motion dynamics of the robot, or from the modeling errors, which can be seen as a signal with frequency 0 Hz. The latter appears as a static offset in the force estimation and is visualized in Fig. 4. A band-pass filter (BPF) combines the two filters to exploit the advantages of both. In [20], torques filtered with a second-order BPF,

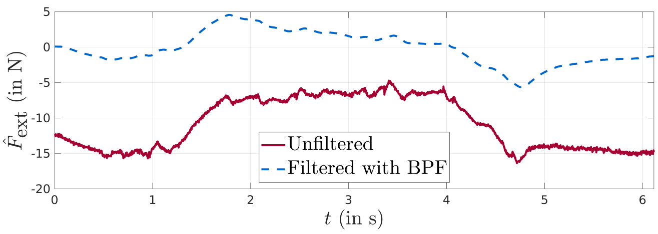


Fig. 5: The unfiltered estimated external force is oscillating due to arm motion, while the static offset is caused by modeling errors and sensor noise. Filtering with the BPF reduces noise and offset when no collision is applied yet.

compared to unfiltered torques, allow reducing the detection threshold with an order 5. Consequently, the torques cross the threshold faster and thus detection delay is lowered. The effect of the BPF is illustrated in Fig. 5. In the subsequent two paragraphs, the application of the BPF in time and frequency domain is discussed.

a) Time domain: Implementation of a BPF for external torques in the time domain for collision detection, has been demonstrated in [21], [37], where [21] additionally compares the BPF to an LPF and HPF and uses two observers in parallel, one to detect slow and one to detect fast impacts. Furthermore, [20] experimentally verifies different types of BPFs and considers higher-order BPFs. For the latter, a limitation is the time delay that grows with increasing order [38], therefore, in this work, we implement a first-order BPF. Note that in contrast to [21], [37], we filter the forces instead of the torques.

The first-order continuous-time transfer function of a BPF is as follows [37]

$$G(s) = \frac{\omega_{\max} s}{(\omega_{\min}\omega_{\max}) + (\omega_{\min} + \omega_{\max})\, s + s^2}, \tag{9}$$

where $\omega_{\min}$ and $\omega_{\max}$ are the minimum and maximum frequency, respectively. Applying this filter to each component of a force $\hat{\boldsymbol{F}}_{\text{ext}}$ in the Laplace domain gives filtered force $\hat{\boldsymbol{F}}'_{\text{ext}}$ as follows

$$\hat{F}'_{\text{ext}} = G(s)\hat{F}_{\text{ext}}. \tag{10}$$

b) Frequency domain: Analyzing a quantity in the frequency instead of the time domain, by using a Fast Fourier Transform (FFT) convolution, reduces noise and phase-shifting and its use in collision detection has been demonstrated in [39], [40]. A window with constant size N_{w} is sliding over the force measurements in the time domain and shifts one time instant further when a new measurement enters the time spectrum. The input for the FFT is a vector of all N_{w} force measurements $\hat{\boldsymbol{F}}_{\text{ext},i}$ that are within the current window in the time domain. This input vector is denoted as $\boldsymbol{x}$. Transforming this time series to the frequency domain with a Discrete Fourier Transform (DFT) for every direction x, y, z of the force gives

$$\hat{F}^m_\omega = \left\| \sum_{n=0}^{N_{\text{w}}-1} w_n x_n \mathrm{e}^{\frac{-2\pi i}{N_{\text{w}}} mn} \right\|, \quad \text{for } m = 0, \ldots, N_{\text{w}} - 1, \tag{11}$$

where the DFT is weighted by window w_n. Note that the i in Eq. (11) denotes a complex number, and not the colliding body link as before. The FFT implements the DFT efficiently: complexity is decreased from $\mathcal{O}(N_{\text{w}}^2)$ to $\mathcal{O}(N_{\text{w}} \log_2 N_{\text{w}})$ [41]. The window w_n in Eq. (11) can take various forms, such as rectangular, which is the most conventional, or a half-Hann window. We use the half-Hann window in this work. It prioritizes the current measurements over the past by adjusting the weight, thus reducing delay, while maintaining high frequency resolution [42].

Choosing N_{w} means considering the trade-off between a high frequency resolution $\Delta\omega$ and a high time resolution Δt [43], [44]. The frequency spectrum of a signal in the current window is divided into 'bins', where each bin has size $\Delta\omega$. A signal showing a peak around a certain frequency will be split between the two bins surrounding that frequency. If the bin size is too large, no clear distinction between two different signals is visible. Therefore, a low value for $\Delta\omega$ is desired. Similarly, the goal is to have a low Δt, which defines the time window size the FFT is calculated over. A smaller time window means less delay. The relation between these two resolutions is described with the window size

$$N_{\text{w}} = \frac{1}{T_{\text{s}}\Delta\omega} = \frac{\Delta t}{T_{\text{s}}}. \tag{12}$$

So far, we have seen how to calculate a frequency-domain power spectrum over a windowed force segment. Focusing on a subset of frequencies in the power spectrum, indicated by $[\omega_{\min}, \omega_{\max}]$, allows filtering. Next, the filtered force magnitude values $\hat{\boldsymbol{F}}_{\omega_{\min}:\omega_{\max}}$ within the spectrum for the current time, are transformed to one value by taking the 1-norm $\|..\|_1$. Experimental comparison between the 1-norm, 2-norm and ∞-norm [40] shows that the 1-norm gives less delay and noise. The final frequency filtered force is then defined as

$$\hat{F}'_{\omega_{\min}:\omega_{\max}} = \left\| \hat{\boldsymbol{F}}_{\omega_{\min}:\omega_{\max}} \right\|_1, \tag{13}$$

where the $\hat{F}'_{\omega_{\min}:\omega_{\max}}$ denotes the final filtered force value at time step t, converted back from decibel to an absolute force value in N. This filtering scheme, using the derivative of the 1-norm instead of Eq. (13), is validated in [40] and compared to the time-domain BPF presented in [37]. The authors conclude that filtering in frequency domain gives shorter detection delay, while keeping equal success rate in collision detection.

B. Setting a threshold

Preventing FPs as much as possible, while keeping the threshold low for fast detection and a high collision detection success rate, means choosing and tuning a threshold well. Model-adaptive techniques have been verified to show good detection performance, especially in case of modeling errors. These thresholds require parameter identification, which is done offline or online, and optimal threshold coefficients are found by solving a least squares problem.

Offline identification requires datasets of the robot moving in a predefined [5] or a randomly set trajectory [23], [45], [46]. The accuracy of the threshold depends on the excitation level of the trajectory. However, it has not been verified if it is possible to set out such a well-designed trajectory for a floating-base system with more than 7 DoF, which generates complex motions in a large workspace. Furthermore, obtaining these datasets and processing them is time-consuming and the review in [4] states that it is unsure if the improvement in detection time is worth the time spent experimenting. Conducting the threshold parameter identification online [22], [47] has only been implemented on manipulators with 2 DoF since computation time rises significantly for robots with a higher number of DoF. We therefore choose not to implement these model-adaptive thresholds on our legged manipulator. Thus, the constant threshold and two simple dynamic thresholds based on velocity and standard deviation are discussed in the following paragraphs.

a) Constant threshold: The most standard method for detecting collision is setting a constant threshold [26], [48] and observing when the force exceeds it

$$\epsilon = \max\left(\frac{|\hat{F}'_{\mathrm{ext},x}|}{b_x}, \frac{|\hat{F}'_{\mathrm{ext},y}|}{b_y}, \frac{|\hat{F}'_{\mathrm{ext},z}|}{b_z}\right) > 1. \tag{14}$$

Here, b_i denotes the threshold for the x, y, and z filtered forces. The binary value ϵ defines if a collision occurs. The main limitation of the constant threshold, as proved in previous works [26], [48], is the detection of FPs when a robot task includes high-frequency content or false negatives (FN) for collisions with light impact.

b) Dynamic threshold based on velocity: Velocity-dependent uncertainties in the robot model are a result of not including a friction model, which is generally based on velocity, or inaccurate friction modeling [45]. The idea of the dynamic threshold based on velocity $\boldsymbol{b}_{\mathrm{dyn},\dot{q}}$ is that the threshold moves along with the inaccuracies appearing in the estimated force, and during arm motion. Once the collision occurs, the estimated force increases, while the desired velocity and thus the dynamic threshold stays low. Consequently, the impact is detected.

The expression below is adapted from [49], [50]. The former additionally includes acceleration in the formulation. We choose to eliminate this since the acceleration term is noisy and after our own experiments showed not to add any value to the threshold. Moreover, [49] designs the threshold for each estimated external torque of a manipulator separately, only taking into account the velocity value of that joint. For legged robots, however, the movement of certain body parts influences the estimated torque values of other body parts. As a result, the expression has been extended to include the vector of velocities as follows

$$b_{\mathrm{dyn},\dot{q}} = b_{\mathrm{stat},\dot{q}} + \sum_{i=1}^{n_{\mathrm{j}}} k_{\dot{q},i}^T \frac{|\dot{q}_{\mathrm{des},i}|}{\dot{q}_{\mathrm{max},i}}, \tag{15}$$

where $b_{\mathrm{dyn},\dot{q}}$ is the dynamic threshold for one of the three filtered force directions, with static part $b_{\mathrm{stat},\dot{q}} > 0$ tuned for each of the three components. Positive gain $k_{\dot{q},i}$ is tuned for each joint separately and maximum joint velocity $\dot{q}_{\mathrm{max},i}$ is obtained from the robot or controller specifications. Note that desired velocities $\dot{q}_{\mathrm{des},i}$ instead of measured are used to avoid delays between the commanded and actual velocity. To get the threshold to move along with the force smoothly, an LPF is added on $b_{\mathrm{dyn},\dot{q}}$, where cut-off frequency $\omega_{b_{\mathrm{dyn}}} > \omega_{\mathrm{max}}$ (ω_{max} being the external force BPF frequency) to prevent threshold delay. The threshold $\boldsymbol{b}_{\mathrm{dyn},\dot{q}} \in \mathbb{R}^3$ can be inserted in Eq. (14) to detect collision.

c) Dynamic threshold based on standard deviation: In [51], a novel dynamic thresholding method is proposed which integrates proximity sensors for perceiving the environment prior to a collision event. Since in this work only proprioceptive sensing is assumed, the threshold formulation in [51] is reduced to the following:

$$b_{\mathrm{dyn},\sigma} = b_{\mathrm{stat},\sigma} + \min\left(c \cdot \frac{\sigma}{\sigma_{\mathrm{max}}}, F_\sigma\right), \tag{16}$$

where $b_{\mathrm{dyn},\sigma}$ is the dynamic threshold for one of the three filtered force directions, with static part $b_{\mathrm{stat},\sigma} > 0$ tuned for each of the three components. The standard deviation σ of the unfiltered force $\hat{F}_{\mathrm{ext}}$ is taken over a fixed size sliding time window of length N_σ and its maximum is σ_{max}. Constant F_σ determines the maximum amount that the dynamic threshold can increase and is experimentally chosen for each force direction, just like the constant c. Similarly to the other dynamic threshold, an LPF is added to bound $b_{\mathrm{dyn},\sigma}$. Again, the obtained dynamic threshold can replace b_x, b_y, b_z in Eq. (14) to detect collision.

V. Collision isolation and identification

Here, we present the third and fourth step of the collision pipeline in Fig. 2, collision isolation and identification.

A. Collision isolation

a) Body link localization: A common approach to isolate the colliding body link is based on the assumption that, when a collision occurs at contact link i in an open kinematic chain structure, this contact does not produce torques along the joints more distal from the link in contact [3]. Consequently, the first link i for which $|\hat{\tau}_{\mathrm{ext},i}| > b$, with b a threshold larger than 0 to account for model inaccuracies, is defined as the one on which the external force is applied. This approach has also been applied on humanoid robots [14], [25].

However, the above-described method does not provide guarantees that the identification of the colliding body link is possible. To investigate this statement further, we conduct and discuss experiments attempting to distinguish between body links in Section VI-D. Note that the estimated external torques $\hat{\boldsymbol{\tau}}_{\mathbf{ext}}$ in Fig. 2 in step 3 are filtered with a BPF first, before they are used as an isolation measure, to reduce offset and noise.

b) Contact point localization: Once the colliding body link is obtained with the above-described technique, an analytical method [3] can be applied to reconstruct the exact contact point on the link. This approach relies on the assumption that the external contact is punctual, so moment $\boldsymbol{m} = \mathbf{0}$. A limitation of this technique is that an accurate force estimate is required. Works that have attempted contact point localization on humanoids demonstrate that this is possible in simulation [13], [14]. During hardware experiments [15], [25], however, the static joint friction has a negative impact on this procedure. Consequently, the force estimate is inaccurate and thus the contact point cannot be reconstructed. We therefore choose not to experimentally validate this method.

B. Collision identification

Assuming we are able to distinguish a force acting on the base or arm with the colliding body link isolation method, we define

$$\hat{\boldsymbol{F}}_{\text{ext}} = \begin{cases} \hat{\boldsymbol{F}}_{\text{ext,arm}}, & \text{if arm collision} \\ \hat{\boldsymbol{F}}_{\text{ext,base}}, & \text{if base collision.} \end{cases} \tag{17}$$

The direction of the force is indicated by looking at each component of $\hat{\boldsymbol{F}}_{\text{ext}}$ and its magnitude is denoted as $|\hat{\boldsymbol{F}}_{\text{ext}}|$.

VI. CASE STUDY: LEGGED MANIPULATOR

We verify the collision estimation, detection, and isolation methods, developed using C++ and ROS, through hardware experiments on the legged manipulator [9], consisting of the quadruped base Anymal C [1] with the robotic arm DynaArm mounted on it. The robot is equipped with a RobotiQ 2F-85 gripper [2] and a BOTA Rokubi SenseOne 6-DoF F/T sensor [3] at its end-effector.

The collected collision data is summarized in Table I, and the scenarios are visualized in Fig. 1. A joystick is used to send references to the robot for arm motion. Examples of collisions in the different scenarios considered for method comparison are shown in the accompanying video: https://youtu.be/tnrX3mA4lZI. Ground truth contact data is read from a hand-held BOTA Rokubi Mini 6-DoF F/T sensor; the collisions are created by either pushing the F/T sensor on different parts of the arm or base of the robot or by holding it still while the arm collides with it.

The torque estimation methods (step 1 of Fig. 2), filtering and threshold approaches (step 2), and isolation technique (step 3) are compared in the following sections. Step 4, collision identification, is only about placing together the outputs of steps 1 and 3 and we will thus not discuss this step in the results. In Table II, the discussed methods are compared based on their computation time and tuning simplicity.

A. Torque estimation comparison

In this first set of experiments, we compare the torque estimation methods, as presented in Section III. We use these approaches to obtain an estimate of the generalized external torque $\hat{\boldsymbol{\tau}}_{\mathbf{ext}}$. Afterwards, we compute the estimated external forces $\hat{\boldsymbol{F}}_{\mathbf{ext}}$ by substituting these torques in the force estimation expression Eq. (8).

[1]https://www.anybotics.com/

[2]https://robotiq.com/products/2f85-140-adaptive-robot-gripper

[3]https://www.botasys.com/

TABLE I: The 416 collisions applied to the legged manipulator, split up into different scenarios.

Scenario		Number of collisions		
		Arm	Base	Total
Standstill		76	33	109
Arm motion		17	23	40
Trotting		12	-	12
Unmodeled load on arm or base	**In stance**	99	-	99
	In motion	13	-	13
Load in gripper	**In stance**	133	-	133
	In motion	10	-	10

TABLE II: Comparison between torque estimation and collision detection (filtering, thresholds) methods.

		Equation number	Tuning simplicity	Computation time (in ms)
Torque estimation	**MBO continuous**	(4)	$+$	$0.8 \cdot 10^{-3}$
	MBO discrete	(5)	$+$	$1.8 \cdot 10^{-3}$
	MBKO	Kalman update steps on (6), (7)	$-$	1.5
Filtering	**Time domain**	(9)	$++$	$1.4 \cdot 10^{-4}$
	Frequency domain	(13)	$+$	DFT: 4.3 FFT: 0.02
Thresholds	**Constant**	(14)	$++$	$1.1 \cdot 10^{-3}$
	Dynamic $\dot{\boldsymbol{q}}$	(15)	$+$	$4.0 \cdot 10^{-3}$
	Dynamic $\boldsymbol{\sigma}$	(16)	$+$	$2.0 \cdot 10^{-3}$

The diagonal of the gain matrix $\mathbf{K}_{\text{O}}$ for the continuous-time MBO is tuned by looking at the torques of each of the joints separately. To ensure a fair comparison between the continuous and discrete-time MBO implementation, the diagonal continuous-time gains k_{O} are converted to the z-domain with $\gamma = e^{-k_{\text{O}}\Delta t}$. For a more detailed report of the calibration routines for tuning the Kalman covariance matrices, readers are referred to [30].

For comparison, we apply collisions during standstill and arm motion, a total of 109 and 40 contacts, respectively (see Table I). We include arm movement specifically since [35] states that the discrete-time MBO outperforms the continuous-time version during motion. Tests are performed over a wide range of arm configurations, contact locations (arm and base), and force magnitudes, which vary between $22 - 165\,\text{N}$ for the base, and $10 - 55\,\text{N}$ for the arm.

We compare the methods based on *absolute error*

$$e = \left| \left(\frac{\left|\hat{\boldsymbol{F}}^{*}_{\text{ext}}\right|}{\left|\boldsymbol{F}^{*}_{\text{ext}}\right|} - 1 \right) \cdot 100\% \right|, \tag{18}$$

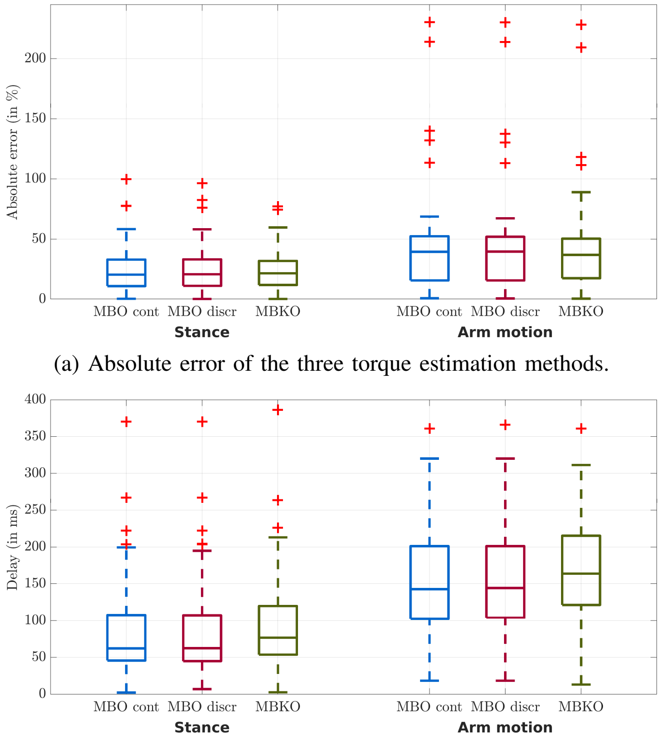
where $|\hat{\boldsymbol{F}}^{*}_{\text{ext}}|$ and $|\boldsymbol{F}^{*}_{\text{ext}}|$ indicate the magnitude peak values of the estimated and ground-truth F/T sensor collision force, respectively. We point out that the experiment under consideration is carried out in a static configuration; in such a scenario, the disturbance effect due to the static friction in the actuators is not negligible and is reflected in a constant offset of about $7\,\mathrm{N}$, which we highlighted in Fig. 4. Without loss of generality, the absolute error is computed by subtracting this offset from the force estimated by each of the method.

Another comparison criterion is the *delay*, which we calculate with a time-domain BPF and a constant threshold. Since in this torque and force estimation step we do not care for eliminating FPs yet, which is mainly important during detection, the thresholds are set to low values. It, therefore, does not indicate the delay of a final detection scheme, but it does indicate how well the estimators perform relative to each other. Finally, the *noise* in estimation when no collision is applied yet, is defined as the standard deviation in force $\hat{\boldsymbol{F}}_{\text{ext}}$.

a) Results: A comparison of the torque estimators is visualized in the boxplots in Fig. 6. Note that for all boxplots, the $+$ sign denotes the outliers. Looking at the absolute error, we notice that the three estimators give similar results. Considering both stance and arm motion, the MBKO has an average error of $29\,\%$ compared to $30\,\%$ for the two MBOs, which is a negligible difference. The MBKO shows higher noise immunity since its noise is $0.07\,\mathrm{N}$ compared to $0.18\,\mathrm{N}$ (MBOs). We also notice this in Fig. 7 when zooming in on the estimated forces before collision. Next, we look at the delay in Fig. 6b, which is slightly larger in the case of the MBKO. The average over both stance and motion is increased from $101\,\mathrm{ms}$ (MBOs) to $113\,\mathrm{ms}$ (MBKO). Finally, the most significant difference is in computational costs (see Table II), which is of order 10^3 larger for the MBKO compared to the other two. Note that this value might differ depending on computer parameters like CPU core and RAM.

b) Discussion: The similar results the two MBOs give is contrary to what is stated in [35]. In that work, it is demonstrated that the discrete-time MBO gives a more accurate estimation than its continuous-time implementation during high dynamic motion. The discretization error mentioned in [35], for which the discrete-time MBO is a solution, generally appears for low sampling frequencies or fast-changing dynamics of the estimated torques [52]. Both [35] and our work use acceptably high sampling frequencies, $1000\,\mathrm{Hz}$ and $400\,\mathrm{Hz}$, respectively. Therefore, it is most probable that the dynamics cause the difference in estimation between continuous- and discrete-time implementation. These fast-changing dynamics can result from the motion of a swing leg, in the case of [35], which is most likely higher in velocity than arm movement, in the case of the legged manipulator.

Comparing the MBOs to the MBKO, we note that the performances are heavily dependent on the tuning, which is more complex for the Kalman filter matrices. Deciding what observer to use for torque estimation, therefore, depends on the tuning technique. Furthermore, we recommend basing the choice on the type of robot since a high-DoF legged robot can increase computation time and tuning complexity for the MBKO significantly. On the other hand, if lower measurement and noise characteristics are preferred over phase delay, the MBKO might be a good choice. Since the two MBOs show similar results, we select the continuous-time MBO for comparing the collision detection and isolation methods in the subsequent sections.



(a) Absolute error of the three torque estimation methods.

(b) Delay of the three torque estimation methods.

Fig. 6: The absolute error and delay of the estimated external force using the three torque estimation methods (MBO continuous, MBO discrete, MBKO) is visualized in boxplots.

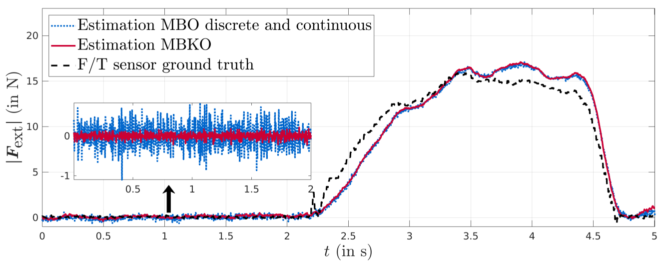


Fig. 7: A collision applied on the arm of the robot (1 in Fig. 1). The continuous- and discrete-time MBOs estimation are so similar, that no clear distinction is visible in the graph. We note that the offset due to modeling errors, as shown in Fig. 4, is subtracted.

B. Force filtering comparison

In the following experiment, we collect data with sampling time $T_{\text{s}} = 2.5\,\mathrm{ms}$. The desired frequency resolution for filter

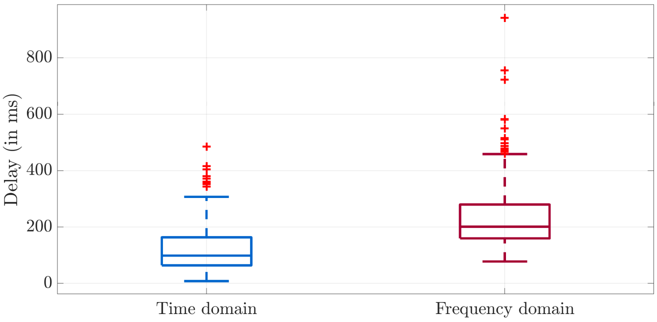


Fig. 8: The boxplots indicate the detection delay of a force filtered in the time or frequency domain over collisions during stance and arm motion.

Eq. (13) is set around $\Delta\omega = 0.8\,\text{Hz}$, resulting in a window in Eq. (12) of $N_{\text{win}} = 512$ with time resolution $\Delta t = 1.3\,\text{s}$. Filtering in frequency domain allows cut-off frequencies in steps of $\Delta\omega$ and therefore, after experimental tuning, the BPF is set as $[\omega_{\min}, \omega_{\max}] = [0.8, 3.2]$ Hz. To make a fair comparison, we set the cut-off frequencies of the time-domain filter equally.

To tune constant thresholds for the three directions of the forces filtered in time and frequency domain, we evaluate the 17 collisions in Table I during arm motion. Thresholds are set such that no FPs and FNs occur and are set to the lowest values possible. Since we consider arm motion, which makes the force estimation noisy, the tuned thresholds should be robust against FPs, meaning there is no need to adjust these in other scenarios. As validation of the methods, we analyze the 109 arm and base collisions at standstill and 23 base collisions during arm motion.

We compare the methods, again, based on delay, over both the tuning and validation sets. Additionally, we define the *success rate* and the *precision* over N collisions as $(N - N_{\text{FN}})/N$ and $N/(N + N_{\text{FP}})$, respectively, where N_{FN} is the number of FNs, and N_{FP} is the number of FPs. Here, N is the number of collisions from the validation set. Finally, we compare the schemes by looking at the noise in filtered force $\hat{\boldsymbol{F}}'_{\text{ext}}$.

a) Results: In Fig. 8, it is clear that time-domain filtering gives a lower detection delay. Computing the average gives $127\,\text{ms}$ compared to $246\,\text{ms}$ in frequency domain. Overall, we achieve a success rate of $99\,\%$ and a precision of $95\,\%$ for the time-domain filter, and $100\,\%, 96\,\%$ (success rate, precision) for frequency domain filtering. Finally, the noise is reduced with $21\,\%$ using the frequency compared to the time domain approach.

b) Discussion: Contrary to the experimental results in [40], which show a reduced detection delay using a frequency-domain compared to a time-domain filter, we demonstrate an improvement in detection time with time-domain filtering. Our implementation is based on a sampling time of $T_{\text{s}} = 2.5\,\text{ms}$ compared to $T_{\text{s}} = 1\,\text{ms}$ in [40]. It is implied that for a small enough sampling time, which results in a lower time resolution Δt and thus less delay, the frequency-domain approach is advantageous. In [40], the authors also state that frequency-domain filtering results in noise reduction, which is confirmed by our results. The difference in success rate and precision between the two methods is negligible, and both show the ability to detect accurately. Next, we look at the computational costs in Table II and note that the computation time for the time-domain BPF is lowest, although the FFT is significantly faster than when the algorithm is implemented as a DFT. It is suggested in [40] to implement the filter on one direction of the force only or on its magnitude, rather than on all three directions x, y and z, to increase computation speed. Nevertheless, a limitation of a legged robot is that the algorithm has to be implemented on all its DoFs making computation expensive. To conclude, we recommend using the frequency-domain approach only on systems with small sampling time ($T_{\text{s}} \leq 1\,\text{ms}$). Additionally, one should take into account the processing power of the implementation to make sure computation time does not rise significantly.

C. Threshold comparison

Similarly to the force filtering comparison, to compare the three thresholds from Section IV-B, we tune the thresholds based on the 17 collisions in Table I during arm motion. Again, the threshold performance is verified by looking at the stance and arm motion collisions. Additionally, we look at the 99 collisions in stance with an unmodeled load on the arm or base. Since the inertia of the robot increases due to the added load, we expect a more significant response to a collision, and it is interesting to see how the dynamic thresholds perform in this scenario. We filter the force with a time-domain BPF.

The comparison criteria are delay, success rate, precision, and *force rise*, defined as $(|\boldsymbol{F}_{\text{ext}}(t_{\text{det}})|/|\boldsymbol{F}^*_{\text{ext}}|) \cdot 100\,\%$. The latter indicates how much percentage of the ground truth peak value the collision force has reached at time of detection t_{det}.

a) Results: The delay in the three scenarios is found in Fig. 9. During stance without payload, the dynamic threshold based on velocity shows faster detection than the constant: the average delay is decreased from $143\,\text{ms}$ to $126\,\text{ms}$. With load, from $154\,\text{ms}$ to $142\,\text{ms}$. Although there is a difference of $17\,\text{ms}$ and $12\,\text{ms}$ in detection time, the collision force magnitude increases only $3\,\%$ during this time, in both cases. On the other hand, during arm motion, the constant detects faster than the velocity threshold. This is as we expect, because the desired velocities on which the velocity threshold is based are high when the arm is moving. Finally, we notice that the difference in delay between the constant and standard deviation techniques is small.

The success rates for the three approaches are high, $100\,\%$ for constant and dynamic based on velocity, and $99\,\%$ for the standard deviation threshold. This also holds for the precision which is $96\,\%$, $94\,\%$ and $95\,\%$ for the constant, velocity, and standard deviation schemes, respectively.

The three thresholds are visualized in Fig. 10. Note that the combination of a BPF force and a dynamic threshold demonstrates robustness against the unmodeled load placed on the arm and the arm motion. Before $t = 16.5\,\text{s}$, the arm

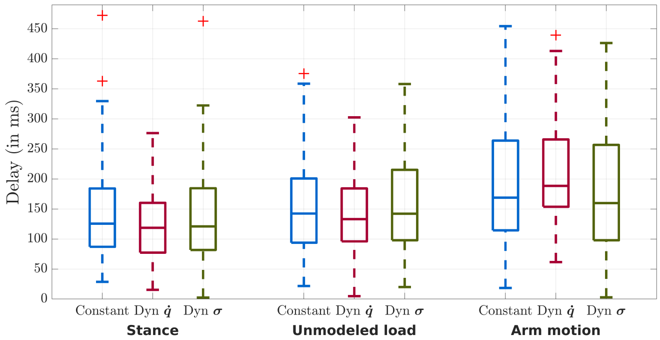
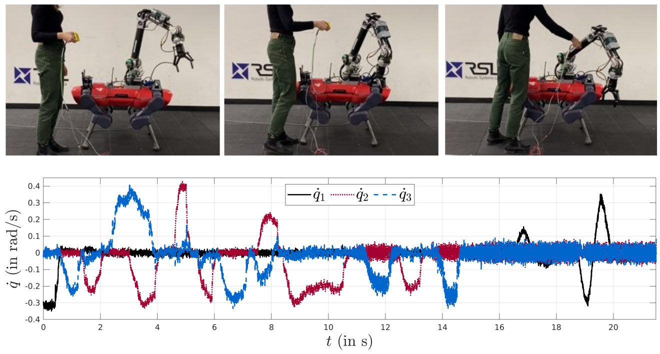
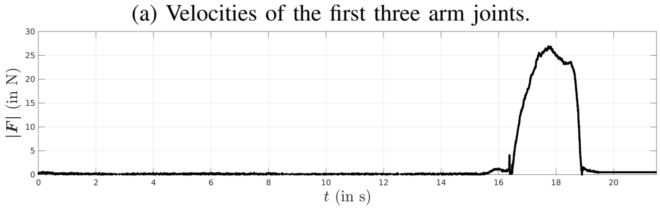


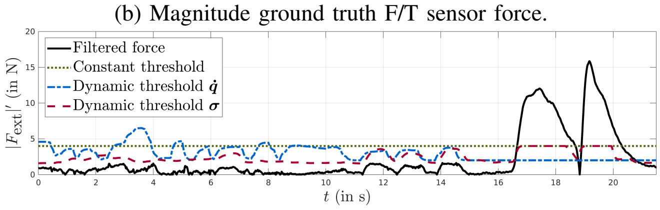
Fig. 9: The boxplots indicate the detection delay of the constant threshold, the dynamic threshold based on velocity 'Dyn $\dot{\boldsymbol{q}}$' and the dynamic threshold based on standard deviation 'Dyn $\boldsymbol{\sigma}$'. We apply collisions during stance, stance with unmodeled load on the arm and arm motion.



(a) Velocities of the first three arm joints.



(b) Magnitude ground truth F/T sensor force.



(c) Filtered force in x-direction with the three thresholds.

Fig. 10: Comparison of constant and dynamic thresholds. The arm is moving until the collision at $t = 16.5\,\mathrm{s}$. An unmodeled load is placed on the forearm.

is moving, and no FPs occur. The concept of the dynamic threshold based on velocity is visible during the collision, where $\dot{\boldsymbol{q}}_{\mathrm{des}} = \mathbf{0}$, and thus the threshold changes into a constant one. On the other hand, the standard deviation threshold increases during the collision since the growth in force results in a larger σ. However, it increases with a delay compared to the force. This delay is caused by the sliding window over which the standard deviation is computed. Furthermore, because of the F_σ value in Eq. (16), the increase in threshold is limited.

b) Discussion: In Table II, we notice that the tuning complexity of the dynamic thresholds is higher than for the constant. The question arises if the small gain in detection time of the velocity over the constant and standard deviation thresholds is worth the additional time needed for parameter tuning. We specifically point out that the external collision force has only increased $3\,\%$ in this period. However, if the reader is aiming to have fast detection in the scenario of human-robot collaboration, to prevent harm to the human, then a slight increase in detection time can be crucial.

D. Collision isolation

The third step of the flowchart in Fig. 2 is the localization of the impact force on the colliding body link. First, we evaluate all 56 base collisions from Table I and select a threshold such that none of the forces applied to the base get detected on the filtered arm torques $\hat{\boldsymbol{\tau}}'_{\mathrm{ext}}$. Next, we count for how many of the 360 arm collisions the arm joint torques exceed their threshold, and achieve a success rate of $98\,\%$.

To test the validity of the approach in Section V-A further, we evaluate the 360 arm collisions. We make a distinction between an impact force occurring on the forearm or upperarm. In Fig. 3, we see that the joints in the kinematic chain prior to the upperarm are arm joints 1 and 2, and 1 to 4 for the forearm. The threshold for the filtered arm torques is set such that all forearm collisions get identified as such during standstill, meaning that either $\hat{\tau}'_{\mathrm{ext},3} > b_3$ or $\hat{\tau}'_{\mathrm{ext},4} > b_4$. The success rate is then determined by counting how many of the upperarm collisions correctly get noticed on this body link, which means $\hat{\tau}'_{\mathrm{ext},3} < b_3$ and $\hat{\tau}'_{\mathrm{ext},4} < b_4$. We do not only analyze the collisions in Table I but also 9 impacts with a human arm, which is an interesting addition due to the different stiffness of the arm compared to the F/T sensor [40].

The results are shown in Fig. 11. A few comments reasoning why certain upperarm collisions are observed on the forearm:

- **Controller gain.** During our experiments, we use a whole-body Model Predictive Control (MPC) or a Proportional Integral Derivative (PID) controller to generate motion. Generally, upperarm pushes in stance get identified correctly. It is most striking, however, that when a whole-body MPC controller that controls the motion of the base was used, only 1 out of 5 of the upperarm collisions was localized accurately. This controller keeps the arm stiff by setting high proportional gains. Consequently, the static friction is high resulting in large joint torque values for all arm joint torques, independent of the contact location along the arm.
- **Load.** If we exclude the MPC controller observations from analysis, we note that during stance, the success rate is slightly lower in presence of an unmodeled load. The load increases the inertia of certain body links, making the response of some torques higher.

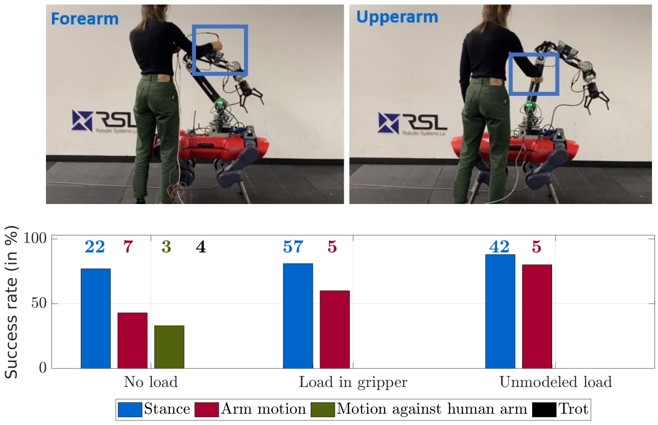


Fig. 11: A bar graph indicating the success rate of collisions correctly isolated at the upperarm. The numerical value above each box defines the total number of collisions on the upperarm for that specific scenario. Note that the success rate during trotting is $0\,\%$.

- **Motion.** During motion, the success rate is significantly lower. Arm movement causes arm torques $\hat{\tau}'_{\mathrm{ext},3}, \hat{\tau}'_{\mathrm{ext},4}$ to increase, even if no collision occurs yet. Depending on which arm joint is in motion at time of contact, an upperarm collision gets detected correctly or not. This dependency can be noticed in the different success rates for the scenarios 'no load', 'load in gripper', and 'unmodeled load' since all these datasets contain arbitrarily generated arm movements.
- **Trot.** During trot, the arm is vibrating and the success rate is $0\,\%$.
- **Configuration, contact point, and direction of the force.** If the arm is in a singular configuration or if the force is applied in a direction close to parallel to the contact link, part of the force will be transferred into the mechanical structure and will not be noticed in the torques. When analyzing the collisions, we note that sometimes, both upperarm and forearm forces are detected at all ($[\hat{\tau}'_{\mathrm{ext},1} \cdots \hat{\tau}'_{\mathrm{ext},4}]^T < \boldsymbol{b}$) and thus no distinction can be made.

To conclude, it is reasonable to assume that the contact link cannot always be isolated accurately and certain assumptions have to be made for this method to work. Thresholds could be set higher to get more accurate isolation during motion, but this comes at the cost of some forearm collisions not being detected.

VII. Conclusions and Recommendations

This paper has successfully extended the collision pipeline reviews for fixed-base manipulators in [3], [4] to a survey outlining collision handling methods for legged robots. Moreover, we have validated the torque estimation, collision detection, and isolation methods on hardware. This work is meant as a guideline when tackling the collision handling problem for legged robots. Based on our experiments, we conclude and recommend the following when choosing methods for each step in the collision pipeline:

1) **Torque estimation.** Experiments with the existing momentum-based observers on our 24-DoF legged manipulator show that the MBKO decreases the noise in force estimation with $0.1\,\mathrm{N}$. However, it is computationally more expensive than the conventional MBO, up to order 10^3, and it increases delay with $12\,\mathrm{ms}$. The computational complexity depends on the robot's number of DoF. A choice between these observers is thus robot dependent;
2) **Collision detection.** Detection methods robust against model inaccuracies, unmodeled payload, and dynamic motion consider band-pass filtering the estimated force. A BPF in the time domain gives a reduction of $119\,\mathrm{ms}$ detection delay compared to a frequency-domain filter using a sampling time of $T_{\mathrm{s}} = 2.5\,\mathrm{ms}$. Since the performance of frequency-domain filtering depends heavily on the time and frequency resolutions, we recommend using this approach only when running the collision pipeline with a sampling time $T_{\mathrm{s}} < 1\,\mathrm{ms}$. Another detection approach is a model-adaptive threshold, which we do not advise for a legged robot due to its high computation time and trajectory-dependency. The use of a dynamic threshold based on velocity requires more tuning than the constant threshold but has a reduction in detection delay of around $14\,\mathrm{ms}$;
3) **Collision isolation.** Although many works claim to be able to localize the colliding body link by looking at external torque components $\hat{\boldsymbol{\tau}}_{\mathrm{ext}}$, this only holds making certain assumptions. In certain scenarios (e.g., when the external force is parallel to the axis of a joint, singular configuration, high proportional controller gains, during motion), the effect of the collision force is not observable from the estimated filtered torques $\hat{\boldsymbol{\tau}}'_{\mathrm{ext}}$. In these cases, the colliding body link cannot be identified.

For future review studies, we recommend looking into the improvement of the force estimate using offline robot model identification before the torque estimation step. This process has been described extensively for fixed-base manipulators [4], but a roadmap for legged robots is still missing. Furthermore, method comparison for the last two steps in the collision handling pipeline, classifying an external force and reacting accordingly, would serve as an addition to this work.

References

[1] J. Fryman and B. Matthias, "Safety of industrial robots: from conventional to collaborative applications," in *ROBOTIK: German Conference on Robotics*, pp. 1–5, 2012.

[2] S. Mahmood, K. O. Ampadu, K. Antonopoulos, C. Panagiotou, S. A. P. Mendez, A. Podlubne, C. Antonopoulos, G. Keramidas, M. Hübner, D. Goehringer, and N. Voros, "Prospects of robots in assisted living environment," *Electronics*, vol. 10, no. 17, pp. 1–22, 2021.

[3] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot collisions: a survey on detection, isolation, and identification," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292–1312, 2017.

[4] S. Mamedov and S. Mikhel, "Practical aspects of model-based collision detection," *Frontiers in Robotics and AI*, vol. 7, no. 1, pp. 1–15, 2020.

[5] Y. Li, Y. Li, M. Zhu, Z. Xu, and D. Mu, "A nonlinear momentum observer for sensorless robot collision detection under model uncertainties," *Mechatronics*, vol. 78, no. 1, pp. 1–16, 2021.

[6] F. Flacco, T. Kroeger, A. De Luca, and O. Khatib, "A depth space approach for evaluating distance to objects: with application to human-robot collision avoidance," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 80, no. 1, pp. 7–22, 2015.

[7] H. Baltzakis, A. Argyros, and P. Trahanias, "Fusion of laser and visual data for robot motion planning and collision avoidance," *Machine Vision and Applications*, vol. 15, no. 2, pp. 92–100, 2003.

[8] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*, vol. 1. London: Springer, 2010.

[9] C. D. Bellicoso, K. Kramer, M. Stauble, D. Sako, F. Jenelten, M. Bjelonic, and M. Hutter, "ALMA - Articulated locomotion and manipulation for a torque-controllable robot," in *IEEE International Conference on Robotics and Automation*, pp. 8477–8483, 2019.

[10] V. Morlando, A. Teimoorzadeh, and F. Ruggiero, "Whole-body control with disturbance rejection through a momentum-based observer for quadruped robots," *Mechanism and Machine Theory*, vol. 164, no. 2, pp. 1–20, 2021.

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

[12] J. P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, "A unified MPC framework for whole-body dynamic locomotion and manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4688–4695, 2021.

[13] S. Faraji and A. J. IJspeert, "Designing a virtual whole body tactile sensor suit for a simulated humanoid robot using inverse dynamics," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 5564–5571, 2016.

[14] J. Vorndamme, M. Schappler, and S. Haddadin, "Collision detection, isolation and identification for humanoids," in *IEEE International Conference on Robotics and Automation*, pp. 4754–4761, 2017.

[15] T. Mattioli and M. Vendittelli, "Interaction force reconstruction for humanoid robots," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 282–289, 2017.

[16] L. Hawley, R. Rahem, and W. Suleiman, "External force observer for small- and medium-sized humanoid robots," *International Journal of Humanoid Robotics*, vol. 16, no. 6, pp. 1–25, 2019.

[17] F. Flacco, A. Paolillo, and A. Kheddar, "Residual-based contacts estimation for humanoid robots," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 409–415, 2016.

[18] A. Bolotnikoval, S. Courtois, and A. Kheddar, "Compliant robot motion regulated via proprioceptive sensor based contact observer," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 854–859, 2019.

[19] A. Bolotnikova, S. Courtois, and A. Kheddar, "Contact observer for humanoid robot Pepper based on tracking joint position discrepancies," in *IEEE International Symposium on Robot and Human Interactive Communication*, pp. 29–34, 2018.

[20] C. N. Cho and J. B. Song, "Collision detection algorithm robust to model uncertainty," *International Journal of Control, Automation and Systems*, vol. 11, no. 4, pp. 776–781, 2013.

[21] Z. Li, J. Ye, and H. Wu, "A virtual sensor for collision detection and distinction with conventional industrial robots," *Sensors*, vol. 19, no. 10, pp. 1–19, 2019.

[22] A. Caldas, M. Makarov, M. Grossard, P. Rodriguez-Ayerbe, and D. Dumur, "Adaptive residual filtering for safe human-robot collision detection under modeling uncertainties," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 722–727, 2013.

[23] P. Cao, Y. Gan, and X. Dai, "Model-based sensorless robot collision detection under model uncertainties with a fast dynamics identification," *International Journal of Advanced Robotic Systems*, vol. 16, no. 3, pp. 1–15, 2019.

[24] K. Masuya and K. Ayusawa, "A review of state estimation of humanoid robot targeting the center of mass, base kinematics, and external wrench," *Advanced Robotics*, vol. 34, no. 21-22, pp. 1380–1389, 2020.

[25] T. Ito, K. Ayusawa, E. Yoshida, and A. Kheddar, "Experimental study for controller-friendly contact estimation for humanoid robot," in *IEEE Workshop on Advanced Robotics and its Social Impacts*, vol. 2019, pp. 28–33, 2019.

[26] A. De Luca, A. Albu-Schäffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the DLR-III lightweight manipulator arm," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 1623–1630, 2006.

[27] S. He, J. Ye, Z. Li, S. Li, G. Wu, and H. Wu, "A momentum-based collision detection algorithm for industrial robots," in *IEEE International Conference on Robotics and Biomimetics*, pp. 1253–1259, 2015.

[28] T. Ren, Y. Dong, D. Wu, and K. Chen, "Collision detection and identification for robot manipulators based on extended state observer," *Control Engineering Practice*, vol. 79, no. 1, pp. 144–153, 2018.

[29] R. Priemer, *Fundamentals of digital signal processing*. CRC Press, 2018.

[30] A. Wahrburg, J. Bös, K. D. Listmann, F. Dai, B. Matthias, and H. Ding, "Motor-current-based estimation of Cartesian contact forces and torques for robotic manipulators and its application to force control," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 2, pp. 879–886, 2018.

[31] S. Shaik, J. Popat, and T. Kishore Kumar, "Kalman filter based phase delay reduction technique," in *International Conference on Recent Trends in Information Technology*, pp. 3–7, 2016.

[32] A. Wahrburg, E. Morara, G. Cesari, B. Matthias, and H. Ding, "Cartesian contact force estimation for robotic manipulators using Kalman filters and the generalized momentum," in *IEEE International Conference on Automation Science and Engineering*, pp. 1230–1235, 2015.

[33] J. Hu and R. Xiong, "Contact force estimation for robot manipulator using semiparametric model and disturbance Kalman filter," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 4, pp. 3365–3375, 2018.

[34] Z. Cong, A. Honglei, C. Wu, L. Lang, Q. Wei, and M. Hongxu, "Contact force estimation method of legged-robot and its application in impedance control," *IEEE Access*, vol. 8, pp. 161175–161187, 2020.

[35] G. Bledt, P. M. Wensing, S. Ingersoll, and S. Kim, "Contact model fusion for event-based locomotion in unstructured terrains," in *IEEE International Conference on Robotics and Automation*, pp. 4399–4406, 2018.

[36] M. Iskandar, O. Eiberger, A. Albu-Schaffer, A. De Luca, and A. Dietrich, "Collision detection, identification, and localization on the DLR SARA robot with sensing redundancy," in *IEEE International Conference on Robotics and Automation*, pp. 3111–3117, 2021.

[37] C. N. Cho, J. H. Kim, Y. L. Kim, J. B. Song, and J. H. Kyung, "Collision detection algorithm to distinguish between intended contact and unexpected collision," *Advanced Robotics*, vol. 26, no. 16, pp. 1825–1840, 2012.

[38] K. Manal and W. Rose, "A general solution for the time delay introduced by a low-pass Butterworth digital filter: an application to musculoskeletal modeling," *Journal of Biomechanics*, vol. 40, no. 3, pp. 678–681, 2007.

[39] A. Kouris, F. Dimeas, and N. Aspragathos, "Contact distinction in human-robot cooperation with admittance control," in *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1951–1956, 2016.

[40] A. Kouris, F. Dimeas, and N. Aspragathos, "A frequency domain approach for contact type distinction in human-robot collaboration," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 720–727, 2018.

[41] A. J. Roscoe and G. M. Burt, "Comparisons of the execution times and memory requirements for high-speed discrete Fourier transforms and fast Fourier transforms, for the measurement of AC power harmonics," in *International Symposium Metrological Infrastructure*, pp. 40–45, 2011.

[42] N. Solutions, "White paper: Understanding FFTs and windowing," tech. rep., NI Solutions, Austin, Texas, 2021.

[43] J. G. Marsar and S. Walker, "Improving FFT frequency resolution (course notes for FAMU-FSU EEL 4930)," 2015.

[44] N. Kehtarnavaz, *Digital Signal Processing System Design*. Academic Press, 2 ed., 2008.

[45] V. Sotoudehnejad and M. R. Kermani, "Velocity-based variable thresholds for improving collision detection in manipulators," in *IEEE International Conference on Robotics and Automation*, pp. 3364–3369, 2014.

[46] V. Sotoudehnejad, A. Takhmar, M. R. Kermani, and I. G. Polushin, "Counteracting modeling errors for sensitive observer-based manipulator collision detection," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 4315–4320, 2012.

[47] M. Guo, H. Zhang, C. Feng, M. Liu, and J. Huo, "Manipulator residual estimation and its application in collision detection," *Industrial Robot*, vol. 45, no. 3, pp. 354–362, 2018.

[48] A. De Luca and L. Ferrajoli, "Exploiting robot redundancy in collision detection and reaction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3299–3305, 2008.

[49] M. Geravand, F. Flacco, and A. De Luca, "Human-robot physical interaction and collaboration using an industrial robot with a closed

control architecture," in *IEEE International Conference on Robotics and Automation*, pp. 4000–4007, 2013.

[50] Z. Qiu, R. Ozawa, and S. Ma, "Adaptive virtual power-based collision detection and isolation with link parameter estimation," *Advanced Robotics*, vol. 34, no. 12, pp. 814–825, 2020.

[51] C. Escobedo, M. Strong, M. West, A. Aramburu, and A. Roncone, "Contact anticipation for physical human–robot interaction with robotic manipulators using onboard proximity sensors," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 7255–7262, 2021.

[52] Y. F. Wang and Y. W. Li, "Analysis and digital implementation of cascaded delayed-signal-cancellation PLL," *IEEE Transactions on Power Electronics*, vol. 26, no. 4, pp. 1067–1080, 2011.

Chapter 2

Paper 'Collision detection, isolation and identification for a legged manipulator'

Collision detection, isolation and identification for a legged manipulator

Jessie van Dam, Andreea Tulbure, Maria Vittoria Minniti, Firas Abi Farraj, Marco Hutter

Abstract— **To safely deploy legged robots in the real world, it is necessary to provide them with the ability to reliably detect unexpected contacts and accurately estimate the corresponding contact force. In this paper, we propose a collision detection, isolation and identification pipeline for a quadrupedal manipulator. We first introduce an approach to estimate the collision time span based on band-pass filtering and show that this information is key for obtaining accurate collision force estimates. We also propose an accurate collision identification method which is robust against model inaccuracies, unmodeled loads and any other potential source of quasi-static disturbances acting on the robot. We validate our framework using extensive hardware experiments in various scenarios summing up to 416 collisions, including trotting and additional unmodeled load on the robot.**

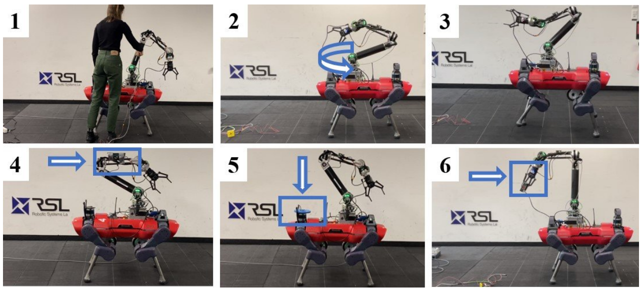


Fig. 1: The quadrupedal robot ANYmal with a 6-DoF manipulator mounted on top. We consider the following scenarios: (1) stance; (2) arm motion; (3) trotting; (4) unmodeled load on the forearm; (5) unmodeled load on the base; and (6) load in the gripper.

I. Introduction

Quadrupedal robots have recently become sufficiently advanced to be deployed in unknown and unstructured environments, where they could operate alongside humans or other robots. In such settings, unexpected collisions with the external environment (e.g. people or objects) are likely to occur and collision detection plays an important role in ensuring the safety of the external environment, and also for keeping the balance of the robot. Thus, robots need to be able to reliably detect such collisions, accurately estimate the corresponding contact forces, and react accordingly.

Literature refers to the framework for responding to collisions as the *collision event pipeline*, separating a collision event into five phases, i.e.: detection, isolation, identification, classification and reaction [1]. In this paper, we focus on the first three phases. Briefly, the collision detection phase defines when a collision happens based on the external estimated torques acting on the robot; the collision isolation phase identifies the location on which the collision is acting; and the collision identification phase estimates the external collision force in magnitude and direction.

Momentum-based collision event pipelines have proven to be successful in collision handling for fixed-based manipulators [1], [2]. However, especially for complex high-DoF robots, the employed robot model might be affected by potential accidents and modeling inaccuracies due to, for example, static friction, wear and tear. In addition, it might be difficult to model all kinds of payloads that the robot might need to carry during a manipulation task. Such elements of uncertainty can lead to failures of the collision detection phase (e.g., detecting a collision when there is none) or inaccurate external force estimation (e.g., considering the force coming from such disturbances as being part of the collision force). Therefore, in this work, we focus on the design of a collision detection and identification method that is robust against such disturbance factors. To achieve this, we recognize the importance of detecting the initial and final time when a collision occurs, and use this information to improve the collision force estimation during the identification phase.

A. Related work

Existing collision event pipelines for robotic systems are based on monitoring the generalized external torques acting on the robot. Many torque estimation methods have been proposed in the literature, e.g. direct estimation [3], static direct estimation [4], filtered dynamics observer [5] or momentum-based observers (MBO) [6], [7], [8], [9], [10]. For collision detection, such estimated torques are usually compared with appropriately-designed thresholds [11], [12]. Constant thresholds suffer from the problem of false positives (FPs) occurring when a robot task includes high-frequency content or false negatives (FNs) for light impact collisions. Another challenge in collision detection is the presence of modeling errors that can generate slow and configuration-dependent variations in the estimated torques. Dynamic thresholds based on velocity [13], [14] or standard deviation of the estimated force [15] have been proposed to cope with such a problem. An effective way to improve the robustness of detection in the presence of model uncertainties [16] or unknown loads [17] is to filter the estimated torques with

This work was supported in part by the Swiss National Science Foundation through the National Centre of Competence in Research Robotics (NCCR Robotics), in part by the Swiss National Science Foundation through the National Centre of Competence in Digital Fabrication (NCCR dfab) and in part by the Swiss National Science Foundation (SNSF) as part of project No.188596.

All authors are with the Robotic Systems Lab, ETH Zurich, Zurich 8092, Switzerland jessievandam@hotmail.com

a band-pass filter (BPF). BPF-based methods filter out the offset in the estimated external torques due to payloads or model inaccuracies, allowing to reduce the detection threshold by an order of 5 [16], and thus obtain faster detection. However, so far BPF-based works have only considered high impact contacts on a low-DoF manipulator and have focused on the detection rather than the identification phase.

The collision detection step is usually followed by a collision isolation phase [1], [18], [19] and a collision identification phase. Collision identification has been studied for fixed-base manipulators [1], and works have also considered wheeled humanoids [20], [21]. Legged robots, on the other hand, impose additional challenges, such as the high number of DoFs which increases the computational time and tuning complexity [18], the static instability during dynamic walking [8], and the necessity of reasoning about a large number of contacts [6]. Thus, much interest has recently been given to tackling the problem of force estimation and collision identification for legged robots [4], [6], [8], [18], [19], [22], [23]. In [4], contacts are applied while a NAO humanoid is at standstill or static equilibrium. Similarly, in [3], four different collisions between $25 - 30\,\mathrm{N}$ lasting $18\,\mathrm{s}$ are applied on a HRP-4 humanoid. The estimation accuracy lies within $5\,\mathrm{N}$, while the robot is standing still. Collision identification during walking with a NAO robot is addressed in [22]; the authors estimate the external collision forces using proprioceptive sensing in combination with an Inertial Measurement Unit (IMU) and force-sensing resistors beneath the feet. Contact wrench estimation for quadrupeds has mostly been restricted to estimating the forces at the feet in contact with the ground [7], [23], [24]. In [8], external forces are estimated at unknown contact points other than the feet; multiple case studies in which collisions are applied to the legs, both during stance and trotting, are validated in simulation.

However, all the described works do not consider the effect of model uncertainties or unmodeled loads on the estimated external forces. In [6], force/torque (F/T) sensors placed on the robot body are used to compensate for the weight of an unknown load; model inaccuracies are not addressed and the method is only validated in simulation.

B. Contributions

In this work, we propose a model-based collision detection, isolation, and identification framework for a quadrupedal manipulator, able to accurately estimate the collision time span and external collision forces in various scenarios, including unmodeled loads on the robot. The main contributions of this work are the following:

- A BPF-based approach for estimating the time span of applied collisions to improve collision identification.
- A robust collision identification approach, based on continuous disturbance force estimation, which can compensate for unmodeled loads and model inaccuracies.
- Extensive experimental validation of the proposed framework on a quadrupedal manipulator in various scenarios, including a comparison of the proposed method against existing collision identification methods for legged robots.

The remainder of the paper is structured as follows. We first discuss needed preliminaries in Sec. II. In Sec. III we introduce our collision detection, isolation and identification methods. We present experimental results, highlighting the performance of our proposed framework in Sec. IV. Sec. V concludes this work.

II. Preliminaries

In this section, we proceed by presenting the model of a floating-base manipulator in Sec. II-A. Later, we give the fundamentals of the first block of the collision pipeline (see Fig. 2), which includes the estimation of the external generalized torques (Sec. II-B) and the external wrench (Sec. II-C).

Throughout the paper, we use the symbol $\hat{a}$ to denote the estimate of a variable a. In addition, the 2-norm of a vector $\boldsymbol{a}$ is denoted as $|\boldsymbol{a}|$, while the absolute value of one of the components of the vector is denoted as $|a|$.

A. Floating-base dynamic model

The equations of motion of a floating-base robot are given by [19], [25]

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{q},\boldsymbol{v})\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{S}^T\boldsymbol{\tau}_{\mathrm{m}} + \boldsymbol{\tau}_{\mathrm{ext}} + \boldsymbol{\tau}_{\mathrm{ft}}, \quad (1)$$

where $\boldsymbol{q},\boldsymbol{v} \in \mathbb{R}^{6+n}$ are the robot generalized coordinates and velocities, respectively, and n is the number of actuated joints. $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{(6+n)\times(6+n)}$ is the inertia matrix, $\boldsymbol{C}(\boldsymbol{q},\boldsymbol{v}) \in \mathbb{R}^{(6+n)\times(6+n)}$ represent the Coriolis terms while $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^{6+n}$ is the vector of gravitational terms. The joint torques $\boldsymbol{\tau}_{\mathrm{m}} \in \mathbb{R}^n$ are mapped into the $6+n$ dimensional space of generalized velocities by the transpose of actuator-selection matrix $\mathbf{S} = \begin{bmatrix}\mathbf{0}_{n_\mathrm{j}\times 6} & \mathbf{I}_{n_\mathrm{j}\times n_\mathrm{j}}\end{bmatrix}$. In this work, we divide the external torques acting on the robot into two components. The first one is assumed to be directly measured using a F/T sensor and is denoted $\boldsymbol{\tau}_{\mathrm{ft}}$. The second one is denoted as $\boldsymbol{\tau}_{\mathrm{ext}}$ and is due to external collision forces that are not measurable with a F/T sensor, and disturbance factors such as modeling inaccuracies, unmodeled payloads, and sensor noise.

B. Torque estimation

To estimate the external torques acting on the robot joints, $\boldsymbol{\tau}_{\mathrm{ext}}$, we use a momentum-based observer approach [12], which is based on the definition of the robot generalized momentum $\boldsymbol{p} = \boldsymbol{M}(\boldsymbol{q})\boldsymbol{v}$. Let $\boldsymbol{n}(\boldsymbol{q},\boldsymbol{v}) := \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{C}(\boldsymbol{q},\boldsymbol{v})\boldsymbol{v} - \dot{\boldsymbol{M}}(\boldsymbol{q},\boldsymbol{v})\boldsymbol{v}$. Based on [12], $\hat{\boldsymbol{\tau}}_{\mathrm{ext}}$ can be computed as

$$\hat{\boldsymbol{\tau}}_{\mathrm{ext}} = \boldsymbol{K}_{\mathrm{O}}\left(\boldsymbol{p}(t) - \int_0^t \boldsymbol{S}^T\boldsymbol{\tau}_{\mathrm{m}} + \boldsymbol{\tau}_{\mathrm{ft}} - \hat{\boldsymbol{n}}(\boldsymbol{q},\boldsymbol{v}) + \hat{\boldsymbol{\tau}}_{\mathrm{ext}}\mathrm{d}s\right), \quad (2)$$

where $\boldsymbol{K}_{\mathrm{O}}$ is a positive diagonal observer gain matrix, and $\hat{\boldsymbol{n}}$ is the estimate of the nonlinear terms $\boldsymbol{n}$, obtained from the rigid body dynamics equations of the robot.

We point out that multiple variants of MBOs have been presented in the literature (e.g. discrete-time MBO [7] and Kalman-based (MBKO) [9]). A more detailed analysis of each of them, as well as the reasoning behind the one used for torque estimation in our framework, is given in Sec. IV-A.

C. Wrench estimation

Let $\boldsymbol{\mathcal{F}}_{\text{ext},i} = (\boldsymbol{F}_{\text{ext},i}, \boldsymbol{m}_{\text{ext},i}) \in \mathbb{R}^6$ be the wrench applied to the colliding link i, consisting of a force and torque component. The force $\boldsymbol{F}_{\text{ext},i}$ is given by $\boldsymbol{F}_{\text{ext},i} = \boldsymbol{F}_{\text{c},i} + \boldsymbol{F}_{\text{dis},i}$, where $\boldsymbol{F}_{\text{c},i}$ is the external collision force, and $\boldsymbol{F}_{\text{dis},i}$ is due to all other disturbance sources. Note that besides detecting the beginning and the end of a collision, the final goal of this paper is to prune $\boldsymbol{F}_{\text{ext},i}$ from its disturbance component and to obtain an accurate estimate of the collision force $\boldsymbol{F}_{\text{c},i}$.

Once $\hat{\boldsymbol{\tau}}_{\text{ext}}$ is obtained from Eq. (2), the external forces and wrench are estimated as

$$\begin{bmatrix} \hat{\boldsymbol{F}}_{\text{f},1} \\ \vdots \\ \hat{\boldsymbol{F}}_{\text{f},4} \\ \hat{\boldsymbol{\mathcal{F}}}_{\text{ext},i} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_{\text{f},1}^T(\boldsymbol{q}) & \dots & \boldsymbol{J}_{\text{f},4}^T(\boldsymbol{q}) & \boldsymbol{J}_i^T(\boldsymbol{q}) \end{bmatrix}^{\#} \hat{\boldsymbol{\tau}}_{\text{ext}}, \quad (3)$$

where the symbol $^{\#}$ denotes the Moore-Penrose pseudoinverse operation, $\boldsymbol{F}_{\text{f},j}$ is the force on contact foot j with $\boldsymbol{J}_{\text{f},j} \in \mathbb{R}^{3\times(6+n)}$ its translational Jacobian, and $\boldsymbol{J}_i \in \mathbb{R}^{6\times(6+n)}$ is the spatial Jacobian of the colliding point i [6], [19]. Here, the contact feet and the collision links are modeled as point contacts and thus are only subject to linear forces. However, since the exact contact location of the collision force is unknown, the spatial Jacobian of an arbitrary robot link is used in Eq (3), which results in the additional torque $\boldsymbol{m}_{\text{ext},i}$ caused by the impact force. In the following sections, we neglect such torque component and only use $\hat{\boldsymbol{F}}_{\text{ext},i}$.

In this work, we are interested in monitoring collisions that could happen during manipulation or human-robot collaboration tasks. In such scenarios, it is reasonable to assume that collisions with the legs are less likely to occur than collisions with the base or the arm of the robot. Thus, we exclude the former type of collisions from our analysis and assume that all forces on the legs are point forces acting on the feet.

III. Collision detection, isolation and identification

As the fundamentals of torque and wrench estimation are explained in Sec.II, in this section we describe the remaining components of the pipeline: detection, isolation, and identification. A scheme of the pipeline is illustrated in Fig. 2 and can be summarized in: (1) observing the external torque $\hat{\boldsymbol{\tau}}_{\text{ext}}$ and estimating external force $\hat{\boldsymbol{F}}_{\text{ext}}$; (2) deciding if a collision occurs ($\epsilon = 1$ or $\epsilon = 0$) by filtering the force and comparing it to a threshold; (3) determining if the arm or base is the colliding body part; and (4) isolating the disturbance from the force estimate to identify the magnitude and direction of the force using our identification method. With the final estimated contact force $\hat{\boldsymbol{F}}_{\text{c}}$, a reaction strategy can be implemented.

A. Collision detection

In existing works [16], [17], it is common to detect collisions by observing when the external generalized torques $\hat{\boldsymbol{\tau}}_{\text{ext}}$ cross some specific thresholds. Although such an approach can lead to accurate detection for a fixed-base robot, in the case of a high-DoF quadrupedal manipulator, the definition of the detection thresholds becomes more complex if done in the high-dimensional space of the generalized torques. Furthermore, an arm collision could cause the vector $\boldsymbol{\tau}_{\text{ext}}$ to have larger values along the base components compared to the arm components. On the other hand, since the arm and base forces are computed from Eq. (3), they capture variations in all the components of $\boldsymbol{\tau}_{\text{ext}}$. Thus, in this work, the estimated forces $\hat{\boldsymbol{F}}_{\text{ext,arm}}$ and $\hat{\boldsymbol{F}}_{\text{ext,base}}$ are used as inputs to the collision detection block (Fig. 2).

1) Force filtering: During collision detection, we first filter $\hat{\boldsymbol{F}}_{\text{ext,arm}}$ and $\hat{\boldsymbol{F}}_{\text{ext,base}}$ with a band-pass filter. We indicate the resulting filtered forces with $\hat{\boldsymbol{F}}'_{\text{ext,arm}}$ or $\hat{\boldsymbol{F}}'_{\text{ext,arm}}$. While it has been proven that the use of a high-order BPF can increase the overall robustness of the detection [16], [17], it increases the time delay [26] compared to a first-order BPF. We therefore opt for the latter in our experiments.

2) Collision detection and time span estimation: During a collision, the high-pass property of the BPF causes two peaks on the filtered forces, one at the start and one at the end of the impact. These peaks are the result of the high frequencies due to the sudden force change when the contact is applied and removed. This phenomenon is depicted Fig. 3 (top-right plot), where the ground truth collision force (top-left plot) is filtered with a BPF. In the proposed method, we employ this two-peak BPF phenomenon to detect not only the beginning but also the end of the collision.

Briefly, a collision is detected when the band-pass filtered force crosses the chosen threshold b on either the positive ($+b$) or the negative ($-b$) half-plane. The end of the collision is then detected at the beginning of the second peak, i.e. when the signal crosses the threshold on the opposite half. This criterion is applied to each component of the band-pass filtered force. The time of detection is visualized in Fig. 3 (lower plot). The detection of the beginning and the end of the collision using the band-pass filtered force, rather than the estimate of the force itself, is of particular importance as it is very robust against external disturbances. It also allows for an accurate estimation of the time span of the collision.

The output of the collision detection is a variable ϵ, defined as:

$$\epsilon = \begin{cases} 1, & \text{if in collision} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

B. Collision isolation

A common approach to isolate the colliding body link is based on the assumption that, when a collision occurs at contact link i in an open kinematic chain structure, this contact does not produce torques along the joints more distal from the link in contact [1], [3], [6]. Consequently, the first link i for which $|\hat{\tau}_{\text{ext},i}| > b$, with b a threshold larger than

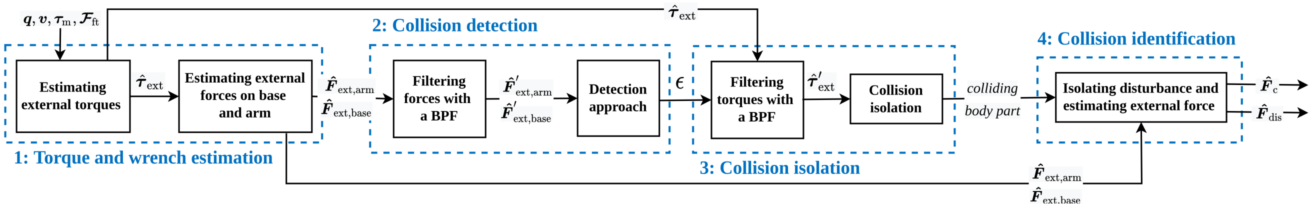


Fig. 2: Collision-event pipeline presented in this work. The pipeline consists of the following four phases: torque and wrench estimation (Sec. II-B, II-C), collision detection (Sec. III-A), collision isolation (Sec. III-B), and collision identification (Sec. III-C)

.

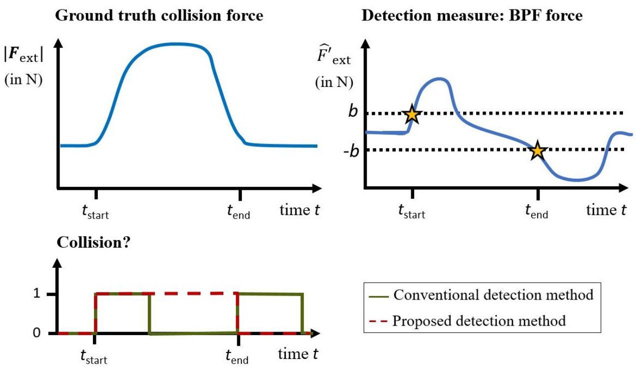


Fig. 3: The collision force *(upper left)* is estimated and filtered with a BPF *(upper right)*, where a constant threshold with value b is set for detection and the stars indicate the detected start and end of the collision with the proposed method. The conventional detection approach detects two collisions, while the proposed detection technique detects the time span accurately *(lower plot)*.

0 to account for model inaccuracies, is defined as the one on which the external force is applied. However, in certain scenarios (e.g., when the external force is parallel to the axis of a joint), the effect of the collision force is not observable from the generalized torques $\hat{\boldsymbol{\tau}}_{\text{ext}}$.

Even if the identification of the colliding body link is not possible in every scenario, we observed that the arm joint torques can be used to make a distinction between a collision occurring at the base or the arm with a sufficient degree of confidence as will be shown later in the results of the performed experiments. Therefore, here we use the following collision isolation rule: After a collision is detected, if one of the arm torques $\hat{\tau}'_{ext,i}$ crosses its threshold, we conclude that the collision is occurring at the arm. Otherwise, it is occurring at the base.

C. Collision identification

As pointed out in Sec.III-A, applying a BPF to the estimated external force is a solution to make the collision detection algorithm robust to model inaccuracies and unmodeled loads. However, this may alter the magnitude characteristics of the estimated force. Thus, the band-pass filtered forces cannot be used to identify the magnitude of the collision force.

Existing methods try to improve the force estimation accuracy by performing off-line model identification [2], [27]. However, this may be a complex and time-consuming process and errors will always remain [2]. Moreover, the identified model remains sensitive to changes in the robot configuration, wear and tear or accidents that might occur to the robot in a real-world environment.

Therefore, here we propose the following collision identification method. For brevity, we define:

$$\hat{\boldsymbol{F}}_{\text{ext}} = \begin{cases} \hat{\boldsymbol{F}}_{\text{ext,arm}}, & \text{if arm collision} \\ \hat{\boldsymbol{F}}_{\text{ext,base}}, & \text{if base collision} \end{cases} \tag{5}$$

as defined in Eq. (3). Let $\boldsymbol{F}_{\text{dis}}$ be the disturbance signal that includes modeling errors, unmodeled payloads and sensor noise. At time k, we compute:

$$\hat{\boldsymbol{F}}_{\text{dis}}(k) = \begin{cases} (1-\alpha)\,\hat{\boldsymbol{F}}_{\text{dis}}(k-1) + \alpha \hat{\boldsymbol{F}}_{\text{ext}}(k), & \text{if } \epsilon = 0 \\ \hat{\boldsymbol{F}}_{\text{dis}}(k_{\text{det}}), & \text{if } \epsilon = 1 \end{cases} \tag{6}$$

where $\alpha = e^{-\omega T_s}$, ω is the cut-off frequency of the low-pass filter and T_s is the sampling time. If a collision is detected ($\epsilon = 1$) at time k_{det}, the disturbance force is fixed, assuming it stays constant during the collision. At time k, the collision force $\hat{\boldsymbol{F}}_{\text{c}}(k)$ is computed by subtracting the disturbance signal from the estimated value, i.e.: $\hat{\boldsymbol{F}}_{\text{c}}(k) = \hat{\boldsymbol{F}}_{\text{ext}}(k) - \hat{\boldsymbol{F}}_{\text{dis}}(k)$. Finally, the direction of $\hat{\boldsymbol{F}}_{\text{c}}$ is set equal to the direction of $\hat{\boldsymbol{F}}_{\text{ext}}$, and its magnitude is defined as $|\hat{\boldsymbol{F}}_{\text{c}}|$.

IV. Experiments

We verify the proposed pipeline through hardware experiments on the quadrupedal robot ANYmal [1] with a 6-DoF arm mounted on top (Fig. 1). The robot is equipped with a RobotiQ 2F-85 gripper [2] and a BOTA Rokubi SensOne 6DoF F/T sensor [3] at its end-effector.

In total, in our experiments, we apply 416 collisions to the robot in various case studies visualized in Fig. 1. The collision distribution is presented in Table I. Examples of collisions in these scenarios, and visualization of our identification method's performance, are shown in the accompanying video: `https://youtu.be/ESrYyz6nicA`. Ground truth contact data is read from

[1] https://www.anybotics.com/
[2] https://robotiq.com/products/2f85-140-adaptive-robot-gripper
[3] https://www.botasys.com/

TABLE I: Collision distribution among the different scenarios.

Scenario		Number of collisions
Stance	**no load**	109
	load gripper	133
	unmodeled load	99
Arm motion	**no load**	40
	load gripper	10
	unmodeled load	13
Trotting	**no load**	12

a hand-held BOTA Rokubi Mini 6-DoF F/T sensor; the collisions are created by either pushing the F/T sensor on different parts of the arm and base of the robot or by holding it still while the arm collides with it. A joystick is used to send references to the robot.

In the following sections, we first provide some experimental analysis to explain the reasoning behind our choice of continuous-time MBO for torque estimation (Sec. IV-A) and constant thresholds for detection (Sec. IV-B). Afterwards, we validate the proposed collision-event pipeline in hardware tests (Sec. IV-C).

A. Comparison of torque estimation methods

To decide which of the existing torque-estimation methods is most suitable for our identification approach, we compare them in hardware experiments. We consider scenarios where collisions are applied on the arm, while the robot is in stance (Fig. 4). We consider the following methods:

- **Direct estimation** [3]: computes $\hat{\boldsymbol{\tau}}_{\text{ext}}$ directly from Eq. (1).
- **Static direct estimation** [4]: similar to [3], but neglects the acceleration and velocity-dependent terms.
- **Continuous-time MBO** [6]: implements Eq. (2).
- **Discrete-time MBO** [7]: discretized implementation of the MBO [6];
- **Third-order MBO** [8]: similar to the continuous-time MBO, but using a higher-order LPF. This results in more accurate estimations due to a sharper filtering action.
- **MBKO** [9]: reformulates the continuous-time MBO [6] as a Kalman filter.

Except for the MBKO [9], all these approaches have been used on legged robots in previous literature. A comparison was therefore essential to understand the differences between them and the impact of these differences in the scenario at hand.

We use the methods above to obtain an estimate of the generalized external torque $\boldsymbol{\tau}_{\text{ext}}$. Afterwards, we compute the estimated external force $\hat{\boldsymbol{F}}_{\text{ext,arm}}$ as explained in Sec.II-C. The results from our comparison, considering 40 collisions in stance and arm motion, are reported in Table II. In Fig. 4, the response of the selected methods to one of the collisions applied to the arm of the robot is visualized. In the table, we use *noise* to refer to the standard deviation over the time in between collisions. To compute the detection time, we band-pass filter the estimated forces and set a threshold, equal for each method. We then compute the *delay* of each method with respect to the fastest one. The *absolute error* is defined as:

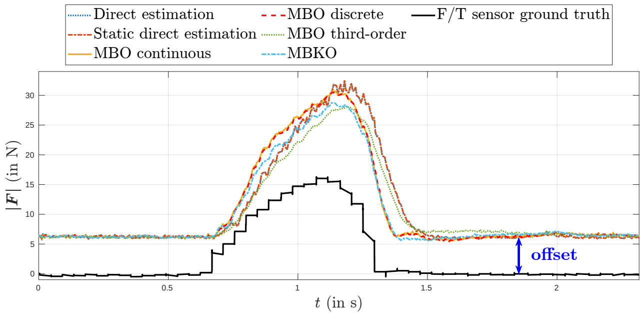


Fig. 4: One of the collisions applied on the arm of the robot during stance. Due to the static friction present in the actuators, an offset is visible between the estimated and the ground truth force.

TABLE II: Comparison of torque estimation methods during an experiment where a collision is applied on the arm of the robot.

	Tuning simplicity	Computation time	Noise (in N)	Delay (in ms)	Absolute error (in %)
Direct estimation	++	+	**0.72**	21	**37**
Static direct estimation	++	+	**0.72**	21	**37**
MBO continuous	+	+	0.30	**0**	34
MBO discrete	+	+	0.30	**2**	33
MBO third-order	−	+	0.32	**26**	**31**
MBKO	−	−−	**0.21**	18	**30**

$$e = \left| \left(\frac{\left|\hat{\boldsymbol{F}}_{\text{c}}^{*}\right|}{\left|\boldsymbol{F}_{\text{ext}}^{*}\right|} - 1 \right) \cdot 100\% \right|, \tag{7}$$

where $|\hat{\boldsymbol{F}}_{\text{c}}^{*}|$ and $|\boldsymbol{F}_{\text{ext}}^{*}|$ indicate the magnitude peak values of the estimated and ground-truth F/T sensor collision force, respectively.

We point out that the experiment under consideration is carried out in a static configuration; in such a scenario, the disturbance effect due to the static friction in the actuators is not negligible and is reflected in a constant offset of about 7 N, which we highlighted in Fig. 4. Without loss of generality, the absolute error reported in Table II is computed by subtracting this offset from the force estimated by each of the methods.

It is visible from Table II that the methods based on the direct estimation of $\boldsymbol{\tau}_{\text{ext}}$ have a high absolute error. In addition, the force estimated with such methods shows oscillations when the force estimate is at its peak value. The third-order MBO and MBKO have the highest estimation accuracy in terms of absolute error and noise, respectively. However, computation time is significantly higher for the

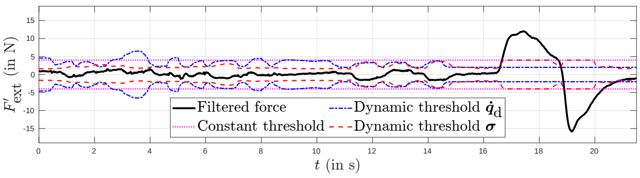


Fig. 5: Comparison of constant and dynamic thresholds. The dynamic threshold is based on the desired joint velocity $\dot{\boldsymbol{q}}_{\text{d}}$ and on estimated force standard deviation $\boldsymbol{\sigma}$. The arm is moving until the collision at $t = 16.5\,\text{s}$.

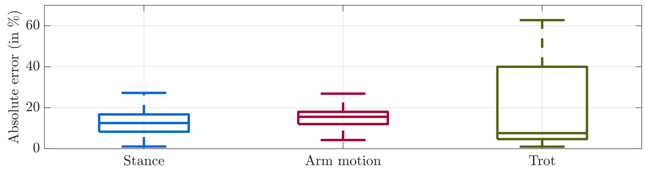


Fig. 6: Boxplots comparing the absolute estimation error of the collision time span.

MBKO, and the third-order MBO has a large delay. Furthermore, both methods have a higher tuning complexity, which is an important aspect to consider when working with a high-DoF quadrupedal manipulator. The higher-order filter could also possibly result in oscillations and instability [28].

Thus, we conclude that the continuous-time and discrete-time MBOs provide the best trade-off between delay and estimation accuracy. Since they show similar results, we select the continuous-time MBO for our experiments.

B. Threshold comparison

We compare constant and dynamic thresholds to assess how they influence the collision detection performance in Fig. 5. We consider a dynamic threshold based on joint velocities from [13] and a dynamic threshold based on estimated force standard deviation from [15] (although we do not include proximity sensors). The dynamic thresholds move along with the variations in the filtered force that arise due to model errors. For conciseness, in Fig. 5, we only plot the component of the estimated force along the main contact direction.

We found out that an average of $14\,\text{ms}$ improvement in detection delay is obtained using a dynamic threshold; this result was obtained in experiments that we conducted with the robot in stance, performing arm motion, and in the presence of an unmodeled payload. A plot from one of such experiments is reported in Fig. 5.

In fact, dynamic thresholds are especially helpful to increase detection robustness in cases such as trotting or arm motion compared to when the robot is in a static configuration. A static threshold needs to be conservative to avoid FPs resulting from such scenarios, where $\hat{\boldsymbol{\tau}}_{\text{ext}}$ may increase due to high accelerations. This is especially the case for our experiments, which include backward, forward, and sideways trotting, where such variations can arise due to the high-frequency impact of the robot feet with the ground. In this work, however, we opted for using three different static thresholds for the following scenarios: trotting, arm motion, and stance. This proved sufficiently robust and did not exhibit any decrease in performance w.r.t. dynamic thresholds which, on the other hand, remain sensitive to tunable parameters.

C. Collision detection and identification results

In this section, we present the experimental results of our pipeline and discuss our findings. We test our collision identification method in different scenarios in the presence of the following factors of variation:

- **Mode.** Stance, arm motion, and trotting.
- **Added load.** As shown in Fig.1, we add a payload on different links of the robot. In particular, we consider the following cases: an unmodeled $0.58\,\text{kg}$ payload is placed on arm or base, or a $2\,\text{kg}$ load on the base (*Unmodeled load*); or a load of $0.58\,\text{kg}$ is added to the gripper (*Load gripper*) with its force measured by the F/T sensor placed at the robot end-effector.
- **Duration and magnitude of force.** Magnitudes of the applied force range in the interval $5 - 165\,\text{N}$, and time span of collision is in $0.3 - 6.0\,\text{s}$.

In all the experiments we use a cut-off frequency $\omega = 0.5\,\text{Hz}$ for estimating the disturbance force $\boldsymbol{F}_{dis}$ using Eq. 6. The cut-off frequencies of the BPF for $\hat{\boldsymbol{F}}_{\text{ext}}$ are selected as $0.4, 3.0\,\text{Hz}$.

1) Collision detection and time span estimation: We validate our collision detection algorithm over a dataset of 416 collision experiments on both arm and base. We define the *success rate* and the *precision* over N collisions as $(N - N_{\text{FN}})/N$ and $N/(N + N_{\text{FP}})$, respectively, where N_{FN} is the number of FNs, and N_{FP} is the number of FPs. Overall, we achieve a $99\,\%$ success rate and a precision of $98\,\%$ for collision detection. We can detect all collisions during stance and arm motion with and without unmodeled loads, i.e. all scenarios mentioned in Fig. 1 except for trotting. All undetected collisions occur during trotting and represent very challenging scenarios, such as when the magnitude of the collision force is below $10\,\text{N}$. Note that the noise of the raw MBO output during trotting can reach up to 35 N as can be seen in Fig. 9b.

Next, in Fig. 6 we analyze the time span estimation accuracy of our proposed approach over a set of 156 detected collisions performed in stance, arm motion, and trotting without considering additional loads. The average absolute error of time span estimation is $12\,\%$, $16\,\%$ and $20\,\%$, respectively, for the previously mentioned three scenarios.

2) Robustness analysis of collision identification: We compare the performance of our method in three different cases: *No load*, *Load gripper* and *Unmodeled loads* of different weights, placed on various locations on the robot. The external collision force estimation accuracy of the proposed identification method in the different scenarios is visualized in Fig. 7.

The variability is higher in the *Load gripper* and *Unmodeled load* scenarios, compared to the *No load* scenario. This

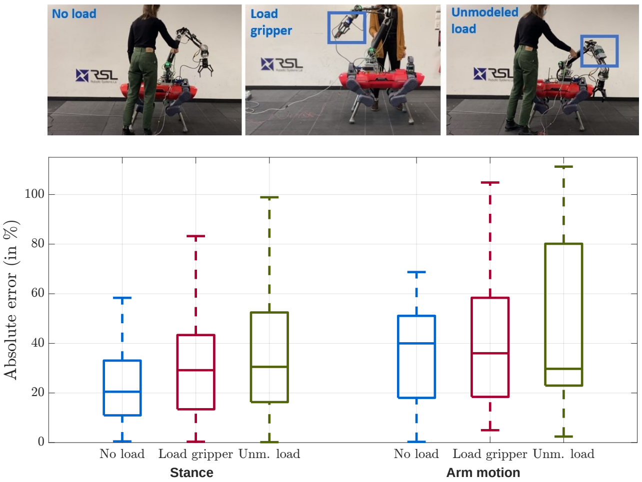


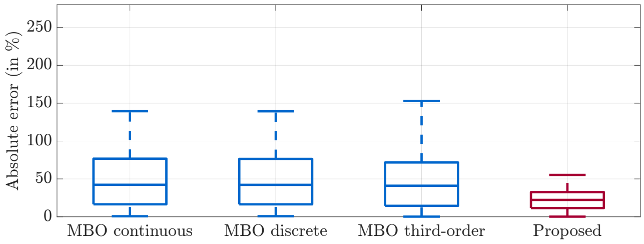
Fig. 7: Boxplots comparing the absolute estimation error of the external force magnitude in various scenarios. ‘Load gripper’ indicates presence of an object in the gripper and ‘unmodeled load’ an object on the arm or base.

is because of the increased inertia of the robot caused by the payload, which results in a more significant response to a collision. Additionally, the medians of the absolute errors between the three scenarios are comparable. This underlines the fact that our collision identification approach compensates robustly for the unmodeled load, due to the continuous estimation of the disturbance force $\boldsymbol{F}_{dis}$, as introduced in Sec. III-C.

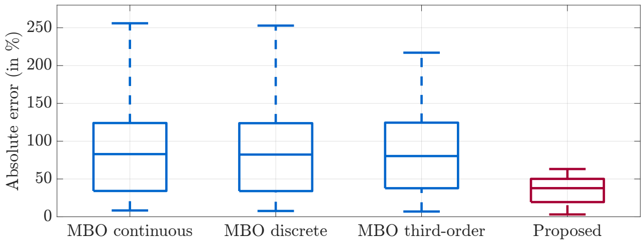
Furthermore, we evaluate the repeatability of our collision identification method. To do this, we split a dataset of 73 arm and base collisions in stance into 25 sets of doubles, triples, and quadruples of contacts with equal magnitude and location. The average of the standard deviation of the error computed within these sets is $5\,\%$, compared to a standard deviation of the error of $12\,\%$ within all collisions without the split.

3) Comparison against baseline collision identification methods: To validate the performance of our collision identification approach, we compare it to the state-of-the-art methods from Sec. IV-A that have been verified on legged robots. Hence, we select the continuous-time MBO from [6], discrete-time MBO [7] and third-order MBO [8]. As explained in Sec. IV-A, we use Eq. (3) to obtain the estimated external forces $\hat{\boldsymbol{F}}_{\text{ext,arm}}$ and $\hat{\boldsymbol{F}}_{\text{ext,base}}$. We carry out two different studies.

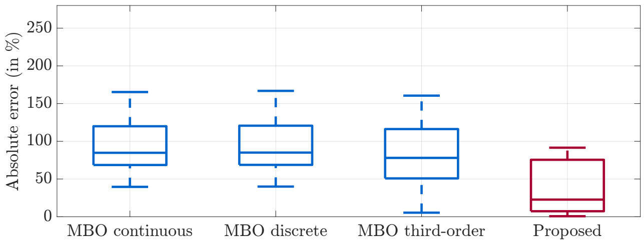
First, we compare the overall performance of our method in three scenarios: stance, arm movement, and trotting, without additional loads on the robot. The results for the first comparison study are shown in Fig. 8. The error metric e from Eq. (7) is used to compute the errors. With our method, a significant improvement in estimation accuracy can be achieved in various scenarios. The average absolute error is reduced from $52\,\%$ in methods [6], [7] and $49\,\%$ in [8] to $23\,\%$ during stance. While for trotting, the average error is reduced from $100, 101\,\%$ in [6], [7] and $87\,\%$ in [8] to $42\,\%$. Note that due to the high-frequency high impact forces of the feet in this case, the estimated forces show a large noise level. Therefore, we add an LPF to the estimated forces during trotting for all the methods in the comparison.



(a) Collisions on base and arm during stance.



(b) Collisions on base and arm during arm motion.

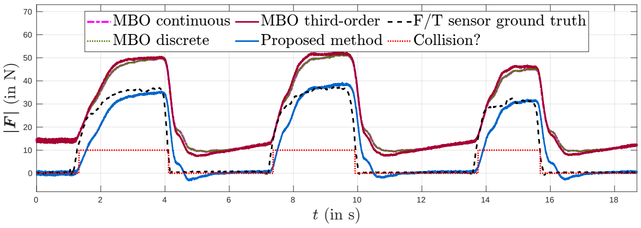


(c) Collisions on arm during trotting.

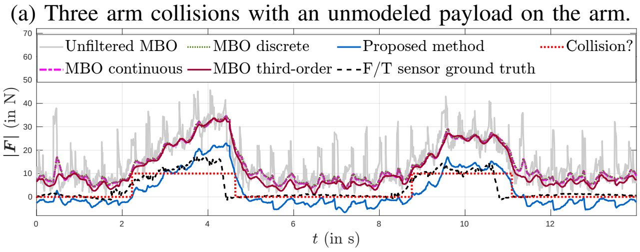
Fig. 8: Boxplots comparing the external force magnitude estimation of the following collision identification methods: continous-time MBO, discrete-time MBO, third-order MBO introduced in Sec. IV-A, and ours.

Secondly, we show a detailed analysis of the performance in two collision scenarios: with an additional load on the arm and during trotting. The corresponding plots are presented in Fig. 9. Note that the offset in the forces estimated with the conventional MBOs is due to errors coming from modeling inaccuracies and the unmodeled payload (in Fig. 9a). As shown in the offset in Fig. 9a, state-of-the-art methods cannot compensate for the unmodeled payload, while our method is able to do so.

Moreover, trotting is a challenging scenario with many high frequency variations in the estimated forces, as it can be seen in the unfiltered MBO signal presented in Fig. 9b. However, the trend line of the two collision forces is followed well and the absolute error is reduced with our approach: from $98, 97\,\%$ [6], [7] and $92\,\%$ [8] to $31\,\%$ during the first collision, and from $102\,\%$ [6], [7] and $88\,\%$ [8] to $18\,\%$ during the second.



(a) Three arm collisions with an unmodeled payload on the arm.



(b) Two arm collisions during trotting. Note that an LPF has been added to the estimated forces.

Fig. 9: Comparison of external collision forces from two very challenging scenarios estimated with state-of-the-art identification methods.

V. Conclusions and Future Work

In this paper, we introduce a collision event pipeline for quadrupedal manipulators. This includes a method for computing the time span of collisions under model inaccuracies and unmodeled loads, an estimation of disturbances resulting from these factors on the colliding robot parts, and, finally, an improved identification of the collision force itself.

We verify our approach by carrying out extensive hardware experiments, including unmodeled loads at different locations on the robot and trotting, summing up to 416 collisions. A comparison with other state-of-the-art approaches is also presented. It is worth mentioning that, up to our knowledge, such a thorough experimental validation has not been done previously in the literature on legged robots. Previous works have mentioned experiments with a maximum of 3 collisions in [4], 4 collisions in [3] and 10 collisions in [22].

In future work, we aim to extend this work by classifying collisions and designing appropriate reaction strategies. Moreover, to improve isolation, approaches that have been validated on manipulators, such as Bayesian filtering [29], [30] and machine learning [31]–[33] can be extended to the legged robot case.

References

[1] S. Haddadin, A. De Luca, and A. Albu-Schäffer, “Robot collisions: a survey on detection, isolation, and identification,” *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292–1312, 2017.

[2] S. Mamedov and S. Mikhel, “Practical aspects of model-based collision detection,” *Frontiers in Robotics and AI*, vol. 7, no. 1, pp. 1–15, 2020.

[3] T. Ito, K. Ayusawa, E. Yoshida, and A. Kheddar, “Experimental study for controller-friendly contact estimation for humanoid robot,” in *IEEE Workshop on Advanced Robotics and its Social Impacts*, vol. 2019, pp. 28–33, 2019.

[4] T. Mattioli and M. Vendittelli, “Interaction force reconstruction for humanoid robots,” *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 282–289, 2017.

[5] M. Van Damme, P. Beyl, B. Vanderborght, V. Grosu, R. Van Ham, I. Vanderniepen, A. Matthys, and D. Lefeber, “Estimating robot end-effector force from noisy actuator torque measurements,” in *IEEE International Conference on Robotics and Automation*, pp. 1108–1113, 2011.

[6] J. Vorndamme, M. Schappler, and S. Haddadin, “Collision detection, isolation and identification for humanoids,” in *IEEE International Conference on Robotics and Automation*, pp. 4754–4761, 2017.

[7] G. Bledt, P. M. Wensing, S. Ingersoll, and S. Kim, “Contact model fusion for event-based locomotion in unstructured terrains,” in *IEEE International Conference on Robotics and Automation*, pp. 4399–4406, 2018.

[8] V. Morlando, A. Teimoorzadeh, and F. Ruggiero, “Whole-body control with disturbance rejection through a momentum-based observer for quadruped robots,” *Mechanism and Machine Theory*, vol. 164, no. 2, pp. 1–20, 2021.

[9] A. Wahrburg, E. Morara, G. Cesari, B. Matthias, and H. Ding, “Cartesian contact force estimation for robotic manipulators using Kalman filters and the generalized momentum,” in *IEEE International Conference on Automation Science and Engineering*, pp. 1230–1235, 2015.

[10] G. Garofalo, N. Mansfeld, J. Jankowski, and C. Ott, “Sliding mode momentum observers for estimation of external torques and joint acceleration,” in *IEEE International Conference on Robotics and Automation*, pp. 6117–6123, 2019.

[11] A. De Luca and L. Ferrajoli, “Exploiting robot redundancy in collision detection and reaction,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3299–3305, 2008.

[12] A. De Luca, A. Albu-Schäffer, S. Haddadin, and G. Hirzinger, “Collision detection and safe reaction with the DLR-III lightweight manipulator arm,” in *IEEE International Conference on Intelligent Robots and Systems*, pp. 1623–1630, 2006.

[13] M. Geravand, F. Flacco, and A. De Luca, “Human-robot physical interaction and collaboration using an industrial robot with a closed control architecture,” in *IEEE International Conference on Robotics and Automation*, pp. 4000–4007, 2013.

[14] Z. Qiu, R. Ozawa, and S. Ma, “Adaptive virtual power-based collision detection and isolation with link parameter estimation,” *Advanced Robotics*, vol. 34, no. 12, pp. 814–825, 2020.

[15] C. Escobedo, M. Strong, M. West, A. Aramburu, and A. Roncone, “Contact anticipation for physical human–robot interaction with robotic manipulators using onboard proximity sensors,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 7255–7262, 2021.

[16] C. N. Cho, Y. L. Kim, and J. B. Song, “Adaptation-and-collision detection scheme for safe physical human-robot interaction,” *9th Asian Control Conference, ASCC*, 2013.

[17] C. N. Cho and J. B. Song, “Collision detection algorithm robust to model uncertainty,” *International Journal of Control, Automation and Systems*, vol. 11, no. 4, pp. 776–781, 2013.

[18] S. Faraji and A. J. IJspeert, “Designing a virtual whole body tactile sensor suit for a simulated humanoid robot using inverse dynamics,” in *IEEE International Conference on Intelligent Robots and Systems*, pp. 5564–5571, 2016.

[19] F. Flacco, A. Paolillo, and A. Kheddar, “Residual-based contacts estimation for humanoid robots,” in *IEEE-RAS International Conference on Humanoid Robots*, pp. 409–415, 2016.

[20] A. Bolotnikoval, S. Courtois, and A. Kheddar, “Compliant robot motion regulated via proprioceptive sensor based contact observer,” in *IEEE-RAS International Conference on Humanoid Robots*, pp. 854–859, 2019.

[21] A. Bolotnikova, S. Courtois, and A. Kheddar, “Contact observer for humanoid robot Pepper based on tracking joint position discrepancies,” in *IEEE International Symposium on Robot and Human Interactive Communication*, pp. 29–34, 2018.

[22] L. Hawley, R. Rahem, and W. Suleiman, “External force observer for small- and medium-sized humanoid robots,” *International Journal of Humanoid Robotics*, vol. 16, no. 6, pp. 1–25, 2019.

[23] Z. Cong, A. Honglei, C. Wu, L. Lang, Q. Wei, and M. Hongxu, “Contact force estimation method of legged-robot and its application in impedance control,” *IEEE Access*, vol. 8, pp. 161175–161187, 2020.

[24] J. Yang, H. Sun, D. Wu, X. Chen, and C. Wang, *Slip model-based foot-to-ground contact sensation via Kalman filter for miniaturized quadruped robots*. Springer, 2019.

[25] J. P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, "A unified MPC framework for whole-body dynamic locomotion and manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4688–4695, 2021.

[26] K. Manal and W. Rose, "A general solution for the time delay introduced by a low-pass Butterworth digital filter: an application to musculoskeletal modeling," *Journal of Biomechanics*, vol. 40, no. 3, pp. 678–681, 2007.

[27] D. Lim, D. Kim, and J. Park, "Momentum observer-based collision detection using LSTM for model uncertainty learning," in *IEEE International Conference on Robotics and Automation*, pp. 4516–4522, 2021.

[28] R. Priemer, *Fundamentals of digital signal processing*. CRC Press, 2018.

[29] L. Manuelli and R. Tedrake, "Localizing external contact using proprioceptive sensors: the contact particle filter," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 5062–5069, 2016.

[30] J. Bimbo, C. Pacchierotti, N. G. Tsagarakis, and D. Prattichizzo, "Collision detection and isolation on a robot using joint torque sensing," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 7604–7609, 2019.

[31] N. Briquet-Kerestedjian, A. Wahrburg, M. Grossard, M. Makarov, and P. Rodriguez-Ayerbe, "Using Neural Networks for classifying human-robot contact situations," in *European Control Conference*, pp. 3279–3285, 2019.

[32] J. Liang and O. Kroemer, "Contact localization for robot arms in motion without torque sensing," *IEEE International Conference on Robotics and Automation*, pp. 6322–6328, 2021.

[33] G. Cioffi, S. Klose, and A. Wahrburg, "Data-efficient online classification of human-robot contact situations," in *European Control Conference*, pp. 608–614, 2020.

Appendix A

Collision event pipeline

The collision event pipeline divides a collision into five phases and is introduced in [1]. The steps are visualized in Fig. A-1, with beneath each step the outputs. In the following paragraphs, the phases as they are defined in [1] are explained, with additionally the pre- and post-collision steps.

Pre-collision phase Before a collision occurs, the objective of a robot is to either avoid the collision while continuing to execute its task or to adjust its motion such that the impact is minimized. Both of these goals rely on path planning, which can be done either online or offline. To anticipate the collision, exteroceptive sensors can be used to measure e.g. distance, light intensity, or sound amplitude. Since this report assumes the use of proprioceptive sensors only, the focus is on minimizing impact, by going through the next phases.

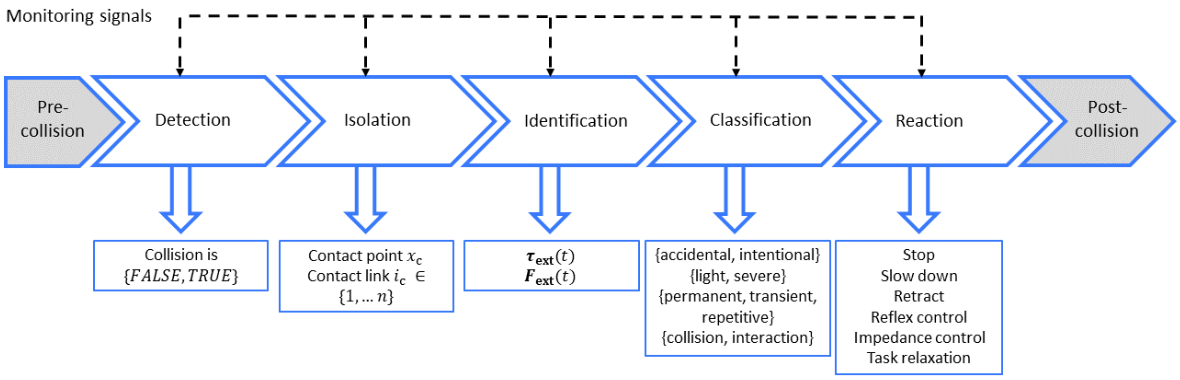


Figure A-1: The five phases of the collision event pipeline with their outputs. Monitoring signals serve as an input.

Collision detection Raw sensor readings or an estimated monitoring signal serve as an input to this phase. In literature, one often looks at the external torques on the robot structure as a monitoring signal. Appendix B goes into more details on torque estimators and other signals that can be observed. The obtained signal can additionally be passed through a filter to remove unwanted components or features of a signal. Finally, a threshold is set on the signal such that the output of this phase becomes the mapping of the monitoring signal into a binary class TRUE or FALSE, depending on if the collision occurred or not.

Collision isolation To solve the collision isolation problem is to find robot link i_c where the contact occurred, together with the collision contact point x_c. Detecting multiple collisions and estimating their contact forces can be achieved with the addition of either tactile or exteroceptive sensors [2]. Artificial skin tactile sensing in [2] is also used for additional information on the exact location of contact. The methods based on the use of these two sensors are beyond the scope of this report. From now on, when discussing monitoring and detection methods, the assumption of one collision only holds.

Achieving the exact collision location is challenging, even when these methods are based on accurate robot dynamics [3]. The robot's links are dynamically coupled, thus affecting isolation computations, since the dynamic impact effect of a collision will spread to other link variables or joint commands.

The most common solution to the isolation problem using external torques has been presented in [1, 4, 5, 6, 7] and is an analytical approach for localization. It has also been demonstrated to work on legged robots in [8, 9, 10, 11]. The analytical method suffices in detecting the contact link by looking at joint torques, however, errors can arise when a contact force does not exert any torque at the joint before the contact link [12]. Moreover, as mentioned before, a unique solution does not exist. Finally, it depends on an accurate force and torque estimate, which is not always available for legged robots [11, 10]. When isolation is only used for the implementation of a reaction strategy to ensure safety, and the robot is a fixed-base manipulator, using this method will be acceptable.

An approach that gives a more exact location of the contact force is a probabilistic strategy using Bayesian filtering [13, 12]. The precise contact location of forces anywhere on the robot structure can be estimated, even on links close to the robot base. Thus, this is a more accurate localization strategy and it does not suffer under modeling errors. However, the approach by [13] requires at least 161 ms computation time for one contact estimation, and for three even 395 ms. Moreover, it relies on an accurate robot model and has only been validated in simulation, whereas hardware would add sensor noise and friction. The method proposed by [12] requires the robot to perform small exploratory movements while maintaining contact with the colliding contact to converge to the contact location. When the goal of collision detection is to move away from a contact as fast as possible, these movements are undesired. Another approach for localization is a machine learning technique using the estimated external joint torque. In [14], supervised learning is used to locate the contact on either the upper or lower robot arm. Similarly, [15, 16] use a Neural Network (NN) to determine the contact location on the upper or lower manipulator arm. In [17], a machine learning algorithm using $\boldsymbol{\tau}_{\text{ext}}$ as input can distinguish the robot link and can differentiate between different contact points on that link. The approach results in a unique solution and only 1D torques are required as input, instead of the 6D wrench needed as input for the analytical method described above. Another localization alternative is given by [18], whose algorithm is not based on machine learning but on a mesh of the robot surface. Local optimization around the approximate contact location is used to isolate the contact on a pre-specified point. The limitations of these machine learning approaches are that the training phase takes much time and more importantly, the localization is dependent on the training data and the human who applied the collisions. A final approach to solving the collision isolation problem is given by [19], who proposes a virtual power-based collision detection index. This method, however, has only been tested on a 2-Degrees of Freedom (DoF) robot arm moving in a one directional plane.

Collision identification To identify the collision is to find the direction and intensity of the contact wrench, which is the sum of forces and moments exerted by the environment on the robot [20]. This force is defined either with external wrench $\boldsymbol{\mathcal{F}}_{\text{ext}}(t)$ at the contact or external torques $\boldsymbol{\tau}_{\text{ext}}(t)$ during the collision event.

Collision classification A contact force after detection can be classified, such that the robot can react appropriately. A distinction is made between a light or severe and an intentional or unwanted force. The latter distinction is also referred to as the difference between interaction and collision. Additionally, the force's time course is categorized as either permanent, transient, or repetitive.

Reaction to collision Once a contact is classified, the robot should decide on its reaction strategy. Depending on if an external force is classified as a collision or an interaction, a different reaction is desired. Considering safety in physical Human-Robot Interaction (pHRI), a **collision impact force** requires the robot to either stop its motion completely [3, 20]; to move away from the contact [21, 22]; to fully comply with an interaction force, applied by a human, in order to follow the human's intentions and lose contact after some time [20, 3]; or to move around the contact occurring along the robot structure, to continue task execution at the end-effector [21, 23]. On the other hand, when an **interaction impact force** is applied at the end-effector, while the robot collides at another point along its structure, the robot can use the nullspace to move around the colliding force [21, 23, 24]. If the interaction takes place at either the end-effector or an arbitrary point along the robot structure, the robot can comply, thus letting itself be guided by the human. If pHRI happens along the structure, while the end-effector is executing a task, the nullspace can be exploited again, this time aiming for manual guidance by the human such that the human can reconfigure the robot's joint positions.

Post-collision phase The reaction strategy determined in the previous phase is executed by the design of an appropriate controller.

Appendix B

Torque estimation

The disturbances acting on the robot are given in the terms of estimated external torques $\boldsymbol{\tau}_{\text{ext}}$. This appendix first introduces the state-of-the-art torque estimation methods in Section B-1. The external observers that are experimentally validated in Chapter 1 are defined in Section B-2. Additionally, the baseline torque estimation methods, with which a comparison is made in Chapter 2, are discussed.

B-1 State-of-the-art

Looking at torques $\boldsymbol{\tau}_{\text{ext}}$, external forces can be estimated and, by noticing changes in this term, a collision can be detected. To detect a collision force, one can also consider observing a different signal, such as the motor torque $\boldsymbol{\tau}_{\text{m}}$, its derivative $\dot{\boldsymbol{\tau}}_{\text{m}}$ or the motor current i [1, 25]. However, this results in low detection accuracy and therefore the focus in our work is on estimating $\boldsymbol{\tau}_{\text{ext}}$.

Direct estimation is one of the simplest methods [1]. The need for $\ddot{\boldsymbol{q}}$ is a disadvantage of this scheme since its estimation amplifies measurement noise due to numerically differentiating the joint positions twice. Other schemes are the energy observer, which is not always able to detect a collision; the velocity observer, which shows nonlinear observer error dynamics, uses undesired inversion of $\mathbf{M}(\boldsymbol{q})$ and contains noisy acceleration term $\ddot{\boldsymbol{q}}$; and inverse dynamics, which can only be used on a predefined trajectory [1]. Moreover, some external observers make use of Inertial Measurement Unit (IMU) measurements, which gives an accurate $\hat{\boldsymbol{\tau}}_{\text{ext}}$, however, no IMUs are attached to the joints and links of Alma C and thus these methods are outside of consideration [26, 27]. To improve estimation in case of an inaccurate dynamical model, an approach based on the Extended Kalman Filter (EKF) is presented in [28]. A limitation is the high computational cost, as a consequence of calculating the torque Jacobian $\mathbf{A}$ that expresses the partial derivatives of the joint torques with respect to each other. For tests with a 7-Degrees of Freedom (DoF) robot arm, torque data is collected with a maximum frequency of 56 Hz. Increasing the DoFs to 24 results in even higher computation times. Another improvement in calculation of $\hat{\boldsymbol{\tau}}_{\text{ext}}$ in case of model uncertainties is shown by learning techniques [29, 30, 31, 32]. Although machine learning methods show significantly higher detection accuracy, they are dependent on the user, the robot, and the task or trajectory. Furthermore, the system is often modeled as a grey or black box, meaning that if the system fails, it is not clear what has caused the failure.

In Section 1-III, observers based on the generalized momentum are introduced. Additionally, in Section 2-IV-C, the proposed identification method is compared against state-of-the-art torque estimation approaches. In the following sections, these torque estimation schemes are discussed and the derivations of their final expressions are given.

B-2 Derivation of torque estimation schemes

Note that for ease of notation, the torques measured by force/torque (F/T) sensors on the body, indicated with $\boldsymbol{\tau}_{\text{ft}}$, are omitted in the following derivations.

B-2-1 Direct estimation

Rewriting Eq. (1-2), the direct estimation of $\boldsymbol{\tau}_{\text{ext}}$ is given by [1]

$$\hat{\boldsymbol{\tau}}_{\text{ext}} = \hat{\mathbf{M}}(\boldsymbol{q})\dot{\boldsymbol{v}} + \hat{\mathbf{C}}(\boldsymbol{q}, \boldsymbol{v})\boldsymbol{v} + \hat{\boldsymbol{g}}(\boldsymbol{q}) - \mathbf{S}^T \boldsymbol{\tau}_{\text{m}}. \tag{B-1}$$

B-2-2 Momentum Based Observer - continuous time

The time derivative of the generalized momentum $\boldsymbol{p} = \mathbf{M}(\boldsymbol{q})\boldsymbol{v}$ is denoted as $\dot{\boldsymbol{p}}$ and defined as follows [22]

$$\begin{aligned} \dot{\boldsymbol{p}} &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{M}(\boldsymbol{q})\boldsymbol{v}\right) = \dot{\mathbf{M}}(\boldsymbol{q})\boldsymbol{v} + \mathbf{M}(\boldsymbol{q})\dot{\boldsymbol{v}}, \\ &= \boldsymbol{\tau}_{\text{ext}} + \mathbf{S}^T \boldsymbol{\tau}_{\text{m}} + \dot{\mathbf{M}}(\boldsymbol{q})\boldsymbol{v} - \mathbf{C}(\boldsymbol{q}, \boldsymbol{v})\boldsymbol{v} - \boldsymbol{g}(\boldsymbol{q}), \end{aligned} \tag{B-2}$$

where the term $\mathbf{M}(\boldsymbol{q})\dot{\boldsymbol{v}}$ in the first line is substituted with a rewritten version of the robot dynamic model, as given in Eq. (1-2). The following matrix identity, based on skew-symmetry, is used to simplify the above expression

$$\dot{\mathbf{M}}(\boldsymbol{q}) = \mathbf{C}(\boldsymbol{q}, \boldsymbol{v}) + \mathbf{C}^T(\boldsymbol{q}, \boldsymbol{v}). \tag{B-3}$$

The nonlinear terms $\boldsymbol{n}(\boldsymbol{q}, \boldsymbol{v}) := \boldsymbol{g}(\boldsymbol{q}) + \mathbf{C}(\boldsymbol{q}, \boldsymbol{v})\boldsymbol{v} - \dot{\mathbf{M}}(\boldsymbol{q})\boldsymbol{v}$ are then shortened to

$$\boldsymbol{n}(\boldsymbol{q}, \boldsymbol{v}) := \boldsymbol{g}(\boldsymbol{q}) - \mathbf{C}^T(\boldsymbol{q}, \boldsymbol{v})\boldsymbol{v}. \tag{B-4}$$

Finally, the dynamics of the generalized momentum, as defined in Eq. (B-2), are expressed as

$$\dot{\boldsymbol{p}} = \mathbf{S}^T \boldsymbol{\tau}_{\text{m}} - \boldsymbol{n}(\boldsymbol{q}, \boldsymbol{v}) + \boldsymbol{\tau}_{\text{ext}}. \tag{B-5}$$

The idea of the Momentum Based Observer (MBO) is to observe the dynamics of the generalized momentum in Eq. (B-5). When looking at the block diagram of the observer in Figure B-1, it is noted that the dynamics of the estimated external joint torques $\hat{\boldsymbol{\tau}}_{\text{ext}}$ are given as follows

$$\dot{\hat{\boldsymbol{\tau}}}_{\text{ext}} = \mathbf{K}_{\text{O}}(\dot{\boldsymbol{p}} - \dot{\hat{\boldsymbol{p}}}). \tag{B-6}$$

Integrating this expression and using the nonlinear terms from Eq. (B-4) gives estimated torques $\hat{\boldsymbol{\tau}}_{\text{ext}}$ as the observer output

$$\begin{aligned}
\hat{\boldsymbol{\tau}}_{\text{ext}} &= \mathbf{K}_{\text{O}} \int_0^t \left(\dot{\boldsymbol{p}} - \dot{\hat{\boldsymbol{p}}}\right) \mathrm{d}s, \\
&= \mathbf{K}_{\text{O}} \left(\boldsymbol{p}(t) - \int_0^t \dot{\hat{\boldsymbol{p}}}(s)\mathrm{d}s - \boldsymbol{p}(0)\right), \\
&= \mathbf{K}_{\text{O}} \left(\boldsymbol{p}(t) - \int_0^t \left(\mathbf{S}^T \boldsymbol{\tau}_{\text{m}} - \hat{\boldsymbol{n}}(\boldsymbol{q}, \boldsymbol{v}) + \hat{\boldsymbol{\tau}}_{\text{ext}}\right) \mathrm{d}s - \boldsymbol{p}(0)\right),
\end{aligned} \tag{B-7}$$

where momentum $\boldsymbol{p}(t)$ is calculated using an estimation of the inertia matrix: $\boldsymbol{p} = \hat{\mathbf{M}}(\boldsymbol{q})\boldsymbol{v}$. Note that the above equation is similar to the MBO expression in Eq. (1-4), however, in Eq. (B-7) the initial momentum $\boldsymbol{p}(0)$ is added. In this work, $\boldsymbol{p}(0) = \mathbf{0}$ is assumed. For implementation, the integral in Eq. (B-7) is discretized. In [33], various integration schemes are compared and it is concluded that a trapezoidal approach [34] gives acceptably large sampling rates

$$\int_0^t f(x)\mathrm{d}x \approx \sum_{k=1}^{K} \frac{f(x_{k-1}) + f(x_k)}{2} T_s, \tag{B-8}$$

where $f(x)$ is the function to be discretized for $k = 1, \ldots, K$ time steps and T_{s} the sampling time.

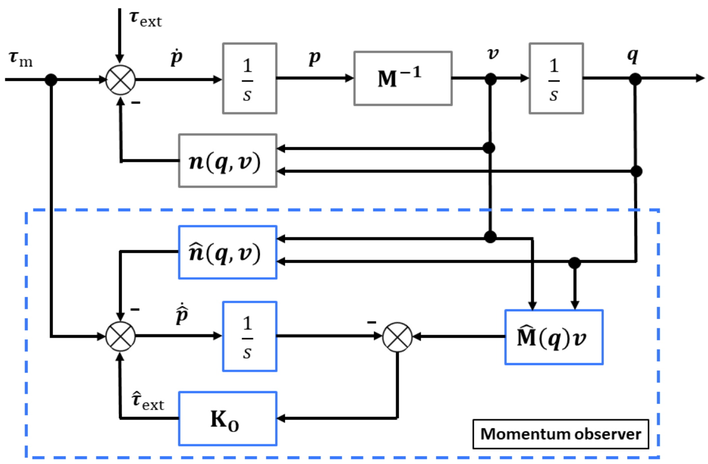


Figure B-1: Block diagram of the Momentum Based Observer (MBO), which observers the generalized momentum dynamics $\dot{\boldsymbol{p}}$ to compute an estimate of the external torques $\boldsymbol{\tau}_{\text{ext}}$.

The ideal dynamic relation between $\boldsymbol{\tau}_{\text{ext}}$ and $\hat{\boldsymbol{\tau}}_{\text{ext}}$ is obtained by rewriting Eq. (B-6) [22]

$$\begin{aligned}
\dot{\hat{\boldsymbol{\tau}}}_{\text{ext}} &= \mathbf{K}_{\text{O}}(\dot{\boldsymbol{p}} - \dot{\hat{\boldsymbol{p}}}), \\
&= \mathbf{K}_{\text{O}} \left(\mathbf{S}^T \boldsymbol{\tau}_{\text{m}} - \boldsymbol{n}(\boldsymbol{q}, \boldsymbol{v}) + \boldsymbol{\tau}_{\text{ext}} - \left(\mathbf{S}^T \boldsymbol{\tau}_{\text{m}} - \hat{\boldsymbol{n}}(\boldsymbol{q}, \boldsymbol{v}) + \hat{\boldsymbol{\tau}}_{\text{ext}}\right)\right), \\
&= \mathbf{K}_{\text{O}} \left(\boldsymbol{\tau}_{\text{ext}} - \hat{\boldsymbol{\tau}}_{\text{ext}}\right),
\end{aligned} \tag{B-9}$$

where a perfect model is assumed, and thus $\hat{\boldsymbol{n}}(\boldsymbol{q}, \boldsymbol{v}) = \boldsymbol{n}(\boldsymbol{q}, \boldsymbol{v})$. When mapping the monitoring signal from time to frequency domain using the Laplace transform and working component wise, the following expression is obtained

$$\hat{\tau}_{\text{ext},i} = \frac{k_{\text{O},i}}{s + k_{\text{O},i}} \tau_{\text{ext},i} = \frac{1}{1 + T_{\text{O},i} s} \tau_{\text{ext},i}, \quad i = 1, \ldots, N, \tag{B-10}$$

where $k_{\text{O},i}$ are the observer gains on the diagonal of $\mathbf{K}_\text{O}$ and $T_{\text{O},i}$ is the time constant associated to component $\hat{\tau}_{\text{ext},i}$, defined as $T_{\text{O},i} = \frac{1}{k_{O,i}}$. The total robot number of DoF is $N = 6 + n_\text{j}$ and s denotes the complex Laplace variable. If gain matrix $\mathbf{K}_\text{O}$ goes to infinity, signal $\hat{\boldsymbol{\tau}}_\text{ext}$ approaches $\boldsymbol{\tau}_\text{ext}$. Eq. (B-10) represents a first-order low-pass filter, which is stable, linear and decoupled.

B-2-3 Momentum Based Observer - discrete time

To get a better understanding of how the MBO can be implemented in discrete-time [35], first, the continuous-time MBO is rewritten.

Rewriting the continuous-time MBO Let $\boldsymbol{w} = k_\text{O}\boldsymbol{p} - \hat{\boldsymbol{\tau}}_\text{ext}$. Note that in [35], gains k_O are assumed equal for each robot DoF is assumed. Substituting this in the final expression for the MBO, Eq. (B-7), and assuming $\boldsymbol{p}(0) = \mathbf{0}$, the following is obtained

$$\boldsymbol{w} = k_\text{O} - \int_0^t \left(\mathbf{S}^T \boldsymbol{\tau}_\text{m} - \hat{\boldsymbol{n}}(\boldsymbol{q}, \boldsymbol{v}) + k_\text{O}\boldsymbol{p} - \boldsymbol{w} \right) \mathrm{d}s. \tag{B-11}$$

The above expression can be seen as a low-pass filter (LPF) acting on the inertial and nonlinear terms, which is defined as follows

$$\boldsymbol{w} = \frac{k_\text{O}}{s + k_\text{O}} \left(\mathbf{S}^T \boldsymbol{\tau}_\text{m} - \hat{\boldsymbol{n}}(\boldsymbol{q}, \boldsymbol{v}) + k_\text{O}\boldsymbol{p} \right). \tag{B-12}$$

Then, the external torques are substituted back in the equation using $\boldsymbol{w} = k_\text{O}\boldsymbol{p} - \hat{\boldsymbol{\tau}}_\text{ext}$, such that

$$\hat{\boldsymbol{\tau}}_\text{ext} = k_\text{O}\boldsymbol{p} - \frac{k_\text{O}}{s + k_\text{O}} \left(\mathbf{S}^T \boldsymbol{\tau}_\text{m} - \hat{\boldsymbol{n}}(\boldsymbol{q}, \boldsymbol{v}) + k_\text{O}\boldsymbol{p} \right). \tag{B-13}$$

Note that the observer filters the disturbances $\boldsymbol{\tau}_\text{ext}$ by feed-forwarding the momentum $\boldsymbol{p}$ and the dynamic model terms.

Implementing the MBO in discrete-time This implementation starts by applying a discrete-time filter [36] to both sides of the direct estimation expression Eq. (B-1)

$$\begin{aligned} \frac{1-\gamma}{1-\gamma z^{-1}} \boldsymbol{\tau}_\text{ext} &= \frac{1-\gamma}{1-\gamma z^{-1}} \left(\hat{\mathbf{M}}(\boldsymbol{q})\dot{\boldsymbol{v}} + \hat{\mathbf{C}}(\boldsymbol{q}, \boldsymbol{v})\boldsymbol{v} + \hat{\boldsymbol{g}}(\boldsymbol{q}) - \mathbf{S}^T \boldsymbol{\tau}_\text{m} \right), \\ \hat{\boldsymbol{\tau}}_\text{ext} &= \frac{1-\gamma}{1-\gamma z^{-1}} \left(\hat{\mathbf{M}}(\boldsymbol{q})\dot{\boldsymbol{v}} + \hat{\mathbf{C}}(\boldsymbol{q}, \boldsymbol{v})\boldsymbol{v} + \hat{\boldsymbol{g}}(\boldsymbol{q}) - \mathbf{S}^T \boldsymbol{\tau}_\text{m} \right), \end{aligned} \tag{B-14}$$

where, as introduced in Section 1-III-A, z is the discrete-time domain variable, and $0 < \gamma < 1$ are values monotonically related tot the cut-off frequency. It is noted that in the above expression the acceleration $\dot{\boldsymbol{v}}$ is present, which is undesired. Therefore, a closer look is taken at the term $\hat{\mathbf{M}}(\boldsymbol{q})\dot{\boldsymbol{v}}$ in Eq. (B-14), and application of the filter to this term is rewritten as follows

$$\frac{1-\gamma}{1-\gamma z^{-1}}\hat{\mathbf{M}}(\boldsymbol{q})\dot{\boldsymbol{v}} = \sum_{k=0}^{K} h(K-k)\hat{\mathbf{M}}(\boldsymbol{q}(k))\dot{\boldsymbol{v}}(k) \tag{B-15a}$$

$$= \beta\boldsymbol{p}(K+1) - \sum_{k=0}^{K} h(K-k)\Big(\frac{1}{T_{\text{s}}}\Big(\hat{\mathbf{M}}(\boldsymbol{q}(k+1)) - \hat{\mathbf{M}}(\boldsymbol{q}(k))\Big)\boldsymbol{v}(k+1)\dots + \beta\boldsymbol{p}(k+1)\Big), \tag{B-15b}$$

$$= \beta\boldsymbol{p} - \frac{1-\gamma}{1-\gamma z^{-1}}\left(\hat{\mathbf{C}}(\boldsymbol{q},\boldsymbol{v})\boldsymbol{v} + \hat{\mathbf{C}}(\boldsymbol{q},\boldsymbol{v})^T\boldsymbol{v} + \beta\boldsymbol{p}\right), \tag{B-15c}$$

where $\beta = \frac{(1-\gamma)\gamma_i^{-1}}{T_{\text{s}}}$. In Eq. (B-15a), the summation of impulse responses $h(k) = (1-\gamma)\gamma^k$ of the discrete-time filter is evaluated [36]. Next, Eq. (B-15b) analyzes the previous sum using *summation by parts*. Finally, Eq. (B-15c) solves the summation and uses the matrix identity from Eq. (B-3) to eliminate the differentiation of the inertia matrix. Substituting the result of Eq. (B-15) in Eq. (B-14), the final expression for estimating the external torques is given as

$$\begin{aligned}\hat{\boldsymbol{\tau}}_{\text{ext}} &= \beta\boldsymbol{p}(k) - \frac{1-\gamma}{1-\gamma z^{-1}}\left(\beta\boldsymbol{p} + \mathbf{S}^T\boldsymbol{\tau}_{\text{m}} + \hat{\mathbf{C}}(\boldsymbol{q},\boldsymbol{v})^T\boldsymbol{v} - \hat{\boldsymbol{g}}(\boldsymbol{q})\right), \\ &= \beta\boldsymbol{p}(k) - \frac{1-\gamma}{1-\gamma z^{-1}}\left(\mathbf{S}^T\boldsymbol{\tau}_{\text{m}} - \hat{\boldsymbol{n}}(\boldsymbol{q},\boldsymbol{v}) + \beta\boldsymbol{p}\right).\end{aligned} \tag{B-16}$$

Note that this has a similar structure to Eq. (B-13), however, the gains are calibrated in discrete-time. When the dynamic motion is high, discretization errors can arise using Eq. (B-13), but are prevented with Eq. (B-16).

B-2-4 Momentum Based Observer - higher order

Higher-order versions of the MBO are implemented in [37, 38, 39]. Here, the higher-order system as defined in [37] is described. Note that the first-order LPF in Eq. (B-10) is now extended to a filter of order $r > 0$

$$\hat{\tau}_{\text{ext},i} = \frac{k_{\text{O},i}}{s^r + c_{r-1}s^{r-1} + \ldots + c_1 s + c_0}\tau_{\text{ext},i}, \quad i = 1,\ldots,N, \tag{B-17}$$

where k_{O} is the observer gain as defined before, and c_j constant coefficients, where $j = 0,\ldots,r-1$. Similar to the continuous-time MBO, the formulation of the observer is not given in the Laplace-domain but in continuous-time

$$\boldsymbol{\delta}_1 = \mathbf{K}_{\text{O},1}\left(\boldsymbol{p} - \int_0^t\left(\mathbf{S}^T\boldsymbol{\tau}_{\text{m}} - \hat{\boldsymbol{n}}(\boldsymbol{q},\boldsymbol{v}) + \hat{\boldsymbol{\tau}}_{\text{ext}}+\right)\mathrm{d}s\right), \tag{B-18}$$

$$\boldsymbol{\delta}_h = \mathbf{K}_{\text{O},h}\int_0^t\left(\boldsymbol{\delta}_{h-1} - \hat{\boldsymbol{\tau}}_{\text{ext}}+\right)\mathrm{d}s, \tag{B-19}$$

where $h = 1,\ldots,r$ and $\hat{\boldsymbol{\tau}}_{\text{ext}} = \boldsymbol{\delta}_r$. Again, the integrals are discretized using the trapezoidal integration rule.

B-2-5 Momentum Based Kalman filter

The continuous system and output dynamics in Eq. (1-6) and Eq. (1-7) are discretized to a linear time-invariant (LTI) system such that a discrete Kalman filter can be applied. The continuous-time versions of state-space matrices $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$, and the noise covariance matrices $\mathbf{Q}_\tau, \mathbf{Q}_p$ and $\mathbf{R}$ are defined in Chapter 1. Selecting the diagonals of the process and measurement noise matrices, is explained in Appendix D-4. According to [40], discretized matrices $\mathbf{A}_\mathrm{d}, \mathbf{B}_\mathrm{d}, \mathbf{C}_\mathrm{d}$ and $\mathbf{R}_\mathrm{d}$ are denoted as

$$\begin{bmatrix} \mathbf{A}_\mathrm{d} & \mathbf{B}_\mathrm{d} \\ \mathbf{0}_{N\times 2N} & \mathbf{I}_N \end{bmatrix} = \exp\left(\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0}_{N\times 2N} & \mathbf{0}_{N\times N} \end{bmatrix} T_s\right), \tag{B-20}$$

$$\mathbf{C}_\mathrm{d} = \mathbf{C}, \tag{B-21}$$

$$\mathbf{R}_\mathrm{d} = \frac{1}{T_s}\mathbf{R}. \tag{B-22}$$

The process noise covariance matrices are stacked in matrix $\mathbf{Q} = \mathrm{diag}\left(\mathbf{Q}_p, \mathbf{Q}_\tau\right) \in \mathbb{R}^{2N\times 2N}$ and its discretized version $\mathbf{Q}_\mathrm{d}$ is obtained in the following three steps

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} & \mathbf{Q} \\ \mathbf{0}_{2N\times 2N} & -\mathbf{A}^T \end{bmatrix}, \tag{B-23a}$$

$$\begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{0}_{2N\times 2N} & \mathbf{M}_{22} \end{bmatrix} = \exp\left(\mathbf{H}\cdot T_s\right), \tag{B-23b}$$

$$\mathbf{Q}_\mathrm{d} = \mathbf{M}_{12}\left(\mathbf{M}_{11}\right)^T. \tag{B-23c}$$

Then, the discrete-time Kalman filter can be applied to the discretized dynamics of Eq. (1-6) and Eq. (1-7). This method uses an estimate of the covariance matrix $\hat{\mathbf{P}} \in \mathbb{R}^{2N\times 2N}$ and Kalman gain $\mathbf{K} \in \mathbb{R}^{N\times N}$. The following steps are similar to those described in [40], although in our work the external torques $\hat{\boldsymbol{\tau}}_\mathrm{ext}$ are calculated as an output vector instead of the external forces $\hat{\boldsymbol{F}}_\mathrm{ext}$.

The discrete-time Kalman filter

Inputs. Measured $\boldsymbol{q}, \boldsymbol{v}, \boldsymbol{\tau}_\mathrm{m}$. Discretized state-space system matrices $\mathbf{A}_\mathrm{d}, \mathbf{B}_\mathrm{d}$ using Eq. (B-20), and $\mathbf{C}_\mathrm{d}$ using Eq. (B-21). Discretized covariance measurement noise matrix $\mathbf{R}_\mathrm{d}$ from Eq. (B-22) and process noise matrix $\mathbf{Q}_\mathrm{d}$ using Eq. (B-23).

Output. The estimated external torques $\hat{\boldsymbol{\tau}}_\mathrm{ext}(k)$.

Initialize covariance matrix $\hat{\mathbf{P}}(0) = \mathbf{I}_{2N}$ and $\hat{\boldsymbol{x}}(0) = \mathbf{0} \in \mathbb{R}^{2N}$, assuming that the robot is initially in stance and no collision is applied yet. For each time instant k, fives steps are followed.

1. Calculate the output vector $\boldsymbol{y}(k) = \boldsymbol{p}(k) = \mathbf{M}(\boldsymbol{q}(k))\boldsymbol{v}(k)$ and input vector $\boldsymbol{u}(k) = \mathbf{S}^T\boldsymbol{\tau}_\mathrm{m}(k) - \hat{\boldsymbol{n}}(\boldsymbol{q}(k), \boldsymbol{v}(k))$.
2. Predict the state and the covariance matrix

$$\hat{\boldsymbol{x}}(k) = \mathbf{A}_\mathrm{d}\hat{\boldsymbol{x}}(k-1) + \mathbf{B}_\mathrm{d}\boldsymbol{u}(k-1), \tag{B-24}$$

$$\hat{\mathbf{P}}(k) = \mathbf{A}_\mathrm{d}\hat{\mathbf{P}}(k-1)\mathbf{A}_\mathrm{d}^T + \mathbf{Q}_\mathrm{d}. \tag{B-25}$$

3. Calculate the Kalman gain

$$\mathbf{K}(k) = \hat{\mathbf{P}}(k)\mathbf{C}_\mathrm{d}^T \left(\mathbf{C}_\mathrm{d}\hat{\mathbf{P}}(k)\mathbf{C}_\mathrm{d}^T + \mathbf{R}_\mathrm{d}\right)^{-1}. \tag{B-26}$$

4. Estimate the state vector and covariance matrix from step 2 again, but corrected with the current measurements

$$\hat{\boldsymbol{x}}(k) = \hat{\boldsymbol{x}}(k) + \mathbf{K}(k)\left(\boldsymbol{y}(k) - \mathbf{C}_\mathrm{d}\hat{\boldsymbol{x}}(k)\right), \tag{B-27}$$

$$\hat{\mathbf{P}}(k) = \left(\mathbf{I}_{2N} - \mathbf{K}(k)\mathbf{C}_\mathrm{d}\right)\hat{\mathbf{P}}(k)\left(\mathbf{I}_{2N} - \mathbf{K}(k)\mathbf{C}_\mathrm{d}\right)^T + \mathbf{K}(k)\mathbf{R}_\mathrm{d}\mathbf{K}(k)^T. \tag{B-28}$$

5. Obtain the estimate of the external torques

$$\hat{\boldsymbol{\tau}}_\mathrm{ext}(k) = \begin{bmatrix}\mathbf{0}_{N\times N} & \mathbf{I}_N\end{bmatrix}\hat{\boldsymbol{x}}(k). \tag{B-29}$$

Appendix C

Collision detection, isolation and identification

In this appendix, three steps of the collision event pipeline are discussed: detection, isolation, and identification. In Section C-1, the concept of windowing used in frequency-domain filtering is explained. Thereafter, an efficient algorithm for the standard deviation dynamic threshold is introduced. Furthermore, the flowchart behind the collision detection approach proposed in Chapter 2 is presented. Next, in Section C-2, the collision isolation scheme is visualized and explained. Finally, Section C-3 discusses manipulator singularities, which play an important role in the collision identification process.

C-1 Collision detection

C-1-1 Filtering the force

Half-Hann and rectangular window explained When filtering a force in frequency domain, the Fast Fourier Transform (FFT) of the estimated force is weighted by a window w_n, see Eq. (1-11). This window can take various forms, such as rectangular or the form of a half-Hann window. This type of windowing is most common in use since it reduces spectral leakage, has good frequency resolution and can be applied when the nature of the signal is unknown [41]. The Hann window is denoted as [42]

$$w_n = \frac{1}{2}\left(1 - \cos\left(\frac{2\pi n}{2N_\mathrm{w} - 1}\right)\right) \quad \text{for } n = 0, \ldots, N_\mathrm{w} - 1. \tag{C-1}$$

The concept of the sliding window, the transformation of the estimated force $\hat{\boldsymbol{F}}_\mathrm{ext}$ from time to frequency domain and the difference between the two aforementioned windows is visualized in Figure C-1. In the upper two plots, a rectangular window is applied, and in the lower two, a half-Hann window. Since the latter gives larger weights to current measurements in the window, the frequency spectrum (*lower right*) gives larger power values $\hat{F}_\omega$ in the higher frequency range compared to the rectangular window (*upper right*). Only taking into account frequencies between $[\omega_{\text{min}}, \omega_{\text{max}}]$ allows filtering. Finally, the 1-norm of the vector of power values $\boldsymbol{F}_\omega$ is calculated over this range to obtain the filtered force value at time step t_1, as in Eq. (1-13).

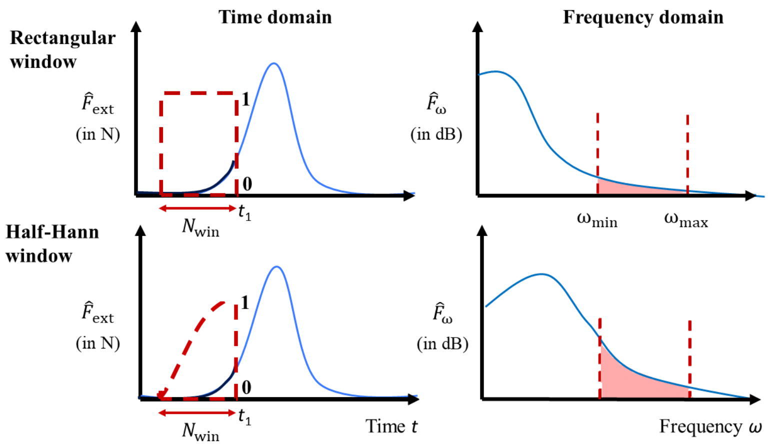


Figure C-1: The FFT of a collision force $\hat{F}_{\text{ext}}$ is calculated over a fixed time window at time instant t_1. Either a rectangular window is applied (*upper left*), or a half-Hann window (*lower left*). Transforming the force values in the window from time to frequency domain results in the graphs on the right.

Limitation of the sliding-window Fast Fourier Transform A downside of using the sliding window FFT, also called the Short-time Fourier Transform (STFT) [43], is that it has a fixed time and frequency resolution Δt and $\Delta\omega$, and that a trade-off exists between these two. This is shown in Figure C-2, and expressed in Eq. (1-12). A good frequency resolution means that $\Delta\omega$ is small and thus that the frequency components of a signal can be separated into multiple blocks. Consequently, signals showing similar frequency characteristics can be distinguished from each other. On the other hand, a small time resolution is desired to notice frequency changes faster.

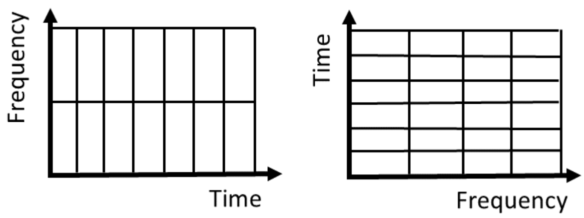


Figure C-2: A limitation of the FFT is that a trade-off exists between the time and frequency resolution. The left shows a better time, and the right a better frequency resolution.

A solution to this problem is an adaptive transform, where the window size N_{win} changes each time step [44]. Its size is dependent on the current window's force characteristics. A second option to solve this issue is a wavelet transform, which looks at the similarity with a predefined wave signal. This wavelet can be scaled and adjusted when the window is sliding over the signal, depending on the current characteristics. Finally, a multi-resolution STFT can be applied, which uses various windows sizes depending on the current signal frequency. These three solutions to a fixed window size have not been implemented yet in frequency domain filtering for robot collision detection, but can be explored in future work.

C-1-2 Setting a threshold

Efficient calculation of the dynamic threshold based on standard deviation The dynamic threshold based on standard deviation σ, as introduced in Section 1-IV-B, requires computation of the standard deviation of estimated force $\hat{\boldsymbol{F}}_{\text{ext}}$ over a sliding window. The conventional algorithm calculates σ over the full dataset for each time instant. This is computationally expensive and unnecessary since each time step only one force value is added to the window, and one removed. A scheme that efficiently calculates the standard deviation, the *rolling variance*, is introduced in [45]. Let the dataset of estimated force values, of one of the force components x y or z, be $(\hat{F}_{\text{ext}}(t-N-1), \dots, \hat{F}_{\text{ext}}(t-1))$ at time step $t-1$. The sliding window has size N. At time t, value $\hat{F}_{\text{ext}}(t)$ is added to the set and $\hat{F}_{\text{ext}}(t-N-1)$ removed, obtaining $(\hat{F}_{\text{ext}}(t-N), \dots, \hat{F}_{\text{ext}}(t))$. The mean $\bar{\hat{F}}_{\text{ext}}$ of the estimated forces, defined at time step t, is as follows

$$\bar{\hat{F}}_{\text{ext}}(t) = \bar{\hat{F}}_{\text{ext}}(t-1) + \frac{\hat{F}_{\text{ext}}(t) - \hat{F}_{\text{ext}}(t-N-1)}{N}. \tag{C-2}$$

Thereafter, the variance σ^2 is calculated as

$$\sigma^2(t) = \sigma^2(t-1) + \left(\hat{F}_{\text{ext}}(t) - \hat{F}_{\text{ext}}(t-N-1)\right) \frac{\hat{F}_{\text{ext}}(t) - \bar{\hat{F}}_{\text{ext}}(t) + \hat{F}_{\text{ext}}(t-N-1) - \bar{\hat{F}}_{\text{ext}}(t-1)}{N-1}. \tag{C-3}$$

The standard deviation is denoted as $\sigma(t) = \sqrt{\sigma^2(t)}$. Note that these calculations are executed for each of the three force components x y and z separately.

C-1-3 Detecting the collision

Detection with external forces instead of torques In Chapter 1 and Chapter 2, collisions are detected by observing when forces $\hat{\boldsymbol{F}}_{\text{ext,arm}}, \hat{\boldsymbol{F}}_{\text{ext,base}}$ cross their threshold, instead of the generalized torques $\hat{\boldsymbol{\tau}}_{\text{ext}}$. One of the reasons given is that the vector $\hat{\boldsymbol{\tau}}_{\text{ext}}$ can have larger values along the base components compared to the arm components during an arm collision. In this scenario, an arm collision is not detected on the arm torques $\hat{\boldsymbol{\tau}}_{\text{ext}}$. On the other hand, the arm and base forces computed with Eq. (1-8) capture the variations in all the components of $\hat{\boldsymbol{\tau}}_{\text{ext}}$. This is because the feet and arm Jacobians in the stacked Jacobian in Eq. (1-8) contain terms > 0 for the first six column entries related to the base. To clarify this, the Jacobians of the legs and arm are provided in this section.

The translational geometric Jacobian of the local frame of foot joint f, j, expressed in the world frame, is defined as follows for the first out of four feet [8, 46]

$${}^{\text{f},1}\mathbf{J} = \Big[\underbrace{\mathbf{I}_3 \quad -{}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}}\mathbf{S}(\boldsymbol{d}_{\mathcal{B}\text{ to f},1})}_{\text{base}} \quad \underbrace{\mathbf{J}_{v,\text{f},1}}_{\text{foot 1}} \quad \underbrace{\mathbf{0}_{3\times 9}}_{\text{feet 2,3,4}} \quad \underbrace{\mathbf{0}_{3\times 6}}_{\text{arm}}\Big], \tag{C-4}$$

where $\mathbf{S}(\boldsymbol{d}_{\mathcal{B}\,\text{to}\,\text{f},j}) \in \mathbb{R}^{3\times3}$ is the skew symmetric matrix associated to the position $\boldsymbol{d}_{\mathcal{B}\,\text{to}\,\text{f},j}$ from base to local joint frame f, j. The position is expressed in the world frame. Furthermore, $\mathbf{J}_{v,\text{f},j} \in \mathbb{R}^{3\times3}$ is the Jacobian expressing the linear velocities of frame f, j in world frame. Note that the Jacobian is 3×3 in size because the leg has 3 Degrees of Freedom (DoF). If the translational Jacobian is not expressed for foot 1, but for one of the other feet, the Jacobian $\mathbf{J}_{v,\text{f},j}$ will shift to the right of the matrix. Finally, ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}} \in \mathbb{R}^{3\times3}$ is the rotation matrix from base $\mathcal{B}$ to world frame $\mathcal{W}$. This transformation is required since the angular base torques in $\hat{\boldsymbol{\tau}}_{\text{ext}}$ are expressed in the base frame for the legged manipulator used in this work. The spatial Jacobian, expressed in the world frame, from the world to the local frame of arm link j, is defined as follows

$$
{}^{j}\mathbf{J}_{\mathcal{W}} = \begin{bmatrix} \mathbf{I}_3 & -{}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}}\mathbf{S}(\boldsymbol{d}_{\mathcal{B}\,\text{to}\,j}) & \mathbf{0}_{3\times12} & \mathbf{J}_{v,j} & \mathbf{0}_{3\times(6-j)} \\ \mathbf{0}_{3\times3} & {}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}} & \mathbf{0}_{3\times12} & \mathbf{J}_{\omega,j} & \mathbf{0}_{3\times(6-j)} \end{bmatrix}, \tag{C-5}
$$

where $\mathbf{J}_{\omega,j} \in \mathbb{R}^{3\times j}$ is the Jacobian expressing the angular velocities of frame j in world frame. Note that j can range from 1 to 6 in case of a 6-DoF arm.

The linear and angular Jacobians of link frame n are formulated as [47]

$$
\begin{bmatrix} \mathbf{J}_{v,n} \\ \mathbf{J}_{\omega,n} \end{bmatrix} = \begin{bmatrix} {}^{\mathcal{W}}\mathbf{R}_{i-1}{}^{i-1}\mathbf{R}_0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times (\boldsymbol{d}_{0\,\text{to}\,n} - \boldsymbol{d}_{0\,\text{to}\,i-1}) \\ {}^{\mathcal{W}}\mathbf{R}_{i-1}{}^{i-1}\mathbf{R}_0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix}, \tag{C-6}
$$

where i ranges from 1 to n and ${}^{i-1}\mathbf{R}_0 \in \mathbb{R}^{3\times3}$ is the rotation matrix from inertial frame 0, which is the frame associated to the first link in line, to frame $i-1$. The distances from frame 0 to $i-1$ are expressed by $\boldsymbol{d}$ and defined in frame 0. The vector $\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ denotes rotation around the z-axis, however, if a different axis is defined as the rotation axis, the vector changes. All local rotation matrices are multiplied by rotation matrix ${}^{\mathcal{W}}\mathbf{R}_{i-1} \in \mathbb{R}^{3\times3}$ to transform the expression to world frame notation. Note that for the feet Jacobians, the kinematic chain $1, \ldots, n$ ranges from from the base to one of the feet: $1, \ldots, 3$. Depending on for which of the 6 arm links j the Jacobian is calculated, i ranges from $1, \ldots, j$. Inertial frame 0 is the shoulder frame of the manipulator.

To summarize, because the base terms are included in the feet and arm Jacobians, the estimated arm force $\hat{\boldsymbol{F}}_{\text{ext,arm}}$ can observe variations in the base components of $\hat{\boldsymbol{\tau}}_{\text{ext}}$. Additionally to $\hat{\boldsymbol{F}}_{\text{ext,arm}}$, base force $\hat{\boldsymbol{F}}_{\text{ext,base}}$ is used for detection. Not all base collisions are reflected in $\hat{\boldsymbol{F}}_{\text{ext,arm}}$ and vice versa. Therefore, to make the detection algorithm robust, collisions are detected by observing when either the base or arm force crosses their threshold.

Proposed collision detection flowchart The detection method, as proposed in Section 2-III-A, is more extensively described with the flowchart in Figure C-3. For every time step, the block scheme for one of the three phases is followed, until a trigger is received to move on to the next phase. Then, the next time instant, the new phase's flowchart is followed. In the initial phase, phase 0, a collision is detected once a force crosses the threshold and additionally, the main direction of the force ($k \in x, y, z$) and sign ($+$ or $-$) of the detected peak is saved. Moving on to phase 1, now that the start of the external force has been detected, the collision keeps being observed ($\epsilon = 1$) until the end of the contact is indicated. This occurs when the force on axis k crosses the threshold again. Finally, in phase 2, it is examined if the filtered forces in all three directions are below the threshold for at least T_{rippling} s. This is meant to make the algorithm robust in case the effect *rippling* occurs, where due to backlash, incorrectly tuned controller gains, a high impact short-lasting force, or added load, the filtered force bounces above and below its boundary. This phenomenon has been observed before for band-pass filter (BPF) forces in [48], but no solution was provided. A second robustness check is added in phase 1, where the time is counted that the forces in all directions are below their thresholds. If this is longer than T_{no2peaks} s, it is decided that the second peak did not appear, the collision boolean ϵ is set to 0 again and the flowchart shifts back to phase 0. Note that during our experiments, the two-peak phenomenon appeared for external forces with a time span as short as 0.3 s.

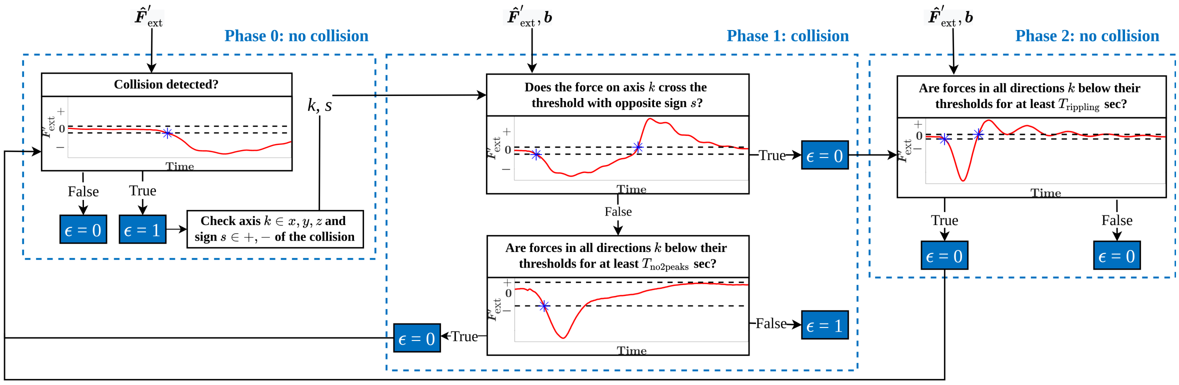


Figure C-3: Collision detection flowchart with as inputs filtered force $\boldsymbol{F}'_{\text{ext}}$ and threshold $\boldsymbol{b}$ and as an output collision bool ϵ. For every time instant, the block scheme is followed for one of the three phases, with initial phase 0.

C-2 Collision isolation

Collision isolation flowchart To isolate the colliding link, which for the legged manipulator in this work is either the base, forearm, or upperarm, the flowchart in Figure C-4 is designed. Step 1 decides if a collision occurs by observing the arm and base forces, as explained in Section C-1-3. In step 2, the scheme observes if a collision occurs on the arm by noticing variations in the first four arm torques $\hat{\boldsymbol{\tau}}_{\text{ext}}$. If these four torques do not cross their threshold, a collision occurs on the base. Additionally, a distinction between the forearm and upperarm is made by observing if arm torque 3 or 4 crosses their threshold. However, as explained in Section 1-VI-D, it is reasonable to assume that the contact link cannot always be isolated accurately. Therefore, in Chapter 2, this additional forearm-upperarm isolation step is excluded.

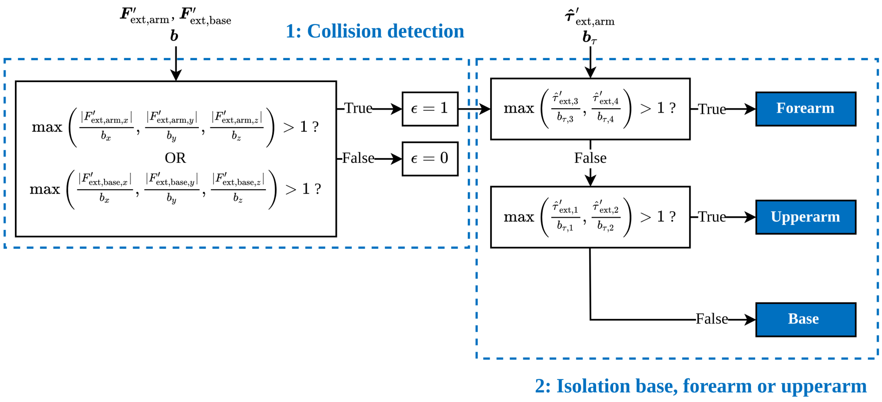


Figure C-4: Collision isolation flowchart consisting of two steps: (1) detecting the collision by following the block scheme of Figure C-3 for both arm and base force; (2) looking at torques $\hat{\tau}_{\text{ext}}$ belonging to the arm and deciding if a collision occurs on the forearm, upperarm or base.

C-3 Collision identification

Manipulator singularities As mentioned in Chapter 1 and Chapter 2, singularities of the manipulator are avoided during experiments. If the arm is in a singular configuration, the end-effector becomes blocked in certain directions and it consequently loses one or more controllable DoFs [47, 49]. In this scenario, small desired velocities commanded by the controller will result in infeasibly high joint velocity and torque references. In the worst case, an external impact force can not be sensed at all if the force lies completely within the transposed Jacobian's nullspace [50].

Obvious singular configurations occur when the arm reaches the limit of its workspace [47]. However, a 6-DoF manipulator has two other common singularities that can not always be easily avoided: the wrist and shoulder singularity [51], depicted in Figure C-5. The *wrist singularity* occurs when the axes of joints 4 and 6 coincide, meaning the wrists are fully stretched. In this case, the robot is blocked to move in the direction of the axis of joint 5. The arm is said to be in a *shoulder singularity* when the center of the robot wrist, joint 5, lies above joints 1 and 2, and when additionally the axis of joint 5 is parallel to the axis of joint 2. Consequently, the robotic arm cannot move in the direction of joint axis 2.

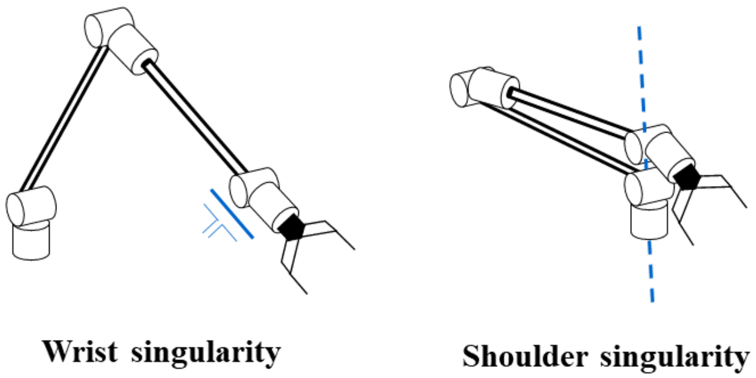


Figure C-5: Singularities of the 6-DoF manipulator DynaArm.

Appendix D

Experimental procedure

In this appendix, the properties of the F/T sensor, gripper and robot used in this work are introduced in Section D-1. Thereafter, Section D-2 describes the calibration routine for the F/T sensor. In Section D-3, the collisions applied to the robot during the experiments, as listed in Table 1-1 and Table 2-1, are split up into different datasets. Finally, Section D-4 discusses the tuned parameters for each collision handling method.

D-1 Properties F/T sensor, gripper and four-legged robot

BOTA Rokubi SensONE 6 DoF F/T sensor The inertial and technical properties of the force/torque (F/T) sensor are listed in Table D-1 [52]. One sensor is placed behind the gripper and one sensor is held in the hand to obtain the ground-truth collision force.

Table D-1: Technical specifications of the BOTA Rokubi SensONE 6 DoF F/T sensor.

Range ($F_{x,y}, F_z, M_{x,y}, M_z$)	$500\,\mathrm{N}, 1200\,\mathrm{N}, 15\,\mathrm{Nm}, 12\,\mathrm{Nm}$
Overload ($F_{x,y}, F_z, M_{x,y}, M_z$)	$2500\,\mathrm{N}, 4500\,\mathrm{N}, 35\,\mathrm{Nm}, 40\,\mathrm{Nm}$
Noise free resolution (100 Hz)	$0.3\,\mathrm{N}, 0.3\,\mathrm{N}, 0.0007\,\mathrm{Nm}, 0.0025\,\mathrm{Nm}$
Weight	$\sim 220\,\mathrm{g}$
Size	$70 \times 35\,\mathrm{mm}$
Sampling rate (max)	800 Hz

RobotiQ gripper The properties of the RobotiQ 2F-85 gripper [53] are given in Table D-2.

ANYmal The robot ANYmal is a four-legged robotic system that can carry payloads of up to 10 kg [54, 49]. Its technical specifications are found in Table D-3.

Table D-2: Technical specifications of the RobotiQ 2F-85 gripper.

Stroke	85 mm
Grip force	20 to 235 N
Form-fit grip payload	5 kg
Friction grip payload	5 kg
Gripper weight	0.9 kg
Closing speed	20 to 150 mm/s

Table D-3: Technical specifications of ANYmal.

Size lying L x W x H	$1054 \times 630 \times 376$ mm
Size standing L x W x H	$1054 \times 520 \times 830$ mm
Weight	50 kg
Operating temperature range	$0 - 40\,°\mathrm{C}$
Maximum walking speed	1 m/s

D-2 Calibration procedure F/T sensor

In this work, two F/T sensors are used. One is placed behind the RobotiQ gripper to measure the forces and torques of a possible object that the gripper is holding during a manipulation task. Before starting the experiments, a calibration routine is followed which moves the manipulator to convenient configurations. The configurations cover multiple positions and orientations of the end-effector. The result of this procedure are the calibration parameters: force offset $\boldsymbol{F}_{\mathrm{ft,offset}}$, torque offset $\boldsymbol{m}_{\mathrm{ft,offset}}$, distance from the F/T sensor to the gripper's center of mass (COM) $\boldsymbol{d}_{\mathrm{s\,to\,com}}$, and mass of the gripper m_{grip}. Note that the latter is specified in Table D-2, but a more accurate value is obtained in this routine. To transform the measured wrench ${}^{\mathcal{L}}\hat{\boldsymbol{\mathcal{F}}}_{\mathrm{ft}} = ({}^{\mathcal{L}}\hat{\boldsymbol{F}}_{\mathrm{ft}}, {}^{\mathcal{L}}\hat{\boldsymbol{m}}_{\mathrm{ft}}) \in \mathbb{R}^6$, defined in local frame $\mathcal{L}$, to final wrench ${}^{\mathcal{W}}\boldsymbol{\mathcal{F}}_{\mathrm{ft}}$, defined in world frame $\mathcal{W}$, four steps are followed [55].

Removing the offset. First, the offset is removed

$$\begin{aligned} {}^{\mathcal{L}}\hat{\boldsymbol{F}}_{\mathrm{ft}} &= {}^{\mathcal{L}}\hat{\boldsymbol{F}}_{\mathrm{ft}} - \boldsymbol{F}_{\mathrm{ft,offset}}, \\ {}^{\mathcal{L}}\hat{\boldsymbol{m}}_{\mathrm{ft}} &= {}^{\mathcal{L}}\hat{\boldsymbol{m}}_{\mathrm{ft}} - \boldsymbol{m}_{\mathrm{ft,offset}}. \end{aligned} \tag{D-1}$$

Filtering the raw sensor readings. Since the sensor measurements are noisy, a low-pass filter (LPF) is added to the measurements

$$\begin{aligned} {}^{\mathcal{L}}\hat{\boldsymbol{F}}'_{\mathrm{ft}}(t) &= {}^{\mathcal{L}}\hat{\boldsymbol{F}}'_{\mathrm{ft}}(t-1) + \left({}^{\mathcal{L}}\hat{\boldsymbol{F}}_{\mathrm{ft}}(t) - {}^{\mathcal{L}}\hat{\boldsymbol{F}}'_{\mathrm{ft}}(t-1)\right)\alpha, \\ {}^{\mathcal{L}}\hat{\boldsymbol{m}}'_{\mathrm{ft}}(t) &= {}^{\mathcal{L}}\hat{\boldsymbol{m}}'_{\mathrm{ft}}(t-1) + \left({}^{\mathcal{L}}\hat{\boldsymbol{m}}_{\mathrm{ft}}(t) - {}^{\mathcal{L}}\hat{\boldsymbol{m}}'_{\mathrm{ft}}(t-1)\right)\alpha, \end{aligned} \tag{D-2}$$

where α is a constant set to 0.4398.

Transforming from sensor frame to world frame. Using rotation matrix ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{L}} \in \mathbb{R}^{3\times3}$, defining rotation from the local F/T sensor frame to the world frame, the wrench is defined as

$$\begin{aligned} {}^{\mathcal{W}}\hat{\boldsymbol{F}}'_{\mathrm{ft}} &= {}^{\mathcal{W}}\mathbf{R}_{\mathcal{L}}\,{}^{\mathcal{L}}\hat{\boldsymbol{F}}'_{\mathrm{ft}}, \\ {}^{\mathcal{W}}\hat{\boldsymbol{m}}'_{\mathrm{ft}} &= {}^{\mathcal{W}}\mathbf{R}_{\mathcal{L}}\,{}^{\mathcal{L}}\hat{\boldsymbol{m}}'_{\mathrm{ft}}. \end{aligned} \tag{D-3}$$

Compensating for gravity. Finally, the measured wrench is compensated by the weight of the gripper

$$
\begin{aligned}
{}^{\mathcal{W}}\boldsymbol{F}_{\text{ft}} &= {}^{\mathcal{W}}\hat{\boldsymbol{F}}'_{\text{ft}} + \begin{bmatrix} 0 \\ 0 \\ -9.81 \cdot m_{\text{grip}} \end{bmatrix}, \\
{}^{\mathcal{W}}\boldsymbol{m}_{\text{ft}} &= {}^{\mathcal{W}}\hat{\boldsymbol{m}}'_{\text{ft}} + \boldsymbol{d}_{\text{s to com}} \times \begin{bmatrix} 0 \\ 0 \\ -9.81 \cdot m_{\text{grip}} \end{bmatrix}.
\end{aligned}
\tag{D-4}
$$

For the hand-held F/T sensor that is used to measure the ground-truth collision force, no gravity compensation is required. Additionally, there is no need to transform the read wrench to the world frame. For comparison between estimated and ground-truth force, the force magnitude is used, which is independent of the frame in which the wrench is defined. In any case, it is stated on the specification of the F/T sensor [52] that the only calibration needed is the offset subtraction. Before each collision, the F/T sensor is held steady close to the robot. In this way, the offset in each collision orientation is obtained and subtracted from the measured F/T values. No LPF is added since it is crucial not to have any delay on the measurements. Consequently, the detection delay of each of the detection and identification methods can be accurately computed. Finally, it should be noted that the measurements of the hand-held F/T sensor are recorded in a different `rosbag` than the collision handling and end-effector F/T sensor variables. To synchronize the time intervals of both, each experiment starts with a collision between the hand-held and the end-effector sensor. In this way, the measurements can be synced during the post-processing of the data.

D-3 Datasets

A detailed description of the experimental datasets from Table 1-1 and 2-1 that are used for analysis of the collision framework, is reported in Table D-4 and D-5. Note that for arm motion, the papers analyze a total of 40 collisions instead of the total of 49. Since datasets 6 and 14 consider collisions with a human arm, of which no ground-truth F/T sensor data is available, this is excluded from evaluation.

D-4 Selected parameters for collision handling methods

In this section, the parameters used in experiments are reported, and the tuning approaches are discussed.

D-4-1 Torque estimation

In Table D-6, the parameters for the continuous-time Momentum Based Observer (MBO) (Section B-2-2), the third-order MBO (Section B-2-4) and the Momentum Based Kalman Observer (MBKO) (Section B-2-5) are found.

Table D-4: Datasets of the collisions applied to the legged manipulator, as an extension of Table 1-1 and 2-1 (part 1).

Dataset	Number of collisions	Sampling time (in ms)	Controller	Base or arm collisions	Collisions during standstill, arm move or trot	Number of different arm positions	Comment
1	12	5.3	Whole-body MPC, EE mode	Arm	Standstill	1	-
2	19	5.3	Whole-body MPC, base mode	Arm	Standstill	1	-
3	11	5.3	Whole-body MPC, EE mode	Base	Standstill	1	-
4	45	3.4	Joint space PID	Arm	Standstill	5	Movement of arm when changing configuration, but no collisions during this time
5	22	2.5	Joint space PID	Base	Standstill	1	-
6	12	3.4	Joint space PID	Arm	Move	12	Last 3 collisions with human arm instead of F/T sensor, so no ground truth force is available
7	8	5.3	Whole-body MPC, EE mode	Arm	Move	8	-
8	60	2.5	Joint space PID	Arm	49 in standstill, 11 while moving	3 in standstill, 11 while moving	Unmodeled 0.58 kg load on forearm
9	14	3.4	Joint space PID	Arm	12 in standstill, 2 while moving	2 in standstill, 2 while moving	Unmodeled 0.58 kg load on forearm
10	46	2.5	Joint space PID	Arm	Standstill	3	Unmodeled 0.58 kg load hanging in gripper
11	79	2.5	Joint space PID	Arm	Standstill	6	Unmodeled 0.58 kg load hanging in gripper
12	16	2.5	Joint space PID	Arm	Standstill	2	Unmodeled 0.58 kg load on base
13	22	2.5	Joint space PID	Arm	Standstill	2	Unmodeled 2.0 kg load on base
14	6	2.5	Joint space PID	Arm	Move	6	Collisions with human arm instead of F/T sensor, so no ground-truth force is available
15	8	2.5	Joint space PID	Base	Move	8	Collisions with base while arm is moving
16	9	2.5	Joint space PID	Arm	8 in standstill, 1 while moving	1 in standstill, 1 while moving	Unmodeled 0.58 kg load fixed in gripper
17	15	2.5	Whole-body MPC, EE mode	Base	Move	15	Collisions with base while arm is moving
18	12	2.5	Whole-body MPC, base mode	Arm	Trot	1	Arm collisions during trotting
19	9	2.5	Joint space PID	Arm	Move	9	Unmodeled 0.58 kg load hanging in gripper

Momentum-based observers To select the gains for the continuous-time MBO in Eq. (B-7), the noise level of each of the 24 estimated external torques $\hat{\tau}_{\text{ext}}$ is examined over multiple datasets. The gains are chosen such that the standard deviation of each torque becomes no larger than 0.4 Nm. The arm and base components of $\hat{\boldsymbol{\tau}}_{\text{ext}}$ show the highest noise level. Consequently, gains k_{O} are chosen to be lower than those for the leg joints. Especially the base positional z-entry and angular y-entry give a noisy estimate. As a result, these gains are the lowest. For the discrete-time MBO in Eq. (B-16), the continuous-time observer gains are converted to the discrete-domain, as explained in Section 1-VI-A.

For the third-order MBO in Eq. (B-18), tuning proved to be slightly more challenging. Because of the higher-order filter, large values in either one of the three gain matrices result

Table D-5: Datasets of the collisions applied to the legged manipulator, as an extension of Table 1-1 and 2-1 (part 2).

Event		**Datasets**	**Number of collisions per dataset**	**Total number of collisions**
Standstill		1, 2, 3, 4, 5	12, 19, 11, 45, 22	109
Arm movement	F/T sensor	6, 7, 15, 17	9, 8, 8, 15	40
	With human arm	6, 14	3, 6	9
Trotting		18	12	12
Arm/base unmodeled load	In stance	8, 9, 12, 13	49, 12, 16, 22	99
	During move	8, 9	11, 2	13
Unmodeled load in gripper	In stance	10, 11, 16	46, 79, 8	133
	During move	16, 19	1, 9	10

Table D-6: Parameters of the torque estimation methods.

Method	Gain matrix	Diagonal of gain matrix
MBO	$\mathbf{K}_{\mathrm{O}}$	$\begin{bmatrix} 15.0 & 15.0 & 3.0 & 15.0 & 3.0 & 15.0 & 40.0 & 40.0 & 40.0 & 40.0 & 40.0 & 40.0 \\ 40.0 & 40.0 & 40.0 & 40.0 & 40.0 & 40.0 & 10.0 & 10.0 & 10.0 & 10.0 & 10.0 & 10.0 \end{bmatrix}^T$
MBO third-order	$\mathbf{K}_{\mathrm{O},1}$	$15.0 \cdot \mathbb{1}, \quad \text{with } \mathbb{1} \in \mathbb{R}^{24\times 1}$
	$\mathbf{K}_{\mathrm{O},2}$	$0.8 \cdot \mathbb{1}, \quad \text{with } \mathbb{1} \in \mathbb{R}^{24\times 1}$
	$\mathbf{K}_{\mathrm{O},3}$	$0.8 \cdot \mathbb{1}, \quad \text{with } \mathbb{1} \in \mathbb{R}^{24\times 1}$
MBKO	$\mathbf{A}_{\tau}$	$\mathbb{0}^T, \quad \text{with } \mathbb{0} \in \mathbb{R}^{24\times 1}$
	$\mathbf{Q}_p$	$\begin{bmatrix} 10.0 & 10.0 & 10.0 & 10.0 & 10.0 & 10.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 20.0 & 20.0 & 20.0 & 20.0 & 20.0 & 20.0 \end{bmatrix}^T$
	$\mathbf{Q}_{\tau}$	$10^3 \cdot \begin{bmatrix} 5.0 & 5.0 & 5.0 & 5.0 & 5.0 & 5.0 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 5.0 & 5.0 & 5.0 & 5.0 & 5.0 & 5.0 \end{bmatrix}^T$
	$\mathbf{Q}_{\dot{\boldsymbol{q}}}$	$10^{-3} \cdot \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 3.2 & 3.1 & 3.9 & 3.4 & 3.4 & 4.2 \\ 3.2 & 3.4 & 4.2 & 3.1 & 3.0 & 3.8 & 10.7 & 12.2 & 11.9 & 14.1 & 30.9 & 15.1 \end{bmatrix}^T$

in a torque estimate that does not converge. Therefore, the diagonal of $\mathbf{K}_{\mathrm{O},1}$ is set to the maximum value that still gives convergence. Lowering this term results in a slow response. Similarly, $\mathbf{K}_{\mathrm{O},2}$ and $\mathbf{K}_{\mathrm{O},3}$ are set to their maximum values.

Momentum-based Kalman observer Tuning the Kalman-based observer, defined in Section B-2-5, can be complex and time-consuming. However, [56] provides a calibration routine to select the matrices. First of all, the matrix $\mathbf{A}_{\tau}$ is typically chosen to be $\mathbf{0}_{24\times 24}$. Defining a negative diagonal allows eliminating constant offsets in the torques. Although offsets due to model inaccuracies are present, these vary over time and with configuration. Our experimental evaluation shows that it is not feasible to choose a negative value, and thus the matrix is set to $\mathbf{0}_{24\times 24}$. Covariance matrix $\mathbf{R}$ defines noise in measurements $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$. In [56], it is shown that noise in $\boldsymbol{q}$ is negligible compared to $\dot{\boldsymbol{q}}$. Thus, to determine the dependency of $\mathbf{R}$ on $\dot{\boldsymbol{q}}$, each arm and leg joint is moved separately at a constant speed. The velocity measurements can be estimated with a Gaussian distribution as $\dot{\boldsymbol{q}} \sim \mathcal{N}(\bar{\dot{\boldsymbol{q}}}, \mathbf{Q}_{\dot{\boldsymbol{q}}})$, where the diagonal of $\mathbf{Q}_{\dot{\boldsymbol{q}}}$ contains speed variances σ^2 of each joint. The variances obtained in the calibration experiment are found in Table D-6. Finally, the measurement noise matrix is calculated as

$$\mathbf{R} = \mathbf{M}\mathbf{Q}_{\dot{q}}\mathbf{M}^T, \tag{D-5}$$

for $\mathbf{M}$ the inertia matrix of the robot. In [56], an additional calibration routine is presented for $\mathbf{Q}_p$. However, it is based on a friction model and since friction modeling is out of the scope of this project, this tuning procedure could not be followed. For both $\mathbf{Q}_p$ and $\mathbf{Q}_\tau$, [56] advises to choose a positive diagonal. Covariance matrix $\mathbf{Q}_p$ presents the process noise in generalized momentum dynamics, which is mostly affected by modeling inaccuracies. Since in our work friction is not taken into account, these uncertainties are high. The largest values of $\mathbf{Q}_p$ are set for the arm joints, which are influenced by the unmodeled friction of a belt in the shoulder joint, and for the base, which has a high parametric uncertainty in the inertial terms. Increasing the gain means larger modeling errors are assumed. On the other hand, a higher gain increases delay, so a trade-off exists. Finally, the matrix $\mathbf{Q}_\tau$ describes the noise in the external torques: the larger the weight, the less the Kalman filter will rely on $\dot{\boldsymbol{\tau}}_{\text{ext}} = \mathbf{0}$. Thus, a high gain means that if a collision occurs, the response is fast. However, a larger weight results in noise amplification. Since a fast response is crucial to ensure safe collision detection, weights of $\mathbf{Q}_\tau$ are set high.

To compare the delay of the torque estimation methods, the collision detection approach of Chapter 2 is applied. Its selected parameters are found in the next section.

D-4-2 Collision detection

Comparison filtering methods The cut-off frequencies of the time- and frequency-domain filtering approaches are given in Section 1-VI-B. The constant thresholds set to calculate delay are $\begin{bmatrix}2.55 & 0.95 & 5.25\end{bmatrix}^T$ N and $\begin{bmatrix}2.15 & 0.8 & 4.1\end{bmatrix}^T$ N for the time-domain and frequency-domain filter, respectively.

Comparison thresholds As mentioned in Section 1-VI-C, the threshold parameters are selected based on the 17 arm collisions during arm movement. The tuned parameters are then fixed for the other collisions, independent of the following variations: arm or base collision, stance or arm motion, and with or without unmodeled payload.

For the constant threshold in Eq. (1-14), this results in the values $\begin{bmatrix}4.0 & 1.4 & 6.7\end{bmatrix}^T$ N. The tuned parameters for the dynamic threshold based on velocity in Eq. (1-15) are found in Table D-7. The gain vectors $\boldsymbol{k}_{\dot{q}}$ for the dynamic threshold are set to values of 2.0 for the legs and 5.0 for the arm joints. Because the force on the arm is affected most by the arm joint velocities, this value is higher. Still, if the legs move, the arm threshold should respond as well which is why the leg joint values are > 0. Note that the arm gains for the vector $\boldsymbol{k}_{\dot{q}}$ for the z-force are 20.0, because this component of the force shows to respond most significantly to motion, with high oscillations in $\hat{F}'_{\text{ext},z}$. The LPF cut-off frequencies for both dynamic thresholds are set higher than the cut-off frequencies for the band-pass filter (BPF) of the filtered force to prevent delay of the threshold. Finally, it is noted that in general, the maximum velocities $\dot{\boldsymbol{q}}_{\max}$ are obtained from the technical specifications or from the controller ranges. However, the former does not provide a maximum value for the velocity. Since in the experiments

Table D-7: Parameters of the dynamic threshold based on velocity for the arm.

Parameter	Value
Static threshold $\boldsymbol{b}_{\mathrm{stat},\dot{q}}$ (in N)	$\begin{bmatrix}2.0 & 1.1 & 3.0\end{bmatrix}^T$
$\dot{\boldsymbol{q}}_{\max}$ (in rad/s)	$\begin{bmatrix}0.1388 & 0.1301 & 0.4500 & 0.1442 & 0.1492 & 0.4406 & 0.1316 & 0.2473 & 0.1930 \\ & 0.2054 & 0.2366 & 0.1842 & 0.6489 & 0.6165 & 0.6577 & 0.5199 & 0.3093 & 0.6957\end{bmatrix}^T$
$\boldsymbol{k}_{\dot{q}}$ for $\hat{F}'_{\mathrm{ext},x}$	$\begin{bmatrix}2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 5.0 & 5.0 & 5.0 & 5.0 & 5.0 & 5.0\end{bmatrix}^T$
$\boldsymbol{k}_{\dot{q}}$ for $\hat{F}'_{\mathrm{ext},y}$	$\begin{bmatrix}2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 5.0 & 5.0 & 5.0 & 5.0 & 5.0 & 5.0\end{bmatrix}^T$
$\boldsymbol{k}_{\dot{q}}$ for $\hat{F}'_{\mathrm{ext},z}$	$\begin{bmatrix}2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 2.0 & 20.0 & 20.0 & 20.0 & 20.0 & 20.0 & 20.0\end{bmatrix}^T$
f_{c} for LPF (in Hz)	2.0

multiple controllers are used, $\dot{\boldsymbol{q}}_{\max}$ is obtained experimentally by taking the maximum values of $\dot{\boldsymbol{q}}$ over datasets 4, 6, 7 and 8.

Acquiring the maximum standard deviation $\boldsymbol{\sigma}_{\max}$, required for the dynamic threshold based on standard deviation in Eq. (1-16), is done by following the same procedure as for obtaining $\dot{\boldsymbol{q}}_{\max}$. Again, datasets 4, 6, 7, and 8 are used. The selected parameters for this dynamic threshold are given in Table D-8. The window size of the standard deviation dynamic threshold is chosen as 1.0 s after experimental evaluation. Reducing the window size gives too much variation in the threshold while increasing it makes the threshold move along with the force too little, giving an almost constant threshold as result. The maximum increase of the threshold is indicated with $\boldsymbol{F}_\sigma$. The z-force gives most oscillations during arm movement and shows the largest peaks, therefore, this value is set a bit higher than the x- and y-values. Because of these large peaks, the z-value of the static part is also set highest. Consequently, the constant c can be set lower for z than for x and y, because $b_{\mathrm{stat},\sigma,z}$ and $F_{\sigma,z}$ already compensate for the high force variation. The static part of the threshold in both Table D-7 and Table D-8 are tuned by increasing and decreasing the three components in steps of 0.1.

To compare the thresholds based on detection time, the estimated force $\hat{\boldsymbol{F}}_{\mathrm{ext}}$ is filtered with a BPF with cut-off frequencies $0.4 - 1.5$ Hz.

Table D-8: Parameters of the dynamic threshold based on standard deviation for the arm.

Parameter	Value
Static threshold $\boldsymbol{b}_{\mathrm{stat},\sigma}$ (in N)	$\begin{bmatrix}1.0 & 0.7 & 3.0\end{bmatrix}^T$
$\boldsymbol{\sigma}_{\max}$ (in N)	$\begin{bmatrix}22.3234 & 24.7723 & 17.0872\end{bmatrix}^T$
Window size (in s)	1.0
$\boldsymbol{F}_\sigma$ (in N)	$\begin{bmatrix}3.0 & 3.0 & 4.0\end{bmatrix}^T$
Constant $\boldsymbol{c}$	$\begin{bmatrix}45.0 & 35.0 & 20.0\end{bmatrix}^T$
f_{c} for LPF (in Hz)	3.0

Detecting the collision The final collision detection method as implemented in Chapter 2 consists of a BPF with three constant thresholds: one for stance, one for arm motion, and one for trotting. The parameters are given in Table D-9. The cut-off frequencies of the BPF during stance and arm motion are chosen as $0.4 - 3.0\,\text{Hz}$. A cut-off frequency as small as $\omega_{\text{min}} = 0.1\,\text{Hz}$ proved to remove the offset due to model inaccuracies. However, when two large external collision forces are applied within 2 s after each other, it turns out a higher ω_{min} is desired to pull the filtered force $\hat{F}'_{\text{ext}}$ towards 0 faster. If this is not the case, $\hat{F}'_{\text{ext}}$ might still be above its threshold during the start of the second collision. This is the case when the effect rippling occurs, for example. The value of $\omega_{\text{max}} = 3.0\,\text{Hz}$ is chosen in a trade-off between decreasing noise and increasing delay. Since trotting results in more high-frequency noise, ω_{max} is reduced. To assure distinction between high-frequency collisions and trotting, ω_{min} is decreased as well. With the tuned constant thresholds as defined in Table D-9, the success rate of 99 % and precision of 98 % is achieved, as mentioned in Section 2-IV-C.

For the collision detection flowchart in Figure C-3 values of $T_{\text{rippling}} = 0.6\,\text{s}$ and $T_{\text{no2peaks}} = 2.5\,\text{s}$ are chosen.

Table D-9: Constant thresholds and BPF cut-off frequencies for the detection approach.

	Constant threshold $\boldsymbol{b}$ (in N)	BPF cut-off frequencies $\omega_{\text{min}}, \omega_{\text{max}}$ (in Hz)
Stance	$\begin{bmatrix}1.8 & 1.8 & 1.8\end{bmatrix}$	$0.4, 3.0$
Arm motion	$\begin{bmatrix}4.0 & 3.0 & 6.5\end{bmatrix}$	$0.4, 3.0$
Trotting	$\begin{bmatrix}12.7 & 10.0 & 7.5\end{bmatrix}$	$0.4, 3.0$

D-4-3 Collision isolation

For distinction between base and arm, and between upperarm and forearm, the BPF cut-off frequencies for the external torques $\hat{\boldsymbol{\tau}}_{\text{ext}}$ are chosen equal to those of the force BPF: $[\omega_{\text{min}}, \omega_{\text{max}}] = [0.4, 3.0]$ Hz. As mentioned in Section 1-VI-D, the thresholds to localize a collision on arm or base are tuned such that none of the 56 base pushes are detected: $\begin{bmatrix}1.30 & 3.50 & 1.70 & 0.075 & 0.95 & 0.16\end{bmatrix}^T$ Nm, for each of the six filtered arm torques $\hat{\boldsymbol{\tau}}'_{\text{ext}}$. The thresholds for joint torques 3 and 4 are tuned on a set of arm collisions during stance, to values that make sure all forearm collisions are detected correctly: $\begin{bmatrix}0.8 & 0.12\end{bmatrix}^T$ Nm for the PID controller, and $\begin{bmatrix}1.8 & 0.11\end{bmatrix}^T$ Nm for the MPC controller. Note that the need for a different threshold, depending on which controller is running, again confirms the conclusion in Section 1-VI-D: it is reasonable to assume that a distinction between upperarm and forearm cannot always be made, especially using one threshold only.

Appendix E

Experimental results

In this appendix, additional results to those given in Chapter 1 and Chapter 2 are presented. Section E-1 highlights the collision detection step. First, the filtering technique comparison discussed in Section 1-VI-B is extended by comparing a frequency-domain filter based on a rectangular window with one based on a half-Hann window. Thereafter, it is demonstrated why the use of estimated forces give more accurate detection than external torques. Finally, the collision detection scheme introduced in Section 2-III-A and in Figure C-3 is verified. Section E-2 ends this appendix by validating the robustness of the identification method presented in Section 2-III-C, and by showing the effect of varying certain parameters (e.g. controller, configuration of robot, direction of force) over the 416 collisions.

E-1 Collision detection

E-1-1 Filtering the force

Comparing filtering methods in time- and frequency-domain In Section 1-VI-B, time- and frequency-domain filtering are compared. For frequency-domain filtering, the half-Hann window in Eq. (C-1) is applied and experimentally validated. On the other hand, the performance of the rectangular window has been verified in [57]. Therefore, in Figure E-1, a comparison of the three different band-pass filter (BPF) techniques is visualized. The minimum and maximum cut-off frequencies are set equal for each method to 0.6 Hz and 2.3 Hz, respectively. As expected, all filtering approaches remove the -3 N offset of the unfiltered force, which is caused by model inaccuracies, and reduce the noise of the unfiltered force. Comparing time-domain with frequency-domain filtering, it is noted that the latter results in more delay because of the sliding window, which takes into account previous measurements. This delay can be reduced by decreasing the time resolution of the window. On the other hand, the noise level is lowered using the frequency-domain approach. This is visible in Figure E-1: where the time-domain filter shows oscillations when the force ends, the frequency-domain filters give a smooth response. The rectangular window is excluded from further analysis since its delay is slightly larger than that of the half-Hann window.

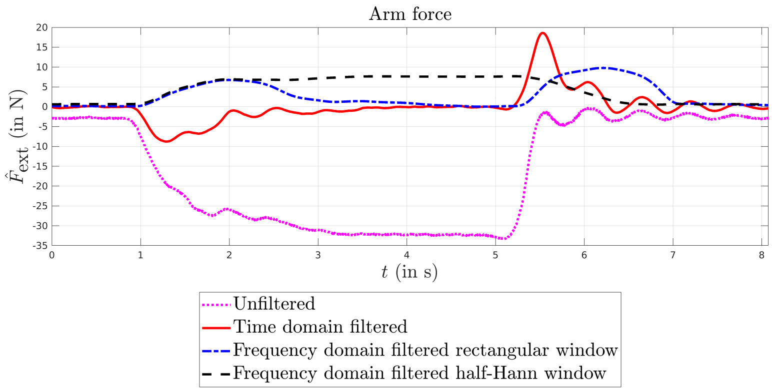
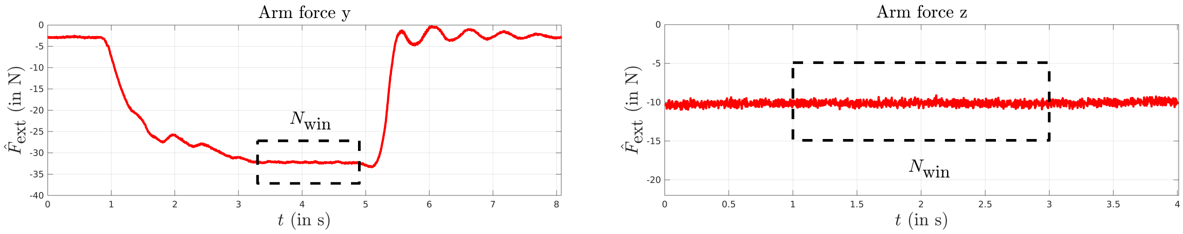


Figure E-1: A collision applied in the $-y$-direction in world frame from the side of the forearm. The difference in BPF filtering methods is shown, together with the unfiltered force.

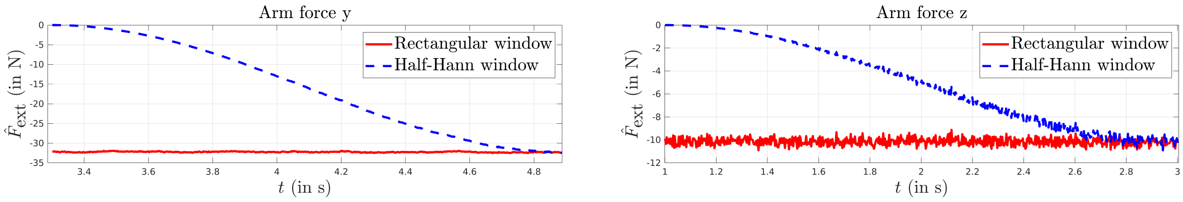
Half-Hann window technique It is striking that the two-peak phenomenon, which is discussed in Section 2-III-A, does not appear for the half-Hann window approach. This is further investigated in Figure E-2. First, the scenario of a long-lasting collision force is examined, as visualized in the three graphs on the left. Starting with the top-left graph, Figure E-2a, a window N_{win} is applied halfway the estimated collision force $\hat{F}_{\text{ext}}$. Here, the estimated force is almost constant and thus contains low-frequency content. Next, the vector of forces within window N_{win} is multiplied with a rectangular and a half-Hann window (see Figure C-1). The result is shown in the left graph of Figure E-2b. Since the half-Hann window is based on a cosine function, the force is weighed from 0 (oldest measurement) to 1 (current measurement). Finally, the Fast Fourier Transform (FFT) is taken over the windowed force in Figure E-2b to obtain the frequency spectrum, resulting in Figure E-2c. As can be seen, using a rectangular window, the values for the power $\hat{F}_{\omega}$ in the frequency range above 0.8 Hz are equal to 0. On the other hand, due to the hanning, the frequency spectrum of the half-Hann window contains values larger than 0 in the higher frequency range. To apply a BPF to the windowed force, the cut-off frequencies for this example are selected as $[\omega_{\text{min}}, \omega_{\text{max}}] = [2.4, 4.8]$Hz. Taking the 1-norm over this frequency spectrum (Eq. (1-13)), the value of the filtered force $\hat{F}'_{\omega_{\text{min}}:\omega_{\text{max}}}$ with hanning is 7.60 N, while it is 0.04 N for the rectangular window.

A similar phenomenon can be seen in the three graphs on the right in Figure E-2, however, here the focus is on the z-component of the estimated external force $\hat{F}_{\text{ext}}$. The robot is at standstill, and no collision is applied. In Figure E-2a, it can be seen that the z-force is constant around -10 N, which is the offset due to model inaccuracies. When applying a half-Hann window, this offset is lowered but not completely removed. Contrary, the use of a rectangular window or a filter in time domain does eliminate this offset. In the right graphs of the example in Figure E-2, the values of the final 1-norm are 2.39 N for the half-Hann window and 0.03 N for the rectangular. To conclude, the half-Hann window can be used as a BPF, but is not able to filter out the offset due to nominal motion and modeling errors completely. Furthermore, it does also not show the two-peak phenomenon during a collision.

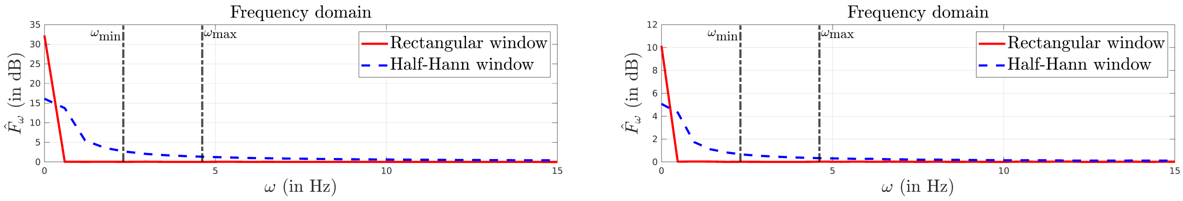
Despite the offset in force estimation that can only be partly eliminated using the half-Hann window, this frequency-domain filter shows the same BPF characteristics as the time-domain variant. Looking at Figure E-3d, it is noticed that not all, but a large part of the offset in estimated force is removed. Additionally, high-frequency noise in the x-force in Figure E-3b is



(a) Estimated collision force $\hat{F}_{\text{ext}}$ on the y-axis *(left)* and offset during standstill on the z-axis *(right)* where the FFT is taken over window with length N_{win}.



(b) The force values in the window with length N_{win} are multiplied with either a rectangular or half-Hann window, where the half-Hann window gives more weight to the current force measurements than the past.



(c) The frequency spectrum of the force values multiplied with the windows.

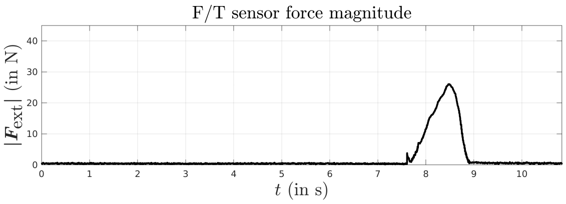
Figure E-2: The concept of the rectangular and half-Hann window is visualized in case of a collision force on the y-axis *(left)* or during a constant offset in standstill on the z-axis *(right)*. When filtering with the rectangular window, the filtered force value $\hat{F}'_{\omega_{\text{min}}:\omega_{\text{max}}}$ is in both scenarios 0, while the half-Hann window gives a value $\hat{F}'_{\omega_{\text{min}}:\omega_{\text{max}}} > 0$.

filtered out. Furthermore, the undesired ripples in the y-component of the force in Figure E-3c between $t = 9\,\text{s}$ and $t = 11\,\text{s}$, which are caused by the shaky response of the arm after a hard collision, are eliminated.

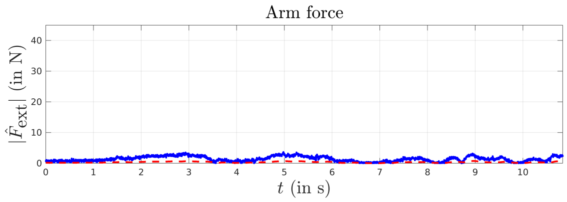
A possible benefit of using the half-Hann window frequency-domain filtering over time-domain filtering is that the filtered force does not show the two-peak phenomenon during a collision. The two peaks can be advantageous to mark the start and ending of a collision, however, in case of high-impact short-lasting collisions (e.g. shorter than 0.3 s), only one peak might appear. If mostly these types of collisions are expected, filtering in the frequency domain can be more robust. However, in that case, the proposed collision detection and identification methods of Section 2-III cannot be applied. Furthermore, in our work, impacts shorter than 0.3 s are not considered.

E-1-2 Detecting the collision

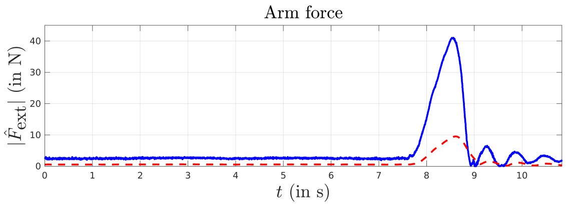
Using forces instead of torques for detection In Section C-1-3, an explanation for the use of forces instead of torques for detection is given. In this paragraph, this choice is further



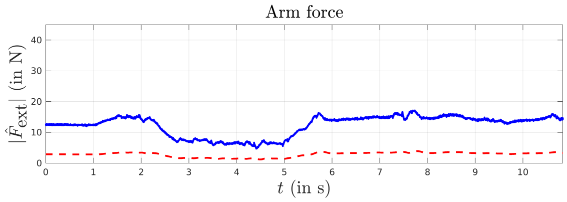
(a) Ground truth force/torque (F/T) sensor collision force.



(b) Force in x-direction.



(c) Force in y-direction.

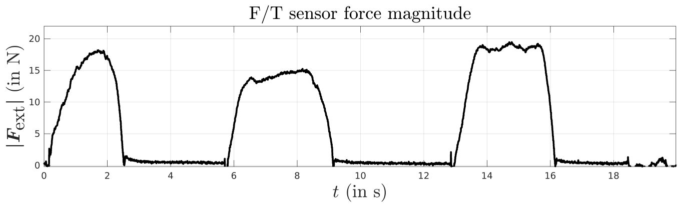


(d) Force in z-direction.

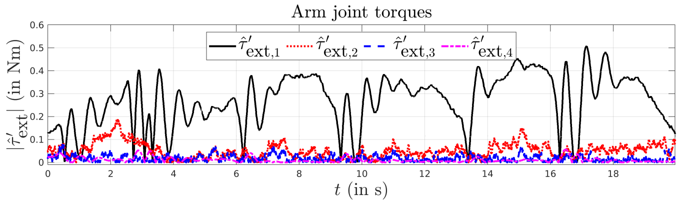
Figure E-3: A collision applied from the side of the forearm. The unfiltered estimated force is visualized together with a frequency-domain band-pass filtered force, using the half-Hann window.

validated. Looking at the arm collisions during stance in dataset 4 (see Table D-4), 44 out of 45 collisions are detected using the arm force, while only 40 out of 45 pushes are detected on the arm torques. Note that for the force and torques, the cut-off frequencies of the BPF are set equal to $[\omega_{\text{min}}, \omega_{\text{max}}] = [0.4, 3.0]$ Hz. The inability to detect the collision occurs for short-lasting forces $(0.5 - 0.7\,\text{s})$, although the magnitude of these undetected forces increases up to 40 N, and for forces acting on the robot in the y-direction on the side of the upperarm. These kinds of collisions are common, and a 40 N force can be harmful. Thus, it is undesired that they are not detected. The three contacts applied at the side of the upperarm, which are undetected in the joint torques, are visualized in Figure E-4a. The response to these collisions of the first four arm joint torques $\hat{\boldsymbol{\tau}}_{\text{ext}}$, is shown in Figure E-4b. No threshold is illustrated inFigure E-4b, but in previous experiments, it shows to be around 1.1 Nm for joint 1 and at least 0.3 Nm for joints 2 to 4. It is clearly visible that the three collisions are not detected. Next, the estimated filtered force is examined in Figure E-4c, which is able to detect the impacts.

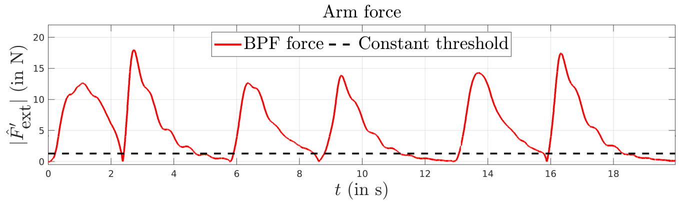
Validation detection approach In this paragraph, the detection approach from Section 2-III-A is further validated. Looking at the detection flowchart in Figure C-3, it is noticed that phase 2 is added such that the detection approach is robust against so-called 'rippling'. This effect is illustrated in Figure E-5b, where after the contact has ended, the estimated BPF force $\hat{F}'_{\text{ext}}$ keeps moving below and above the detection thresholds. In Figure E-5, since the force lasts for a short time only, 0.5 s, the contact is intense and the response of the robot is more significant than during a slow, long-duration force. Consequently, the robot arm shakes a bit after the collision, causing the force rippling. The conventional collision detection method in Figure E-5a detects multiple short collisions, long after the external collision force has disappeared. However, the proposed detection method demonstrates to be robust against this effect.



(a) Magnitude ground truth F/T sensor force.

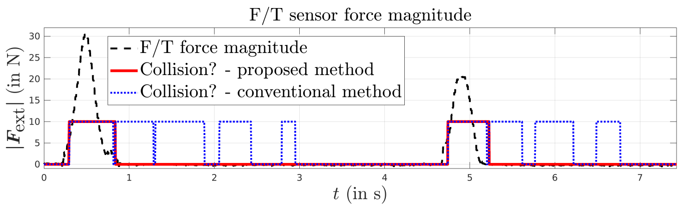


(b) The first four filtered arm joint torques.

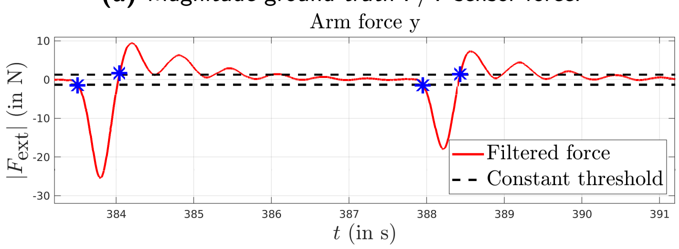


(c) Filtered force in y-direction.

Figure E-4: Three collisions applied on the side of the upperarm, in standstill. The filtered arm joint torques in (b) do not detect the three collisions visualized in (a) since the threshold is higher than any of the values visible. In (c), it is clear that the collisions are detected using the force as a detection measure.



(a) Magnitude ground truth F/T sensor force.



(b) Filtered force in y-direction.

Figure E-5: Two collisions applied on the side of the forearm, in standstill. In graph (a), the ground truth collision force is shown together with the binary value indicating if a collision occurred, for both the conventional and proposed method. It is shown that the proposed method is robust against the rippling effect. In (b), the filtered force is plotted with the start and end of the collisions, as detected with the novel approach, marked with the stars.

E-2 Collision identification

The robustness of the collision identification approach, as presented in Section 2-III-C, is validated in Section E-2-1. Furthermore, additional state-of-the-art comparison are made, extending the comparisons in Section 2-IV-C. Finally, in Section E-2-2, the effect that factors such as controller and arm configuration have on the force estimation, is analyzed.

E-2-1 Additional results

Performance The final collision force, as defined in Section 2-II-C, is given as $\hat{\boldsymbol{F}}_{\mathrm{c}}(k) = \hat{\boldsymbol{F}}_{\mathrm{ext}}(k) - \hat{\boldsymbol{F}}_{\mathrm{dis}}(k)$. In Figure E-6, these three forces are visualized. It can be seen that during detection of collision (*green line*), the estimated disturbance force (*pink line*) stays frozen. Continuously, the disturbance is subtracted from the estimated force and gives an accurate final collision force estimation (*blue line*).

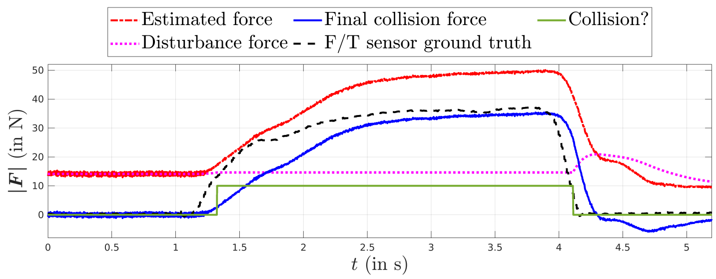
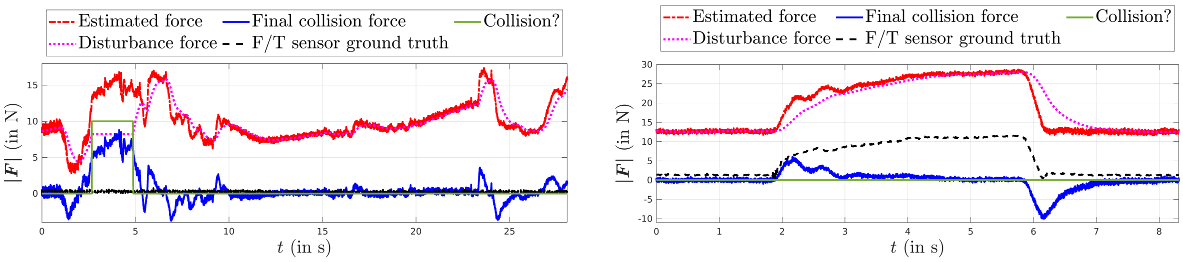


Figure E-6: A collision applied from the top of the forearm, with payload placed on the arm. The disturbance is subtracted from the estimated force, to obtain the final collision force.

As presented in Section 2-IV-C, the number of false positive (FP)s and false negative (FN)s using the proposed detection approach is low: a 99 % success rate and a 98 % precision is achieved over the 416 collisions. However, it is interesting to see what the estimation of the collision force $\hat{\boldsymbol{F}}_{\mathrm{c}}$ gives in case an FP or FN does occur. In Figure E-7a, an FP occurs during arm motion. Consequently, the disturbance force $\hat{\boldsymbol{F}}_{\mathrm{dis}}$ is frozen. The final estimated collision force rises to 8 N. In the scenario where a reaction strategy is implemented, this force value is likely not to trigger any reaction since it is harmless and only lasts 2 s. Next, in Figure E-7b, a collision acts on the robot and is not detected. The force $\hat{\boldsymbol{F}}_{\mathrm{c}}$ increases to 5 N at first, but is pulled towards 0 N afterwards. Since no collision is detected, no reaction is initiated. However, only slow, low-impact external forces are undetectable, and these do usually not form a large risk for either robot or human.

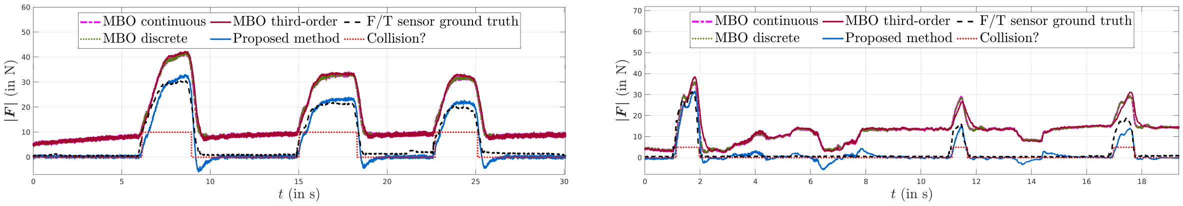
State-of-the-art comparison A detailed analysis of the proposed identification method performance in two challenging scenarios, load on the arm and trotting, is reported in Chapter 2. Extending this analysis, two additional case studies are presented in Figure E-8: collisions during stance and arm motion. In Figure E-8a, the offset due to model inaccuracies is around 10 N. Comparing this to the offset with a load placed on the forearm in Figure 2-9a, it is lower, but the same estimation accuracy is achieved. This implies that the force is identified accurately independent of the weight of the load. Secondly, force identification during arm motion is shown in Figure E-8b, where the estimated force is oscillating due to the movement



(a) An FP occurs during arm motion.

(b) An FN occurs for a slow, low-impact interaction force.

Figure E-7: The detection approach detects an FP and FN. The effect on the final estimated collision force is shown.



(a) Three arm collisions in stance.

(b) Three arm collisions during arm motion.

Figure E-8: Comparison of external collision forces estimated with the proposed and state-of-the-art identification methods.

of the arm. Using the proposed method, the offset is removed and the variations in the force between $t = 4\,\mathrm{s}$ and $t = 8\,\mathrm{s}$ are reduced compared to the state-of-the-art approaches.

E-2-2 Varying parameters

In Section 2-IV-C, a few factors of variations during the experiments are presented: mode (stance, arm motion, trotting), added load, and duration and magnitude of the force. In fact, the following parameters are additionally varied during the experiments:

- **Controller.** For control of the arm, base, and legs, two controllers are available: a whole-body joint space Proportional, Integral, Derivative (PID) controller with gravity and friction compensation and a whole-body Model Predictive Control (MPC) controller [58]. The latter controls the position of the gripper when it is in end-effector (EE) mode and the movements of the base in base mode.
- **Direction and contact location of the force.**
- **Configuration.** The response to a collision is position-dependent: in some arm configurations, the legs respond to the push by bending, while in others the arm gives a more intense reaction.

Below, it is discussed how these parameters influence force estimation.

Effect of the controller, direction, and contact location The whole-body controllers active during collisions control the robot to keep or move towards a certain configuration and differ in what part of the robot they control. Using the PID, each joint can be moved separately, while using the MPC in base or EE mode, a positional and angular command is sent to the base and end-effector, respectively. A second difference between these controllers is the proportional gains. To compare the estimation accuracy of collision forces using each controller, similar collisions are evaluated. All contacts are applied from the side of the upperarm or forearm, with force magnitude and duration within the same range for each controller.

Figure E-9 shows an overview of the relative estimation error. This is equal to the absolute error in Eq. (1-18), without taking the absolute value. A negative value indicates underestimation, while a positive value means that the force is overestimated. The PID has the largest proportional gains set on the arm joints, followed up by MPC base mode and MPC EE mode. As a result of these large gains, the arm is kept stiff and the static friction is high. Consequently, the joint torque values for all arm joint torques are large and thus increase the force estimation and relative error. This effect was also noticed during the experiments, where the arm was shaking in response to a collision using MPC EE mode, due to the higher compliance.

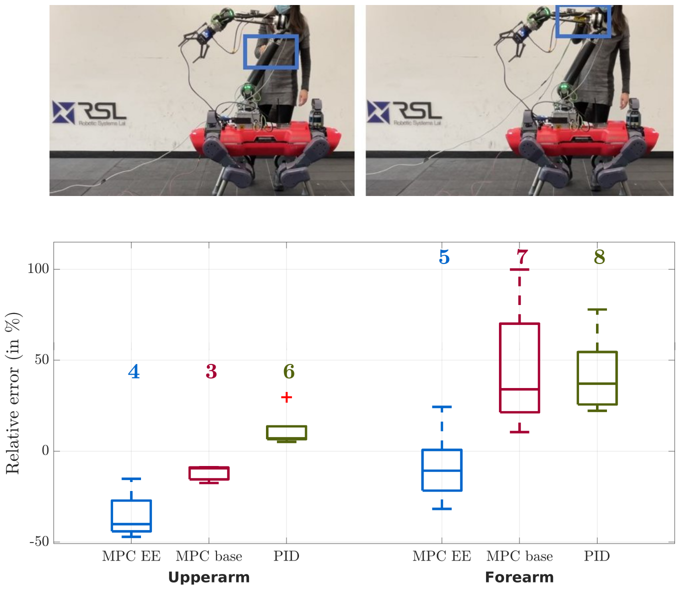


Figure E-9: Comparison of the relative force estimation error between three controllers and between collision occurring on two different body links. The number above each box indicates the amount of observations analyzed. The magnitudes of the contact forces vary between $13 - 25\,\mathrm{N}$.

Next, collisions occurring on the upperarm and forearm are compared. In Figure E-9, the forearm shows higher relative errors than the upperarm, meaning the force is overestimated more. Impact forces occurring on links closer to the base (e.g. upperarm compared to forearm) give larger null spaces in the stacked contact Jacobian, making part of the force undetectable. Not only the contact location, but also the direction of the force affects the estimation. Looking at all the 416 collisions, it is noticed that forces applied on the arm in minus z-direction in the world frame are generally underestimated. A possible explanation is the soft ground on which the robot is standing, which might absorb part of the force. Additionally, the robot shows the most redundancy in the z-direction, resulting in a transfer of the force to not only the arm components but also the legs and the base. The underestimation of base forces can be explained by the base cap, which bends when pressing the ground-truth F/T sensor on it, and thus part of the force is transferred into the material.

Effect of the configuration In three different configurations, similar forces are applied to the side of the upperarm, visualized in Figure E-10. All collisions are obtained from the same dataset, to minimize the effect of other factors. It is noted that the estimation characteristics are closely related. This indicates that the configuration of the arm does not have a significant effect. In the first configuration, however, the forces are underestimated slightly more compared to the other two. This can be explained by the close-to-singular configuration of the arm since the wrist joint is positioned almost straight above arm joints 1 and 2 (wrist singularity, see Section C-3). Consequently, part of the contact force gets lost in the null space of the stacked contact Jacobian. To conclude on the relationship between configuration and force estimation, more datasets like these should be examined to see if similar effects occur.

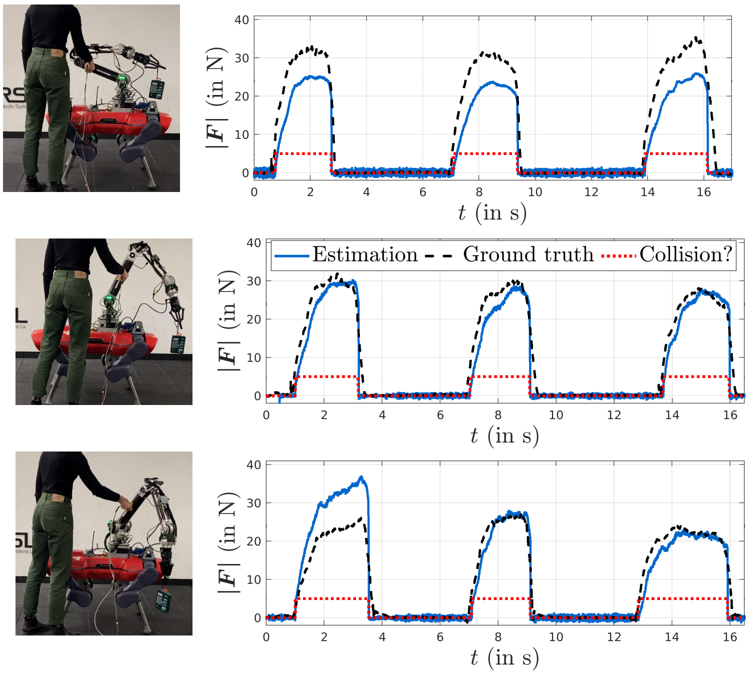


Figure E-10: Three collisions applied on the side of the upperarm in three different configurations, with an object in the gripper. The force is estimated and compared with the ground truth F/T sensor collision force.

Bibliography

[1] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot collisions: a survey on detection, isolation, and identification," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292–1312, 2017.

[2] S. Golz, C. Osendorfer, and S. Haddadin, "Using tactile sensation for learning contact knowledge: discriminate collision from physical interaction," in *IEEE International Conference on Robotics and Automation*, pp. 3788–3794, 2015.

[3] S. Haddadin, "Physical safety in robotics," in *Formal Modeling and Verification of Cyber-Physical Systems*, pp. 249–271, Springer, 2015.

[4] G. Buondonno and A. De Luca, "Combining real and virtual sensors for measuring interaction forces and moments acting on a robot," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 794–800, 2016.

[5] M. Iskandar, O. Eiberger, A. Albu-Schaffer, A. De Luca, and A. Dietrich, "Collision detection, identification, and localization on the DLR SARA robot with sensing redundancy," in *IEEE International Conference on Robotics and Automation*, pp. 3111–3117, 2021.

[6] A. De Luca and R. Mattone, "Sensorless robot collision detection and hybrid force/motion control," in *IEEE International Conference on Robotics and Automation*, pp. 999–1004, 2005.

[7] N. Likar and L. Žlajpah, "External joint torque-based estimation of contact information," *International Journal of Advanced Robotic Systems*, vol. 11, no. 107, pp. 1–11, 2014.

[8] J. Vorndamme, M. Schappler, and S. Haddadin, "Collision detection, isolation and identification for humanoids," in *IEEE International Conference on Robotics and Automation*, pp. 4754–4761, 2017.

[9] S. Faraji and A. J. IJspeert, "Designing a virtual whole body tactile sensor suit for a simulated humanoid robot using inverse dynamics," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 5564–5571, 2016.

[10] T. Ito, K. Ayusawa, E. Yoshida, and A. Kheddar, "Experimental study for controller-friendly contact estimation for humanoid robot," in *IEEE Workshop on Advanced Robotics and its Social Impacts*, vol. 2019, pp. 28–33, 2019.

[11] T. Mattioli and M. Vendittelli, "Interaction force reconstruction for humanoid robots," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 282–289, 2017.

[12] J. Bimbo, C. Pacchierotti, N. G. Tsagarakis, and D. Prattichizzo, "Collision detection and isolation on a robot using joint torque sensing," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 7604–7609, 2019.

[13] L. Manuelli and R. Tedrake, "Localizing external contact using proprioceptive sensors: the contact particle filter," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 5062–5069, 2016.

[14] G. Cioffi, S. Klose, and A. Wahrburg, "Data-efficient online classification of human-robot contact situations," in *European Control Conference*, pp. 608–614, 2020.

[15] N. Briquet-Kerestedjian, A. Wahrburg, M. Grossard, M. Makarov, and P. Rodriguez-Ayerbe, "Using Neural Networks for classifying human-robot contact situations," in *European Control Conference*, pp. 3279–3285, 2019.

[16] J. Liang and O. Kroemer, "Contact localization for robot arms in motion without torque sensing," *IEEE International Conference on Robotics and Automation*, pp. 6322–6328, 2021.

[17] A. Zwiener, C. Geckeler, and A. Zell, "Contact point localization for articulated manipulators with proprioceptive sensors and machine learning," in *IEEE International Conference on Robotics and Automation*, pp. 323–329, 2018.

[18] D. Popov and A. Klimchik, "Real-time external contact force estimation and localization for collaborative robot," in *IEEE International Conference on Mechatronics*, pp. 646–651, 2019.

[19] Z. Qiu, R. Ozawa, and S. Ma, "Adaptive virtual power-based collision detection and isolation with link parameter estimation," *Advanced Robotics*, vol. 34, no. 12, pp. 814–825, 2020.

[20] A. De Luca, A. Albu-Schäffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the DLR-III lightweight manipulator arm," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 1623–1630, 2006.

[21] A. De Luca and L. Ferrajoli, "Exploiting robot redundancy in collision detection and reaction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3299–3305, 2008.

[22] S. Haddadin, "Towards safe robots: approaching Asimov's 1st law," *Springer Tracts in Advanced Robotics*, vol. 90, pp. 1–352, 2014.

[23] M. Jorda, E. G. Herrero, and O. Khatib, "Contact-driven posture behavior for safe and interactive robot operation," in *IEEE International Conference on Robotics and Automation*, pp. 9243–9249, 2019.

[24] L. Sentis and O. Khatib, "A whole-body control framework for humanoids operating in human environments," in *IEEE International Conference on Robotics and Automation*, pp. 2641–2648, 2006.

[25] M. Geravand, F. Flacco, and A. De Luca, "Human-robot physical interaction and collaboration using an industrial robot with a closed control architecture," in *IEEE International Conference on Robotics and Automation*, pp. 4000–4007, 2013.

[26] S. A. B. Birjandi and S. Haddadin, "Model-adaptive high-speed collision detection for serial-chain robot manipulators," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6544–6551, 2020.

[27] S. A. B. Birjandi, J. Kuhn, and S. Haddadin, "Observer-extended direct method for collision monitoring in robot manipulators using proprioception and IMU sensing," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 954–961, 2020.

[28] A. Zelenak, M. Pryor, and K. Schroeder, "An extended Kalman Filter for collision detection during manipulator contact tasks," in *ASME Dynamic Systems and Control Conference*, pp. 1–10, 2014.

[29] A. N. Sharkawy, P. N. Koustoumpardis, and N. Aspragathos, "Neural network design for manipulator collision detection based only on the joint position sensors," *Robotica*, vol. 38, no. 10, pp. 1737–1755, 2020.

[30] F. Dimeas, L. D. Avendaño-Valencia, and N. Aspragathos, "Human-robot collision detection and identification based on fuzzy and time series modelling," *Robotica*, vol. 33, no. 9, pp. 1886–1898, 2015.

[31] T.-Y. Lin, R. Zhang, J. Yu, and M. Ghaffari, "Deep multi-modal contact estimation for invariant observer design on quadruped robots," *arXiv preprint arXiv:2106.15713*, 2021.

[32] D. Lim, D. Kim, and J. Park, "Momentum observer-based collision detection using LSTM for model uncertainty learning," in *IEEE International Conference on Robotics and Automation*, pp. 4516–4522, 2021.

[33] S. Mamedov and S. Mikhel, "Practical aspects of model-based collision detection," *Frontiers in Robotics and AI*, vol. 7, no. 1, pp. 1–15, 2020.

[34] P. C. Hammer, "The midpoint method of numerical integration," *Mathematics Magazine*, vol. 31, no. 4, pp. 193–195, 1958.

[35] G. Bledt, P. M. Wensing, S. Ingersoll, and S. Kim, "Contact model fusion for event-based locomotion in unstructured terrains," in *IEEE International Conference on Robotics and Automation*, pp. 4399–4406, 2018.

[36] L. F. Chaparro, "Introduction to the design of discrete filters," *Signals and Systems using MATLAB*, vol. 1, no. 1, pp. 639–707, 2011.

[37] V. Morlando, A. Teimoorzadeh, and F. Ruggiero, "Whole-body control with disturbance rejection through a momentum-based observer for quadruped robots," *Mechanism and Machine Theory*, vol. 164, no. 2, pp. 1–20, 2021.

[38] S. He, J. Ye, Z. Li, S. Li, G. Wu, and H. Wu, “A momentum-based collision detection algorithm for industrial robots,” in *IEEE International Conference on Robotics and Biomimetics*, pp. 1253–1259, 2015.

[39] T. Ren, Y. Dong, D. Wu, and K. Chen, “Collision detection and identification for robot manipulators based on extended state observer,” *Control Engineering Practice*, vol. 79, no. 1, pp. 144–153, 2018.

[40] A. Wahrburg, E. Morara, G. Cesari, B. Matthias, and H. Ding, “Cartesian contact force estimation for robotic manipulators using Kalman filters and the generalized momentum,” in *IEEE International Conference on Automation Science and Engineering*, pp. 1230–1235, 2015.

[41] N. Solutions, “White paper: Understanding FFTs and windowing,” tech. rep., NI Solutions, Austin, Texas, 2021.

[42] A. Kouris, F. Dimeas, and N. Aspragathos, “A frequency domain approach for contact type distinction in human-robot collaboration,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 720–727, 2018.

[43] N. Kehtarnavaz, *Digital Signal Processing System Design.* Academic Press, 2 ed., 2008.

[44] C. Mateo and J. A. Talavera, “Bridging the gap between the short-time Fourier transform (STFT), wavelets, the constant-Q transform and multi-resolution STFT,” *Signal, Image and Video Processing*, vol. 14, no. 8, pp. 1535–1543, 2020.

[45] T. F. Chan, G. H. Golub, R. J. Leveque, T. F. Chan, G. H. Golub, and R. J. Leveque, “Algorithms for computing the sample variance : analysis and recommendations,” *The American Statistician*, vol. 37, no. 3, pp. 242–247, 1983.

[46] M. Mistry, J. Buchli, and S. Schaal, “Inverse dynamics control of floating base systems using orthogonal decomposition,” in *IEEE International Conference on Robotics and Automation*, pp. 3406–3412, 2010.

[47] M. Hutter, R. Siegwart, and T. Stastny, “Robot dynamics (course notes for ETH Zürich 151-0851-00L),” 2018.

[48] C. N. Cho, J. H. Kim, Y. L. Kim, J. B. Song, and J. H. Kyung, “Collision detection algorithm to distinguish between intended contact and unexpected collision,” *Advanced Robotics*, vol. 26, no. 16, pp. 1825–1840, 2012.

[49] C. D. Bellicoso, K. Kramer, M. Stauble, D. Sako, F. Jenelten, M. Bjelonic, and M. Hutter, “ALMA - Articulated locomotion and manipulation for a torque-controllable robot,” in *IEEE International Conference on Robotics and Automation*, pp. 8477–8483, 2019.

[50] P. Cao, Y. Gan, and X. Dai, “Model-based sensorless robot collision detection under model uncertainties with a fast dynamics identification,” *International Journal of Advanced Robotic Systems*, vol. 16, no. 3, pp. 1–15, 2019.

[51] M. Vaezi, F. C. Samavati, H. E. S. Jazeh, and S. A. A. Moosavian, “Singularity analysis of 6DOF Stäubli TX40 robot,” in *IEEE International Conference on Mechatronics and Automation*, pp. 446–451, 2011.

[52] Bota Systems, "Bota Systems force torque sensors." https://www.botasys.com/, 2022. Accessed: 19-02-2022.

[53] ROBOTIQ, "RobotiQ Gripper 2F-85." https://robotiq.com/products/2f85-140-adaptive-robot-gripper, 2022. Accessed: 19-02-2022.

[54] ANYbotics, "ANYmal." https://www.anybotics.com/, 2022. Accessed: 19-02-2022.

[55] C. Erdogan, M. Zafar, and M. Stilman, "Gravity and drift in force / torque measurements," tech. rep., Georgia Institute of Technology, 2014.

[56] A. Wahrburg, J. Bös, K. D. Listmann, F. Dai, B. Matthias, and H. Ding, "Motor-current-based estimation of Cartesian contact forces and torques for robotic manipulators and its application to force control," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 2, pp. 879–886, 2018.

[57] A. Kouris, F. Dimeas, and N. Aspragathos, "Contact distinction in human-robot cooperation with admittance control," in *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1951–1956, 2016.

[58] J. P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, "A unified MPC framework for whole-body dynamic locomotion and manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4688–4695, 2021.

Glossary

List of Acronyms

RSL	Robotic Systems Lab
ETH	Eidgenössische Technische Hochschule
DoF	Degrees of Freedom
F/T	force/torque
IMU	Inertial Measurement Unit
EKF	Extended Kalman Filter
MBO	Momentum Based Observer
MBKO	Momentum Based Kalman Observer
NN	Neural Network
pHRI	physical Human-Robot Interaction
BPF	band-pass filter
LPF	low-pass filter
FP	false positive
FN	false negative
FFT	Fast Fourier Transform
STFT	Short-time Fourier Transform
MPC	Model Predictive Control
PID	Proportional, Integral, Derivative
EE	end-effector
COM	center of mass
LTI	linear time-invariant

List of Symbols

α	Constant in an LPF related to the cut-off frequency ω as $\alpha = e^{-\omega T_s}$
β	Observer gain value of the discrete-time MBO: $\beta = ((1-\gamma)\gamma^{-1})/T_\mathrm{s})$
ϵ	Detection bool: $\epsilon = 1$ if collision, $\epsilon = 0$ if no collision
Γ	Observer gain matrix (positive, diagonal) of the discrete-time MBO
γ	Observer gain value of the discrete-time MBO
ω_max	Maximum cut-off frequency
ω_min	Minimum cut-off frequency
$\dot{\boldsymbol{\tau}}_\mathrm{ext}$	Time evolution of external torques
$\boldsymbol{\sigma}$	Standard deviations of the estimated external force $\hat{\boldsymbol{F}}_\mathrm{ext}$
$\boldsymbol{\sigma}^2$	Variance of the estimated external force $\hat{\boldsymbol{F}}_\mathrm{ext}$
$\boldsymbol{\sigma}_\mathrm{max}$	Maximum deviations of the estimated external force $\hat{\boldsymbol{F}}_\mathrm{ext}$
$\boldsymbol{\tau}_\mathrm{ext}$	External torques
$\boldsymbol{\tau}_\mathrm{ft}$	External torques resulting from measurements of F/T sensors on the robot structure
$\boldsymbol{\tau}_\mathrm{m}$	Motor joint torques
$\dot{\boldsymbol{p}}$	Time evolution of generalized momentum
$\boldsymbol{g}(\boldsymbol{q})$	Gravity vector
$\boldsymbol{n}(\boldsymbol{q}, \boldsymbol{v})$	Nonlinear terms, quantity used for ease of notation: $\boldsymbol{n}(\boldsymbol{q}, \boldsymbol{v}) := \boldsymbol{g}(\boldsymbol{q}) - \mathbf{C}^T(\boldsymbol{q}, \boldsymbol{v})\boldsymbol{v}$
$\Delta\omega$	Frequency resolution frequency-domain filter
Δt	Time resolution frequency-domain filter
$\hat{\mathbf{P}}$	Estimate of the covariance matrix used in the Kalman filter
$\mathbf{A}$	State-space matrix of continuous-time system
$\mathbf{A}_\mathrm{d}$	State-space matrix of discrete-time system
$\mathbf{B}$	Observer gain matrix (positive, diagonal) of the discrete-time MBO
$\mathbf{B}$	State-space matrix of continuous-time system
$\mathbf{B}_\mathrm{d}$	State-space matrix of discrete-time system
$\mathbf{C}$	State-space matrix of continuous-time system
$\mathbf{C}(\boldsymbol{q}, \boldsymbol{v})$	Matrix capturing centripetal and Coriolis effects
$\mathbf{C}_\mathrm{d}$	State-space matrix of discrete-time system
$\mathbf{J}_{\omega,n}$	Jacobian expressing the angular velocities of frame n
$\mathbf{J}_{\mathrm{f},j}$	Geometric translational Jacobian of contact foot j
$\mathbf{J}_i$	Geometric spatial Jacobian of colliding point i
$\mathbf{J}_{v,n}$	Jacobian expressing the linear velocities of frame n
$\mathbf{K}$	Kalman gain
$\mathbf{K}_{\mathrm{O},i}$	Observer gain matrix (positive, diagonal) of the continuous-time MBO of order r, with $i = 1, \ldots, r$
$\mathbf{K}_\mathrm{O}$	Observer gain matrix (positive, diagonal) of the continuous-time MBO
$\mathbf{M}(\boldsymbol{q})$	Inertia matrix

$\mathbf{Q}$	Process noise covariance matrix defining noise in generalized momentum dynamics and external torques: $\mathbf{Q} = \mathrm{diag}\left(\mathbf{Q}_p, \mathbf{Q}_\tau\right)$
$\mathbf{Q}_\mathrm{d}$	Discretized process noise covariance matrix
$\mathbf{R}$	Measurement noise covariance matrix
$\mathbf{R}_\mathrm{d}$	Discretized measurement noise covariance matrix
$\mathbf{S}$	Actuation matrix
$\mathbf{S}(\boldsymbol{d})$	Skew symmetric matrix of position $\boldsymbol{d}$
$\mathcal{B}$	Base frame
$\mathcal{L}$	Local F/T sensor frame
$\mathcal{W}$	World frame
$\dot{\boldsymbol{q}}_\mathrm{des}$	Desired velocities
$\dot{\boldsymbol{q}}_\mathrm{max}$	Maximum velocities
$\dot{\boldsymbol{v}}$	Generalized accelerations
$\hat{\boldsymbol{\tau}}'_\mathrm{ext}$	Time-domain filtered estimated external torques
$\hat{\boldsymbol{F}}'_\omega$	Frequency-domain filtered estimated external force
$\hat{\boldsymbol{F}}'_\mathrm{ext}$	Time-domain filtered estimated external force
$\boldsymbol{\mathcal{F}}_{\mathrm{ext},i}$	Collision wrench on colliding body i: $\boldsymbol{\mathcal{F}}_{\mathrm{ext},i} = (\boldsymbol{F}_{\mathrm{ext},i}, \boldsymbol{m}_{\mathrm{ext},i})$
$\boldsymbol{\mathcal{F}}_\mathrm{ft}$	Wrench measured by a F/T sensor on the robot structure: $\boldsymbol{\mathcal{F}}_\mathrm{ft} = (\boldsymbol{F}_\mathrm{ft}, \boldsymbol{m}_\mathrm{ft})$
$\boldsymbol{\Phi}_\mathrm{b}$	Cartesian orientation of the base
$\boldsymbol{b}$	Threshold
$\boldsymbol{b}_{\mathrm{dyn},\dot{q}}$	Dynamic threshold based on velocity
$\boldsymbol{b}_{\mathrm{dyn},\sigma}$	Dynamic threshold based on standard deviation
$\boldsymbol{b}_\mathrm{stat}$	Static part of dynamic threshold
$\boldsymbol{d}_{a\text{ to }b}$	Distance from frame a to b
$\boldsymbol{F}_\omega$	The FFT of the external force
$\boldsymbol{F}_{\mathrm{ext},i}$	Collision force on colliding body i
$\boldsymbol{F}_\mathrm{ft}$	Force measured by a F/T sensor on the robot structure
$\boldsymbol{F}_{\mathrm{f},j}$	Force acting on contact foot j
$\boldsymbol{m}_{\mathrm{ext},i}$	Collision force on colliding body i
$\boldsymbol{m}_\mathrm{ft}$	Torque measured by a F/T sensor on the robot structure
$\boldsymbol{p}$	Generalized momentum
$\boldsymbol{q}$	Generalized coordinates
$\boldsymbol{r}_\mathrm{b}$	Cartesian position of the base
$\boldsymbol{u}$	Input vector continuous-time system dynamics
$\boldsymbol{v}$	Generalized velocities
$\boldsymbol{v_p}$	Measurement noise vector defining noise in generalized momentum
$\boldsymbol{w}$	Process noise vector defining noise in generalized momentum dynamics and external torques: $\boldsymbol{w} = \begin{bmatrix}\boldsymbol{w}_p^T & \boldsymbol{w}_\tau^T\end{bmatrix}^T$
$\boldsymbol{x}$	State vector continuous-time system dynamics
$\boldsymbol{y}$	Output vector continuous-time system dynamics

c	Constant to multiply the standard deviation with in the standard deviation dynamic threshold expression
e	Absolute error
F_σ	Constant determining the maximum amount that the standard deviation dynamic threshold can increase
$G(s)$	Transfer function
k	Time step in discrete time
k_{det}	Time instant of collision detection
N	Number of floating-base robot DoF: $N = 6 + n_{\text{j}}$
N_σ	Window size of dynamic threshold based on standard deviation
n_{ft}	Number of F/T sensors on robot structure
n_{j}	Number of actuated joints
N_{w}	Window size FFT
t	Time step in continuous time
T_{O}	Time constant
T_{s}	Sampling time
w_n	Window weighting the FFT function
z	Discrete-time variable
${}^a\mathbf{R}_b$	Rotation matrix from frame b to a
b	Base
c	Collision
des	Desired
dis	Disturbance
dyn	Dynamic
d	Discrete-time
ext	External
ft	The force/torque (F/T) sensor on robot structure
f	Foot
j	Joint
max	Maximum
min	Minimum
stat	Static
$\bar{f}$	Mean value
$\dot{f}$	Derivative
$\hat{f}$	Estimated value
f'	Filtered value
$f^{\#}$	Moore-Penrose pseudoinverse operation