Collision Detection with Energy Residual using a Velocity Observer

Gravina, Giovanbattista Nunziante, Luca

August 2023

Abstract

In this work we tackle the issue of collision detection, a key topic in the context of safe physical human-robot interaction. The problem is addressed in case of rigid robots via a scalar monitoring signal of the kinetic energy of the system. The proposed solution relies on proprioceptive sensors only and assumes that only the robot joint positions are available, while the velocities are estimated using a reduced-order observer. To provide a suitable input to such observer, an additional momentum-based residual signal is formulated in order to estimate the external torque generated during a collision. To highlight the effectiveness of the velocity observer, simulations compare this approach with the estimation based on numerical differentiation in nominal and uncertain conditions.

1 Introduction

Initially confined to factory floors, robots originally operated with little to no interaction with their human counterparts. On the other hand, in the last decades there has been a surge in the interest and production of collaborative robots, also known as cobots, which are smaller and lightweight with respect to industrial robots.

Cobots typically operate without safety cages sharing the workspace with the human operator in dynamic and unstructured environments. Since there is a human in the loop, the relative motions between them and the robot may be fast and unpredictable, so it is not always possible to avoid collisions, but it is critical to detect them and react accordingly.

Indeed, in a *collision event pipeline* [5] up to seven elementary phases can be identified and, after the *precollision phase* where collision-free motions are planned, the next phase is dedicated to *collision detection*, whose output should be a binary decision on whether a collision occurred or not. Finally, after having acquired more information on the collision, the robot should react purposefully, for example stopping or retracting.

In this work, the *collision detection* phase is addressed. In particular, collisions are viewed as faulty behaviors of the robot actuating system and — using a residual based approach — they are detected relying exclusively on proprioceptive sensors. Such approaches [2], [3] have the advantage of relying solely on the measure of joint positions and velocities, and do not require the knowledge of the acceleration. However, cobots — and robots in general — are typically equipped with encoders but lack tachometers, hence the velocity is usually estimated via finite differences. In the following, this approach is compared with the employment of a reduced-order observer [1] for velocity estimation, and their performance in the context of collision detection is analyzed.

This report is structured as follows. In Section 2 some properties and quantities are defined starting from the robot dynamic model. The signals used for collision detection are presented in Section 3, specifying their limitations and computational details. A reduced-order observer to estimate the robot joint velocity is illustrated in Section 4. In Section 5 the experimental setup is presented and different simulations are offered, and finally some conclusive remarks are drawn in Section 6.

2 Preliminaries

Consider a robot manipulator with n rigid joints arranged in an open kinematic chain. Given the generalized coordinates $\boldsymbol{q} \in \mathbb{R}^n$, the robot dynamics can be modeled as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}_m - \boldsymbol{\tau}_F + \boldsymbol{\tau}_{ext} = \boldsymbol{\tau}_{tot} \tag{1}$$

where $\boldsymbol{M}(\boldsymbol{q}) > 0 \in \mathbb{R}^{n \times n}$ is the inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} \in \mathbb{R}^n$ is the Coriolis and centripetal term, and $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$ is the gravity vector. On the right side of (1), $\boldsymbol{\tau}_{tot} \in \mathbb{R}^n$ is the sum of three components: $\boldsymbol{\tau}_m$ represents the commanded torque, $\boldsymbol{\tau}_F$ is the dissipative torque due to friction, and $\boldsymbol{\tau}_{ext}$ is a torque generated when the robot is interacting with the environment, e.g., in case of a collision.

Since $\boldsymbol{M}(\boldsymbol{q})$ is symmetric and positive definite, it holds that

$$0 < \lambda_1 ||\boldsymbol{x}||^2 \leq \frac{1}{2}\boldsymbol{x}^T \boldsymbol{M}(\boldsymbol{q})\boldsymbol{x} \leq \lambda_2 ||\boldsymbol{x}||^2 \quad \forall \boldsymbol{x}, \boldsymbol{q} \in \mathbb{R}^n \tag{2}$$

where

$$\lambda_1 = \min_{\boldsymbol{q} \in \mathbb{R}^n} \frac{\lambda_{min}(\boldsymbol{M}(\boldsymbol{q}))}{2}, \quad \lambda_2 = \max_{\boldsymbol{q} \in \mathbb{R}^n} \frac{\lambda_{max}(\boldsymbol{M}(\boldsymbol{q}))}{2} \ . \tag{3}$$

Moreover, in the factorization of the Coriolis and centripetal vector, the basic property of the skew-symmetry of the matrix $\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$, or equivalently

$$\dot{\boldsymbol{M}}(\boldsymbol{q}) = \boldsymbol{C}^T(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{4}$$

has been enforced.

Finally, a bounded and non-negative scalar function $0 \leq c_0(\boldsymbol{q}) \leq \bar{c}_0$ can be found such that

$$||\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})|| \leq c_0(\boldsymbol{q})||\dot{\boldsymbol{q}}|| \ . \tag{5}$$

To model $\boldsymbol{\tau}_{ext}$, let $\boldsymbol{x}_c \in \mathbb{R}^3$ be a point along the robot structure that is colliding with the environment, and $\boldsymbol{\mathcal{F}}_{ext} \in \mathbb{R}^6$ the collision wrench applied by the environment on the robot. Then, given the geometric Jacobian $\boldsymbol{J}_c(\boldsymbol{q}) \in \mathbb{R}^{6\times n}$ associated to the collision point, and assuming only one collision is occurring, the external joint torque due to $\boldsymbol{\mathcal{F}}_{ext}$ is given by

$$\boldsymbol{\tau}_{ext} = \boldsymbol{J}_c^T(\boldsymbol{q})\boldsymbol{\mathcal{F}}_{ext} \ . \tag{6}$$

The total energy E of the robot is given by the sum of its kinetic energy T and potential energy U_g due to gravity

$$E = T + U_g = \frac{1}{2}\dot{\boldsymbol{q}}^T\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} + U_g(\boldsymbol{q}) \tag{7}$$

with $\boldsymbol{g}(\boldsymbol{q}) = (\frac{\partial U_g(\boldsymbol{q})}{\partial \boldsymbol{q}})^T$. Deriving (7) and considering the dynamic model (1), we obtain the evolution over time of the total energy as

$$\dot{E} = \dot{\boldsymbol{q}}^T\boldsymbol{\tau}_{tot} \tag{8}$$

that represents the power balance in the system, including the external power $P_{ext} = \dot{\boldsymbol{q}}^T\boldsymbol{\tau}_{ext}$.

It is possible to define the generalized momentum of the robot $\boldsymbol{p}$ as

$$\boldsymbol{p} = \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{9}$$

and its time evolution, using (4), is

$$\dot{\boldsymbol{p}} = \boldsymbol{\tau}_{tot} + \boldsymbol{C}^T(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{g}(\boldsymbol{q}) \ . \tag{10}$$

3 Collision Detection Signals

The aim of the collision detection phase is to determine whether a physical collision occurs or not. In this section, we present a scalar signal and a vector signal that can be used for such purpose, and analyze their properties. In the following of this work, we make the assumption that friction is absent or compensated by control, hence $\boldsymbol{\tau}_F = \boldsymbol{0}$.

3.1 Energy-based residual

In case of a collision, the total energy of the system is expected to change following (8). Thus, it is reasonable to tackle the problem of collision detection with an energy-based approach [3]. Indeed, define the scalar residual as

$$\sigma(t) = k_\sigma \left(\hat{T}(t) - \int_0^t (\dot{\boldsymbol{q}}^T(\boldsymbol{\tau}_m - \hat{\boldsymbol{g}}(\boldsymbol{q})) + \sigma)ds - \hat{T}(0) \right) \tag{11}$$

where $\hat{T}(t)$ is the estimate of the robot kinetic energy at instant $t \geq 0$, $\hat{\boldsymbol{g}}(\boldsymbol{q})$ is the estimate of the gravity vector, k_σ is a positive gain and the residual initial

value is $\sigma(0) = 0$. The dynamics of the residual defined in (11), assuming ideal conditions with $E = \hat{E}$, is obtained by simple derivation as

$$\begin{aligned} \dot{\sigma}(t) &= k_\sigma \left(\dot{\hat{T}} + \dot{\hat{U}}_g - \dot{\boldsymbol{q}}^T \boldsymbol{\tau}_m - \sigma \right) \\ &= k_\sigma \left(\dot{E} - \dot{\boldsymbol{q}}^T \boldsymbol{\tau}_m - \sigma \right) \\ &= k_\sigma \left(P_{ext} - \sigma \right) \end{aligned} \tag{12}$$

where (8) has been used, together with the fact that

$$\int_0^t \dot{\boldsymbol{q}}^T \hat{\boldsymbol{g}}(\boldsymbol{q}) ds = \hat{U}_g(t) - \hat{U}_g(0) \ .$$

The dynamics obtained in (12) is the one of an asymptotically stable first-order linear filter, with time constant $1/k_\sigma$. Hence, $\sigma(t)$ will stay identically zero in free motion, approach exponentially P_{ext} in response to a collision, and decay back to zero when the collision is over.

Computational details The formulation in (11) is a convenient one, as it only requires the computation of the kinetic energy and the gravity vector, and each can be performed with a single call to the Newton-Euler algorithm with complexity $O(n)$. Moreover, the computation of $\sigma(t)$ requires only the commanded torque $\boldsymbol{\tau}_m$ that is generally known, the joint position $\boldsymbol{q}$ given by the encoders, and then the joint velocities $\dot{\boldsymbol{q}}$ that need to be estimated in the absence of tachometers.

Limitations Not all collisions can be detected by this simple scalar residual. Indeed, there are two cases in which detection may fail:

L1: collisions that do not generate external power are concealed, and this happens when

$$\dot{\boldsymbol{q}}^T \boldsymbol{\tau}_{ext} = \dot{\boldsymbol{q}}^T \boldsymbol{J}_c^T(\boldsymbol{q}) \boldsymbol{\mathcal{F}}_{ext} = \boldsymbol{V}_c^T \boldsymbol{\mathcal{F}}_{ext} = 0 \iff \boldsymbol{V}_c \perp \boldsymbol{\mathcal{F}}_{ext} \tag{13}$$

where $\boldsymbol{V}_c \in \mathbb{R}^6$ is the screw vector at the contact point $\boldsymbol{x}_c$;

L2: if the robot is at rest, then $\dot{\boldsymbol{q}}^T \boldsymbol{\tau}_{ext} = 0$ and the collision will not be detected. Notice that this can be seen as a particular case of L1;

However, if the position controller is not stiff, the robot may start moving in the direction dictated by $\boldsymbol{\mathcal{F}}_{ext}$ before the force is removed, and both limitations are overcome. Nonetheless, in both scenarios described in L1, L2, if the external force is truly impulsive the collision is not detected.

3.2 Momentum-based residual

Recalling the dynamics of the generalized momentum from (10), define

$$\boldsymbol{\beta}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \coloneqq \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{C}^T(\boldsymbol{q}, \dot{\boldsymbol{q}}) \dot{\boldsymbol{q}} \ . \tag{14}$$

The residual signal $\boldsymbol{r}(t) \in \mathbb{R}^n$ is defined as

$$\boldsymbol{r}(t) = \boldsymbol{K}_r \left(\hat{\boldsymbol{p}}(t) - \int_0^t (\boldsymbol{\tau}_m - \hat{\boldsymbol{\beta}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{r}) ds - \hat{\boldsymbol{p}}(0) \right) \tag{15}$$

where $\boldsymbol{K}_r > 0$ is a diagonal gain matrix, and $\hat{\boldsymbol{p}}(t) = \hat{\boldsymbol{M}}(\boldsymbol{q})\dot{\boldsymbol{q}}$. Using (10),(14) and assuming that $\hat{\boldsymbol{M}} = \boldsymbol{M}$ and $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}$, the dynamics of $\boldsymbol{r}(t)$ can be expressed in terms of $\boldsymbol{\tau}_{ext}$ as

$$\begin{aligned} \dot{\boldsymbol{r}}(t) &= \boldsymbol{K}_r(\boldsymbol{\tau}_{tot} + \boldsymbol{C}^T(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{\tau}_m + \boldsymbol{\beta}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{r}(t)) \\ &= \boldsymbol{K}_r(\boldsymbol{\tau}_{ext} - \boldsymbol{r}(t)) \ . \end{aligned} \tag{16}$$

Equivalently to (12), this is a stable first-order linear filter, and the signal $\boldsymbol{r}(t)$ can be seen as a first-order approximation of the collision torque $\boldsymbol{\tau}_{ext}$.

Computational details Apart from $\boldsymbol{\tau}_m, \boldsymbol{q}, \dot{\boldsymbol{q}}$, computing $\boldsymbol{r}(t)$ from (15) requires $\boldsymbol{p}(t)$, which can be obtained with a single call to the Newton-Euler algorithm with complexity $O(n)$ as before. However, to compute $\hat{\boldsymbol{\beta}}(\boldsymbol{q}, \dot{\boldsymbol{q}})$, the knowledge of the factorization matrix $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is needed. This can be obtained either with $3n$ calls to the standard Newton-Euler algorithm, or with n calls to the modified Newton-Euler algorithm [4] (an implementation of the latter is used in simulations). In any case, the complexity raises to $O(n^2)$.

Limitations Differently from before, since now the residual signal approaches directly $\boldsymbol{\tau}_{ext}$ and not $\dot{\boldsymbol{q}}^T\boldsymbol{\tau}_{ext}$, the limitations L1, L2 are overcome. Collisions that are not detected by $\boldsymbol{r}(t)$ — or by $\sigma(t)$ — are the ones that do not produce an external torque $\boldsymbol{\tau}_{ext}$, and this happens when

$$\boldsymbol{\tau}_{ext} = \boldsymbol{J}_c^T(\boldsymbol{q})\boldsymbol{\mathcal{F}}_{ext} = 0 \iff \boldsymbol{\mathcal{F}}_{ext} \in \ker(\boldsymbol{J}_c^T(\boldsymbol{q})) \ . \tag{17}$$

4 Reduced-order observer

Assuming $\boldsymbol{q}$ is available for measurement, following [1], we detail a reduced-order observer to estimate $\dot{\boldsymbol{q}}$. Differently from full-order estimators, a reduced-order observer exclusively estimates the state quantities that are unavailable, i.e., $\dot{\boldsymbol{q}}$ in this context. Such observers are characterized by a smaller state dimension and faster convergence rates with respect to full-order estimators.

Firstly, defining the robot state as $(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{q}, \dot{\boldsymbol{q}})$ and the output $\boldsymbol{y} = \boldsymbol{x}_1$, the robot dynamics can be expressed in the state space form as follows

$$\begin{aligned} \dot{\boldsymbol{x}}_1 &= \boldsymbol{x}_2 \\ \boldsymbol{M}(\boldsymbol{x}_1)\dot{\boldsymbol{x}}_2 &= -\boldsymbol{C}(\boldsymbol{x}_1, \boldsymbol{x}_2)\boldsymbol{x}_2 - \boldsymbol{g}(\boldsymbol{x}_1) + \boldsymbol{\tau}_{tot} \\ \boldsymbol{y} &= \boldsymbol{x}_1 \ . \end{aligned} \tag{18}$$

A reduced-order observer for $\boldsymbol{x}_2$ can be defined as

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{y})\dot{\boldsymbol{z}} = & -\boldsymbol{C}(\boldsymbol{y},\hat{\boldsymbol{x}}_2)\hat{\boldsymbol{x}}_2 - \boldsymbol{g}(\boldsymbol{y}) - \boldsymbol{M}(\boldsymbol{y})\frac{\partial \boldsymbol{k}(\boldsymbol{y})}{\partial \boldsymbol{y}}\hat{\boldsymbol{x}}_2 + \boldsymbol{\tau}_{tot} \\ \hat{\boldsymbol{x}}_2 = & \boldsymbol{z} + \boldsymbol{k}(\boldsymbol{y}) \end{aligned} \tag{19}$$

where $\boldsymbol{k} : \mathbb{R}^n \mapsto \mathbb{R}^n$ is a differentiable mapping to be specified. For simplicity, we take $\boldsymbol{k}(\boldsymbol{y}) = k_0\boldsymbol{y}$ with $k_0 \in \mathbb{R}^+$ so that $\frac{\partial \boldsymbol{k}(\boldsymbol{y})}{\partial \boldsymbol{y}} = k_0\boldsymbol{I}_n$.

Then, we assume there is an upper bound on the joint velocity norm as

$$||\dot{\boldsymbol{q}}|| \leq v_{max} \ . \tag{20}$$

Finally, defining the estimation error as $\boldsymbol{\epsilon} = \boldsymbol{x}_2 - \hat{\boldsymbol{x}}_2$ and using a Lyapunov argument with the following Lyapunov function

$$V(\boldsymbol{\epsilon}, t) = \frac{1}{2}\boldsymbol{\epsilon}^T\boldsymbol{M}(\boldsymbol{y}(\boldsymbol{t}))\boldsymbol{\epsilon} \ , \tag{21}$$

it can be proved that a high enough k_0 ensures asymptotic — and local exponential — stability of the error dynamics to the origin. In particular, it should be

$$k_0 = \max_{\boldsymbol{y}\in\mathbb{R}^n} \frac{c_0(\boldsymbol{y})(v_{max} + \eta)}{\lambda_{min}(\boldsymbol{M}(\boldsymbol{y}))} \tag{22}$$

where $\eta > 0$ is picked arbitrarily such that $\eta > ||\boldsymbol{\epsilon}||$.

In the following, we use a more conservative estimate of k_0 by maximizing and minimizing separately the numerator and denominator of 22, obtaining

$$k_0 = \frac{\bar{c}_0(v_{max} + \eta)}{2\lambda_1} \ . \tag{23}$$

Note that the observer (19) requires the same input as the robotic system, i.e., $\boldsymbol{\tau}_{tot} = \boldsymbol{\tau}_m + \boldsymbol{\tau}_{ext}$ [1]. The commanded torque $\boldsymbol{\tau}_m$ is usually available. However, the external collision torque $\boldsymbol{\tau}_{ext}$ is not known and requires the *identification phase* of the collision pipeline [5]. Since for our purposes there is no need for the knowledge of $\boldsymbol{\mathcal{F}}_{ext}$, we can use the momentum-based residual from (15) as a virtual sensor for the external joint torque, and use it in the observer dynamics in place of $\boldsymbol{\tau}_{ext}$.

5 Simulations

The experiments presented in this Section are performed following the scheme reported in Figure 1. The components of the block scheme not yet discussed are detailed in the following.

[1] Recall we are under the assumption of $\boldsymbol{\tau}_F = \boldsymbol{0}$, otherwise it should be considered here.

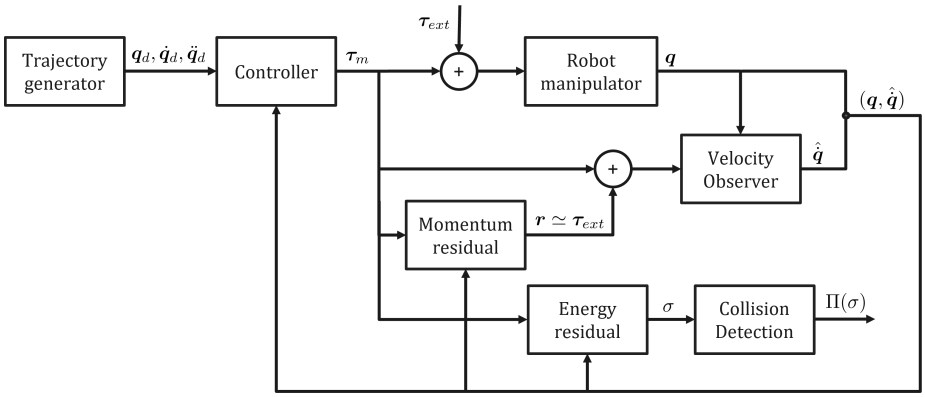

Figure 1: Overall block scheme.

5.1 Experimental Configuration

Robot manipulator The robot at hand is a spatial 3R elbow-type manipulator and each link is considered as a full cylinder with the Center of Mass along the link axis at half its length. The dynamic and kinematic parameters of its three links are reported in Table 1. The robot is endowed with joint encoders, hence the measure of $\boldsymbol{q}$ is available.

Link	Mass $[kg]$	Length $[m]$	Cylinder radius $[m]$
1	15	0.5	0.2
2	10	0.5	0.1
3	5	0.4	0.1

Table 1: Kinematic and dynamic parameters of the 3R robot manipulator.

Trajectory generator A desired motion for the end-effector is generated in terms of $\boldsymbol{p}_d^{ee}, \dot{\boldsymbol{p}}_d^{ee}, \ddot{\boldsymbol{p}}_d^{ee}$ where $\boldsymbol{p}^{ee} \in \mathbb{R}^3$ is the end-effector position, and then mapped to the joint space $\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d$. While the robot moves, different types of external forces $\boldsymbol{\mathcal{F}}_{ext} = \begin{pmatrix} \boldsymbol{f}_{ext}^T & \mathbf{0}^T \end{pmatrix}^T$ are applied at the end-effector, and no external moment is ever exerted. The profile of the desired quantities and of the external force is reported in Figure 2, and seven experimental phases can be identified.

A: Ideally, the robot is at rest at its desired initial resting state $(\boldsymbol{q}_d(0), \mathbf{0})$.

B: The end-effector moves on a circle in the (y, z) plane. During this motion, a force is applied along the x direction, and subsequently another force that lies in the (y, z) plane is applied.

C: The robot is at rest and no force is applied.

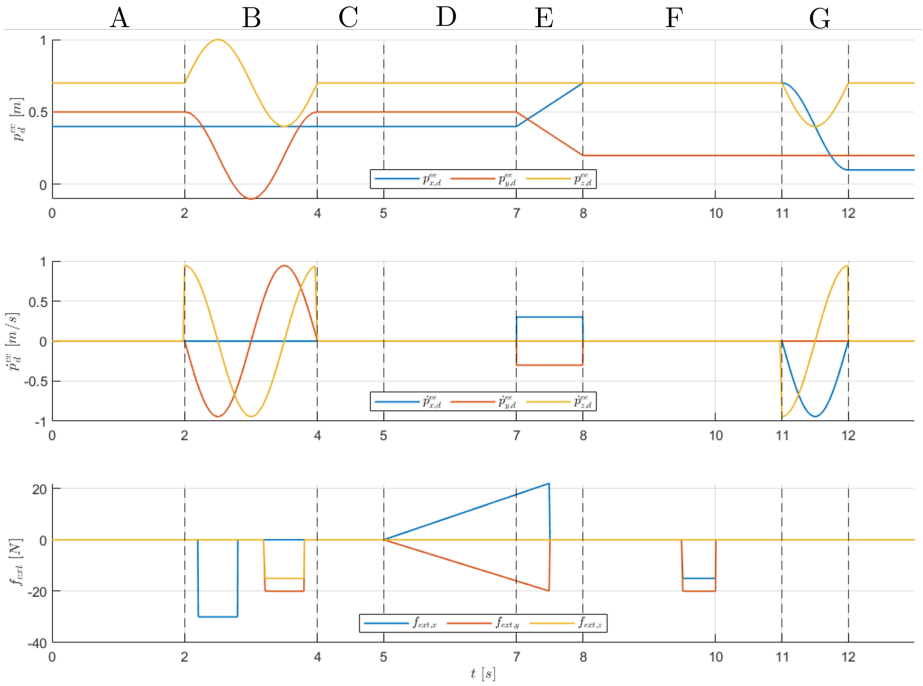

Figure 2: Profile of the desired end-effector trajectory position (top) and velocity (middle), and of the external force applied at the end effector (bottom). The experiment phases are explicitly reported on top.

D: With the robot still at rest, a force with a ramp profile — in both the x and y component — is applied.

E: The robot moves on a straight line in the (x, y) plane, and the external force ceases during this phase.

F: The robot is at rest and a force that lies in the (x, y) plane is applied.

G: The robot moves on a semi-circle in the (x, z) plane in absence of external forces. Radius and period of the circle are the same of phase B.

Controller Given $\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d$ from the trajectory generator and the partially estimated state of the robot $(\boldsymbol{q}, \hat{\dot{\boldsymbol{q}}})$, we employ a Feedback Linearization scheme with a feedforward and proportional-derivative controller as follows

$$\boldsymbol{\tau}_m = \hat{\boldsymbol{M}}(\boldsymbol{q})[\ddot{\boldsymbol{q}}_d + \boldsymbol{K}_p(\boldsymbol{q}_d - \boldsymbol{q}) + \boldsymbol{K}_d(\dot{\boldsymbol{q}}_d - \hat{\dot{\boldsymbol{q}}})] + \hat{\boldsymbol{C}}(\boldsymbol{q}, \hat{\dot{\boldsymbol{q}}})\hat{\dot{\boldsymbol{q}}} + \hat{\boldsymbol{g}}(\boldsymbol{q}) \tag{24}$$

where $\boldsymbol{K}_p, \boldsymbol{K}_d > 0$ are diagonal gain matrices. In ideal conditions, with $\hat{\boldsymbol{M}}(\boldsymbol{q}) = \boldsymbol{M}(\boldsymbol{q})$, $\hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$, $\hat{\boldsymbol{g}}(\boldsymbol{q}) = \boldsymbol{g}(\boldsymbol{q})$, $\hat{\dot{\boldsymbol{q}}} = \dot{\boldsymbol{q}}$, a linear and decoupled behaviour is achieved in the joint space.

Collision Detection To determine whether a collision has occurred, the value of the energy residual $\sigma(t)$ is analyzed. Ideally, since $\sigma \simeq P_{ext}$, it should be $\sigma(t) \neq 0$ only in case of collisions. However, in practice modeling errors and noise are always present, hence a thresholding mechanism is used. Namely, we can define the collision detection signal as

$$\Pi(\sigma(t)) = \begin{cases} 1 & \text{if } |\sigma(t)| \geq \rho \\ 0 & \text{if } |\sigma(t)| < \rho \end{cases} . \tag{25}$$

Note that the choice of this threshold is crucial, as a high value may result in false negatives and missed detections which is clearly a dangerous scenario; however, a small value may lead to false positives hindering the nominal robot operations. For the sake of comparison, the threshold ρ is kept constant throughout all the experiments, and in particular $\rho = 8\ [W]$ has proved to be a suitable choice.

5.2 Simulation Methodology and Results

For the implementation of the reduced-order observer we set

$$v_{max} = 4\ [rad/s], \quad \eta = 1\ [rad/s] \tag{26}$$

and initialize the velocity estimate to zero, i.e., $\hat{\boldsymbol{x}}_2(0) = \hat{\dot{\boldsymbol{q}}}(0) = \boldsymbol{0}$.

In all the following simulations, the robot will start with a non-zero end-effector position error, and an initial joint velocity $\dot{\boldsymbol{q}}(0) = (0.5, -0.5, 1)^T\ [rad/s]$. During phase A, the robot will try to recover both position and velocity errors.

We present a set of simulations to compare the collision detection performances using the energy residual and estimating $\hat{\dot{\boldsymbol{q}}}$ via numerical differentiation

and using the reduced-order observer presented in Section 4. In the first presented simulation we consider the ideal case where all the dynamic and kinematic parameters are exactly known, and the second experiment presents the case in which there are small model uncertainties. In both cases, we will use the controller (24) with $\boldsymbol{K}_p = \boldsymbol{K}_d = 70\boldsymbol{I}_3$.

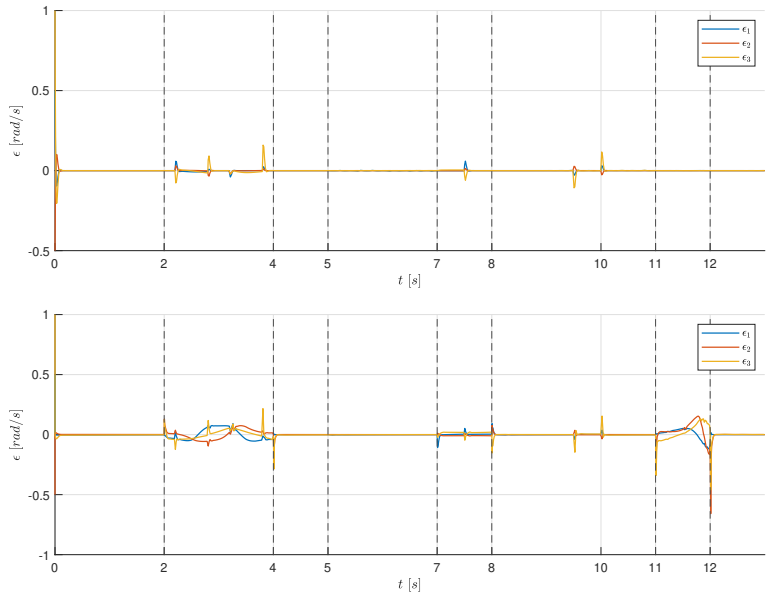

Figure 3: Estimation error $\boldsymbol{\epsilon} = \dot{\boldsymbol{q}} - \hat{\dot{\boldsymbol{q}}}$ using the reduced-order observer (top) and numerical differentiation (bottom) for joint velocity estimation, during the experiment in the ideal case.

Ideal case Firstly, from Figure 3 it is possible to appreciate the better velocity estimation of the observer with respect to numerical differentiation. In particular, in correspondence of abrupt changes of velocity (see Fig. 2), the former method keeps a correct estimation while the latter shows a higher error. The detection performance during the experimental phases A-F presented before are commented in the following for the two velocity estimation schemes, and the profile of relevant quantities are reported in Figure 4 and 5 respectively.

A: No external force is applied, the robot is moving to zero the initial position and velocity errors.

B: During the circular end-effector motion in the (y, z) plane the first push is in the x direction, hence — recalling L1 — it is not detected both in the observer and numerical differentiation case. Contrarily, the second external force, being within the motion plane, is correctly recognized by both detection schemes.

C: The robot should be ideally at rest and no force is applied. However, a spurious detection happens in case of finite differences velocity estimation in the initial instants of this phase, possibly due to the discontinuity in the desired velocity $\dot{\boldsymbol{p}}_d^{ee}$ (see Fig. 2) that causes a higher estimation error (see Fig. 3, bottom).

D: Despite the gradual increase of the force in the (x, y) plane, independently from the velocity estimation method, the residual remains close to 0 since the robot does not move. This result corroborates L2.

E: As soon as the robot tip starts moving on the same plane (x, y) of the exerted external force, a collision is detected in both detection schemes.

F: While the robot is at rest, forces in x and y directions are applied without being detected (similarly to what happens in phase D).

G: During the motion in the (x, z) plane no external force is applied. However, like at the beginning of phase C, multiple erroneous detections occur using the numerical differentiation method.

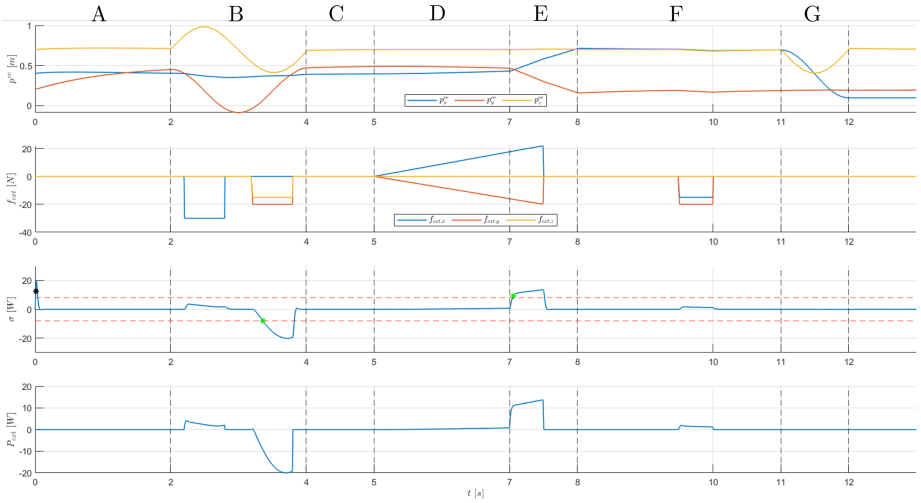

Figure 4: Behaviour of collision detection using the velocity observer described in Section 4 during the experiment in the ideal case. From the top: end-effector position $\boldsymbol{p}^{ee}$; external forces $\boldsymbol{f}_{ext}$; scalar energy residual σ; external power P_{ext}. In the second to last plot, correct detections are marked with a green star, while incorrect ones with a black star. The experiment phases are explicitly reported on top.

Note that the profile of the energy residual σ resembles P_{ext}, as expected, especially when using the reduced-order observer (see Fig. 4).

Model uncertainties To investigate the robustness to model uncertainties, the actual link masses are increased by 5% with respect to the nominal values.

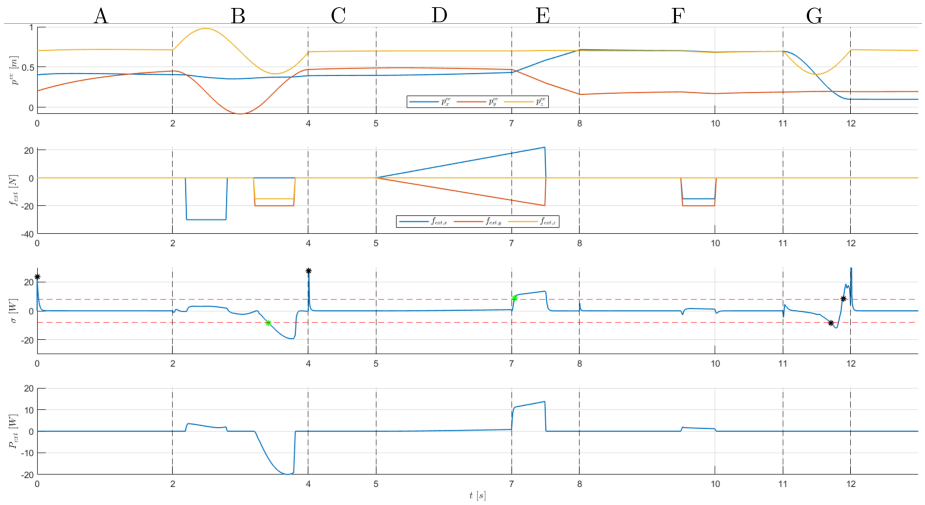

Figure 5: Behaviour of collision detection using numerical differentiation during the experiment in the ideal case. From the top: end-effector position $\boldsymbol{p}^{ee}$; external forces $\boldsymbol{f}_{ext}$; scalar energy residual σ; external power P_{ext}. In the second to last plot, correct detections are marked with a green star, while incorrect ones with a black star. The experiment phases are explicitly reported on top.

As shown in Figure 6, the detection pattern based on the reduced observer is the same of the nominal case (see Fig. 4), while using numerical differentiation an additional incorrect detection happens with respect to Figure 5.

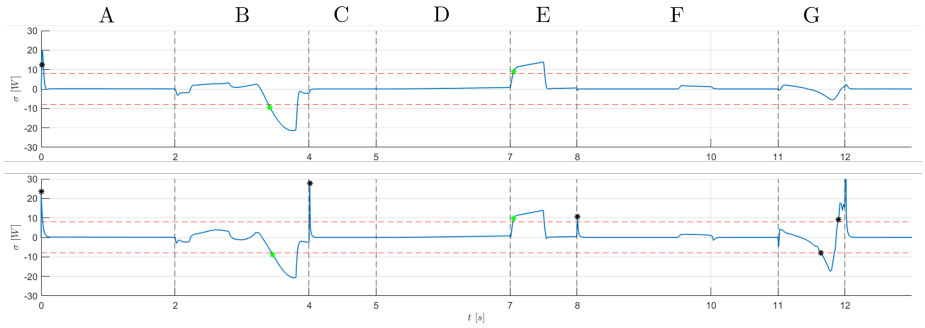

Figure 6: Profile of the energy residual σ in case of 5% link mass uncertainty, using the velocity observer (top) and numerical differentiation (bottom). Correct collision detections are marked with a green star, while incorrect ones with a black star. The experiment phases are explicitly reported on top.

Compliant behaviour The values of the controller gains $\boldsymbol{K}_p, \boldsymbol{K}_d$ used so far lead to a fairly accurate tracking of the desired trajectory. However, as previously stated in Section 3.1, a more compliant behaviour allows to improve the

collision detection capability of $\sigma(t)$ at the expense of a worse trajectory tracking. Such compliance is reached reducing the gains to $\boldsymbol{K}_p = 50\boldsymbol{I}_3$, $\boldsymbol{K}_d = 10\boldsymbol{I}_3$ and is clearly shown in Figure 7. The collision detection performance based on the energy residual are remarkably different with respect to the stiffer case, especially in the phases B and F. Although the external forces are applied either in directions orthogonal to the end-effector motion (first applied force in phase B) or with the robot at rest (phase F), since we are considering non-impulsive forces, the robot starts moving in the push directions leading to the detection of the collisions, as shown in Figure 8.

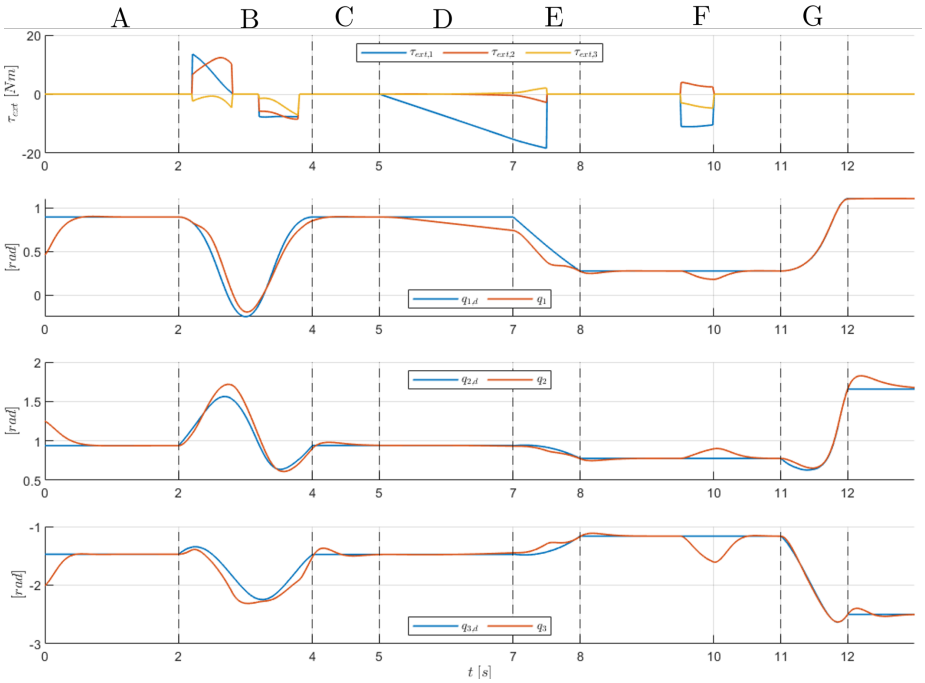

Figure 7: Tracking performance using a soft controller. From the top: profile of $\boldsymbol{\tau}_{ext}$ due to collisions; desired (blue) and actual (orange) joint position profile. The experiment phases are explicitly reported on top.

6 Conclusions

In this work, we have presented an energy-based and a momentum-based signals that can be used for collision detection. Actually, since the latter is a virtual sensor of the joint torque due to collisions, we have employed it for this purpose in the reduced-order velocity observer.

Simulations have shown that this approach to velocity estimation outperforms the classical numerical differentiation method, both in terms of estimation error and in terms of collision detection performance. Moreover, experimental

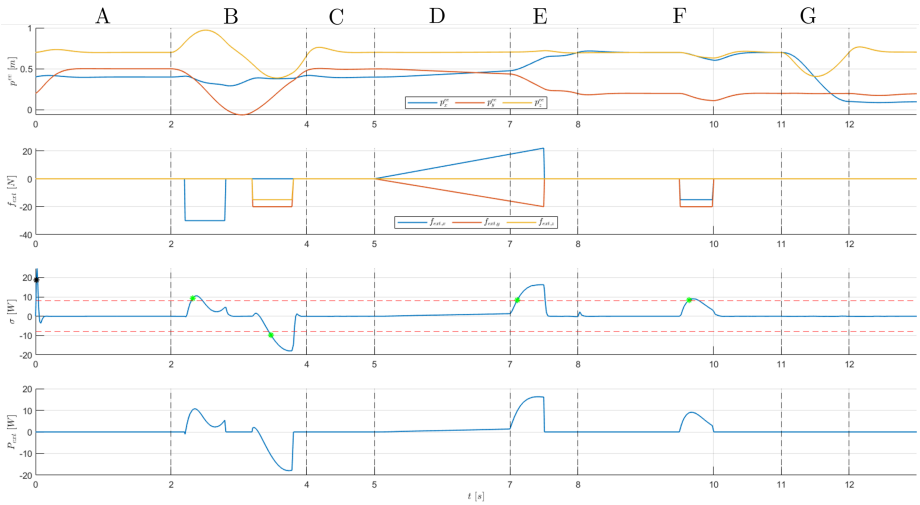

Figure 8: Behaviour of collision detection using the velocity observer described in Section 4 and a soft controller. From the top: end-effector position $\boldsymbol{p}^{ee}$; external forces $\boldsymbol{f}_{ext}$; scalar energy residual σ; external power P_{ext}. In the second to last plot, correct detections are marked with a green star, while incorrect ones with a black star. The experiment phases are explicitly reported on top.

results have also highlighted the limitations of the scalar energy residual — collisions while at rest and in the direction orthogonal to the contact point motion — and how they may be bypassed giving up some tracking performance. However, in practice these limitations are not critical in terms of safety: in the first case the robot is at rest and the human is colliding with it, so the robot does not represent a threat; in the second scenario, the robot body subject to collision is moving on a plane orthogonal to the direction of the push, hence the colliding body does not have a velocity component pointing towards the source of the external force.

Finally, it is important to note that in this work the momentum-based residual has been used solely as proxy of the external torque due to collisions: in practice, it can be used jointly with the energy residual to have a more robust collision detection scheme which is also free from the aforementioned limitations. Moreover, this vector residual can be readily used in the following phases of isolation and identification in the *collision event pipeline*.

References

[1] Andrea Cristofaro and Alessandro De Luca. Reduced-order observer design for robot manipulators. *IEEE Control Systems Letters*, 7:520–525, 2023.

[2] A. de Luca and R. Mattone. Sensorless robot collision detection and hybrid force/motion control. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 999–1004, 2005.

[3] Alessandro De Luca, Alin Albu-Schaffer, Sami Haddadin, and Gerd Hirzinger. Collision detection and safe reaction with the dlr-iii lightweight manipulator arm. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1623–1630, 2006.

[4] Alessandro De Luca and Lorenzo Ferrajoli. A modified newton-euler method for dynamic computations in robot fault detection and control. In *2009 IEEE International Conference on Robotics and Automation*, pages 3359–3364, 2009.

[5] Sami Haddadin, Alessandro De Luca, and Alin Albu-Schäffer. Robot collisions: A survey on detection, isolation, and identification. *IEEE Transactions on Robotics*, 33(6):1292–1312, 2017.