

Robot Collisions: A Survey on Detection, Isolation, and Identification

Sami Haddadin¹, *Member, IEEE*, Alessandro De Luca², *Fellow, IEEE*, and Alin Albu-Schäffer³, *Fellow, IEEE*

Abstract—Robot assistants and professional coworkers are becoming a commodity in domestic and industrial settings. In order to enable robots to share their workspace with humans and physically interact with them, fast and reliable handling of possible collisions on the entire robot structure is needed, along with control strategies for safe robot reaction. The primary motivation is the prevention or limitation of possible human injury due to physical contacts. In this survey paper, based on our early work on the subject, we review, extend, compare, and evaluate experimentally model-based algorithms for real-time collision detection, isolation, and identification that use only proprioceptive sensors. This covers the context-independent phases of the collision event pipeline for robots interacting with the environment, as in physical human–robot interaction or manipulation tasks. The problem is addressed for rigid robots first and then extended to the presence of joint/transmission flexibility. The basic physically motivated solution has already been applied to numerous robotic systems worldwide, ranging from manipulators and humanoids to flying robots, and even to commercial products.

Index Terms—Collision detection, collision identification, collision isolation, flexible joint manipulators, human-friendly robotics, physical human–robot interaction (pHRI), safe robotics.

I. INTRODUCTION

HUMAN-FRIENDLY robots will soon become flexible and versatile robotic coworkers helping humans in complex or physically demanding work in industrial settings. In addition, they will be an integral part of our daily life as multipurpose service assistants in homes. As a common characteristic in these foreseen applications, robots should be able to operate in very dynamic, unstructured, and partially unknown environments, sharing the workspace with a human user, preventing upcoming undesired collisions, handling unavoidable or intentional

physical contacts in a safe and robust way, and reactively generating sensor-based motions. Achieving this robot behavior is the global objective of physical human–robot interaction (pHRI) research [1], [2].

The pathway to the ambitious goal of close collaboration of humans and robots involves novel mechanical designs of manipulator links and actuation, aimed at reducing inertia/weight and based on compliant components, extensive uses of external sensors, so as to allow fast and reliable recognition of human–robot proximity, and development of human-aware motion planning and control strategies.

One of the core problems in pHRI is the handling of collisions between robots and humans, with the primary motivation of limiting possible human injury due to physical contacts.¹ Indeed, undesired collisions should be avoided by monitoring the workspace with external sensors so as to anticipate dangerous situations. However, since relative motions between robot and human may be very fast or hardly predictable, use of exteroceptive sensors may not be sufficient to prevent collisions. Moreover, when a direct and intentional human–robot interaction is desired, contacts are actually needed for task execution, requiring contact classification that distinguishes between intended and unintended contacts. In fact, one is interested in gathering the maximum amount of physical information from the impact event, such as which is the contact location and intensity, in order to let the robot react in the most appropriate fashion. For systematizing the contact handling problem, we introduce a unified framework entitled the *collision event pipeline*, which aims at embracing all relevant phases collisions may undergo.

Note that the schemes presented in this work may easily be extended to other classes of robots beyond manipulators. In fact, systematic collision handling is, e.g., highly beneficial in mobile robots, which are sought to be guided through appropriately detected and identified user forces [3], upper-bodies of anthropomorphic systems [4], or even flying robots [5]. More work is needed for humanoid robots [6], in view of the floating base, which requires an explicit treatment for the angular momentum of the system.

A. Collision Event Pipeline and State of the Art

We propose to generally consider up to seven elementary phases in the complete *collision event pipeline* (see Fig. 1).

¹As a matter of fact, the same problem is relevant also in robot manipulation, where impacts may occur with an uncertain environment and potentially fragile objects.

Manuscript received February 1, 2017; accepted June 12, 2017. Date of publication October 5, 2017; date of current version December 14, 2017. This paper was recommended for publication by Associate Editor H. Kress-Gazit and Editor A. Kheddar upon evaluation of the reviewers' comments. This work was partially supported by the European Commission within the FP7 project SAPHARI under Grant 287513, by the European Union Horizon 2020 research and innovation program under Grant 688857, and by the Alfred-Krupp Award for Young Professors. (*Corresponding author: Sami Haddadin.*)

S. Haddadin is with the Institute of Automatic Control and with the Hannover Center for Systems Neuroscience, Leibniz University of Hannover, 30167 Hannover, Germany (e-mail: sami.haddadin@irt.uni-hannover.de).

A. De Luca is with the Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, 00185 Roma, Italy (e-mail: deluca@diag.uniroma1.it).

A. Albu-Schäffer is with the Institute of Robotics and Mechatronics, DLR—German Aerospace Center, 82234 Weßling, Germany (e-mail: alin.albu-schaeffer@dlr.de).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TRO.2017.2723903

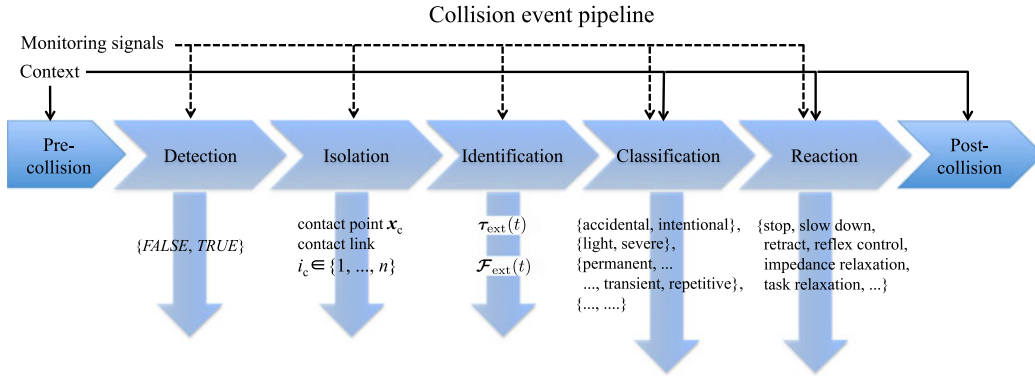


Fig. 1. Seven phases of the collision event pipeline and their expected outputs.

Various *monitoring signals* can be used to gather information about the event. Some phases are (almost) instantaneous, others are not. Furthermore, the phases from detection to identification are independent from *context*, whereas the remaining phases depend on internal and external factors, including the human/environment state and the on-going task.

1) *Pre-collision Phase*: The two primary goals here are collision avoidance and/or anticipatory robot motion to minimize impact effects. Planning a nominal collision-free path requires (at least, local) knowledge of the current environment geometry [7]. Offline motion planning techniques are computationally expensive, and their efficient conversion to online methods capable of handling instantaneous changes is still on-going research [8], [9], in particular when considering human-aware situations [10], [11]. Anticipating a collision is typically based on the use of additional external sensors, such as onboard vision [12], [13] or RGB-D cameras placed in the environment [14]. Many algorithms were designed for generating (incremental) collision-free paths in real time, e.g., using artificial potentials [15], [16], elastic strips [17], or other similar variants [18]–[20]. Other approaches like [11], [21]–[23] aim for effectively planning collision-free robot motions also incorporating the prediction of human behavior.

However, the inherent speed of human motion (humans are generally an order of magnitude faster than any typical high gear ratio robot) makes it basically impossible to *a priori* guarantee collision-free behavior.

2) *Collision Detection*: The collision detection phase, whose binary output denotes whether a robot collision occurred or not, is characterized by the transmission of contact wrenches, often for very short impact durations. The occurrence of a collision, which may happen anywhere along the robot structure, shall be detected as fast as possible. A major practical problem is the selection of a threshold on the monitoring signals, so as to avoid false positives and achieve high sensitivity at the same time. A rather intuitive approach is to monitor the measured currents in robot electrical drives, looking for fast transients possibly caused by a collision [24], [25]. Another proposed scheme compares the actual commanded motor torques (or motor currents) with the nominal model-based control law (i.e., the instantaneous motor torque expected in the absence of collision), with any difference being attributed to a collision [26].

This idea has been refined by considering the use of an adaptive compliance control [27], [28]. However, tuning of collision detection thresholds in these schemes is difficult because of the highly varying dynamic characteristics of the control torques. One way to obtain both collision detection and isolation is to use sensitive skins [29]–[32]. However, it is obviously more practical and reliable to detect and possibly isolate a collision without the need of additional tactile sensors.

3) *Collision Isolation*: Knowing which robot part (e.g., which link of a serial manipulator) is involved in the collision is an important information that can be exploited for robot reaction. Collision isolation aims at localizing the contact point x_c , or at least which link i_c out of the n -body robot collided. On the other hand, the previously mentioned monitoring signals used in [24]–[28] are in general not able to achieve reliable collision isolation (even when robot dynamics is perfectly known). In fact, they either rely on computations based only on the nominal desired trajectory or compute joint accelerations by inverting the mass matrix and thus spreading the dynamic effects of the collision on a single link to several joints. Alternatively, they use acceleration estimates for torque prediction and comparison, which inherently introduces noise (due to double numerical differentiation of position data) and intrinsic delays. The common drawback of these methods is that the effect of a collision on a link propagates to other link variables or joint commands due to robot dynamic couplings, affecting thus the isolation property.

4) *Collision Identification Phase*: Other relevant quantities about a collision are the directional information and the intensity of the generalized collision force, either in terms of the acting Cartesian wrench $\mathcal{F}_{ext}(t)$ at the contact or of the resulting external joint torque $\tau_{ext}(t)$ during the entire physical interaction event. This information characterizes (in some cases, completely) the collision event. The first method that achieved simultaneously collision detection, isolation, and identification was proposed in [33] and [34]. The basic idea was to view collisions as faulty behaviors of the robot actuating system, while the detector design took advantage of the decoupling property of the robot *generalized momentum* [35], [36].

5) *Collision Classification Phase*: Based on the information generated in the previous phases, we can interpret the collision nature in a context-dependent way, such as classifying the col-

lision as accidental or intentional [37], light or severe, or even labeling its time course as permanent, transient, or repetitive.

6) *Collision Reaction Phase*: Indeed, the robot should react purposefully in response to a collision event, i.e., taking into account available contextual information. Because of the fast dynamics and high uncertainty of the problem, the robot reaction should be embedded in the lowest control level. For instance, the simplest reaction to a collision is to stop the robot. However, this may possibly lead to inconvenient situations, where the robot is unnaturally constraining or blocking the human [38], [39]. To define better reaction strategies, information from collision isolation, identification, and classification phases should be used. Some examples of successful collision reaction strategies have been given in [40]–[42] and are shortly mentioned in Fig. 1.

7) *Postcollision Phase*: Once a safe condition has been reached after the reaction to a collision, the robot should autonomously decide how to proceed, e.g., whether to try to recover the original task, or to abandon it and how [42]. For instance, if the collision was classified as intentional, the robot may recognize that the human wish was to start a specific physical collaboration. This decision making is still a relatively open problem. Its resolution cannot be done purely at the control level, as global environmental information and reasoning is certainly needed. One possible approach using machine learning is given in [37].

B. Contribution

The focus of this paper is on collision detection, isolation, and identification, i.e., phases 2–4 in the collision event pipeline of Fig. 1. We start from our early works [34], [35], [40], [43], [44] and take advantage of the extensive experience gained over the years in developing, using, and refining our original methods. They were successfully used, e.g., for a hydraulically driven humanoid [45], for flying robots [5], or even in commercial products like the KUKA LWR iiwa [46], FRANKA EMIKA [47], or ABB YuMi [48]. The main characteristics of our methods are the following:

- 1) use only proprioceptive robot sensors (similarly to [49], however, only forces at the robot end-effector are estimated there);
- 2) handle collisions that may occur anywhere along the robotic structure, even repeatedly (i.e., at any place and any time);
- 3) have an elegant physical motivation, being based on monitoring quantities such as total energy or generalized momentum of the robot;
- 4) can be implemented efficiently either for fully rigid robots or for robots with flexible joints, taking advantage of special features in the latter case;
- 5) are independent from the method used to command/control the robot in any phase, alleviating thus the critical issue of defining collision detection thresholds;
- 6) provide directional and intensity information for a safe robot reaction after isolation of the collision.

Apart from summarizing the original methods, the main novel contributions of this paper are the following.

- 1) First, we propose the *collision event pipeline* as a unified framework for presenting and classifying existing and fu-

ture results in the robot collision handling literature (see Section I-A).

- 2) We derive a computationally more favorable version of the energy observer that is based on kinetic energy only (see Section III-A).
- 3) We elaborate on the isolation and identification properties of the momentum-based monitoring method, showing sufficient conditions for localizing the contact point on the colliding link and estimating the contact force vector using only proprioceptive information (see Section IV-B).
- 4) We compare and rate all considered methods in terms of computational effort, required measurement quantities, and further characteristics for rigid and flexible robots (see Sections V and VI).
- 5) We provide in Section VII new experimental results with the link momentum observer for the DLR/KUKA LWR, including its capability for external force estimation as well as collision isolation and identification. In addition, the first experimental analysis using the energy observer introduced in Section III-A is presented.
- 6) Finally, a simulative analysis of real-world effects on the momentum-based monitoring method is carried out (see Section VII).

In addition, some practically relevant results and comments concerning the implementation of the proposed methods are also provided.

- 1) We discuss the relevant implementation/computational aspects of our collision detection and isolation algorithms (see Section III-F).
- 2) We comment on the relevance of motor/link-side friction and of thresholding on the collision detection performance (see Section IV-A).
- 3) We also introduce a variant of the momentum observer for flexible robots that does not require any information on the joint stiffness and uses only motor- and link-side position measurements (see Section VI).

This paper is organized as follows. Section II outlines the robot dynamics and useful properties for rigid robots and flexible joint robots. Section III provides the theoretical background on monitoring signals that can be used in the collision event pipeline. The detection, isolation, and identification phases for the momentum observer are presented in Section IV. Sections V and VI compare all considered methods for rigid and flexible joint robots, respectively. Section VII reports simulations and experimental results for the momentum and energy observers. Finally, Section VIII concludes this paper.

II. ROBOT DYNAMICS AND PROPERTIES

In this section, we summarize the dynamic modeling and the relevant properties of robots, while incorporating collisions with the environment into the formulation.

A. Rigid Robots

We consider robot manipulators as open kinematic chains of rigid bodies, having n rigid joints. The generalized coordinates $q \in \mathbb{R}^n$ can be associated with the position of the links. The

motor positions $\theta \in \mathbb{R}^n$, as reflected through the gear ratios of rigid transmissions, are assumed to coincide with the link coordinates, i.e., $\theta = q$. The dynamic model is

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau_m - \tau_F \quad (1)$$

where $M(q) \in \mathbb{R}^{n \times n}$ is the symmetric and positive-definite inertia matrix, $C(q, \dot{q})\dot{q} \in \mathbb{R}^n$ is the centripetal and Coriolis vector, factorized with the matrix C of Christoffels' symbols, and $g(q) \in \mathbb{R}^n$ is the gravity vector. The nonconservative terms on the right-hand side of (1) are the active motor torque $\tau_m \in \mathbb{R}^n$ and the dissipative torque $\tau_F \in \mathbb{R}^n$, mostly due to motor friction (e.g., $\tau_F = f(\dot{q})$ with $f(\dot{q})\dot{q} \geq 0$). The torque applied to the robot by the (electrical) actuators is $\tau_m = K_i \dot{i}_m$, with motor current \dot{i}_m and diagonal current-to-torque gain matrix $K_i > 0$. In the following, we assume that motor currents can be measured, the gain matrix is known, and that τ_m corresponds to the commanded torque.

A basic property of the robot dynamics is the skew-symmetry of matrix $\dot{M}(q) - 2C(q, \dot{q})$, which is also equivalent to the identity (see, e.g., [50])

$$\dot{M}(q) = C(q, \dot{q}) + C^T(q, \dot{q}). \quad (2)$$

Next, consider the inclusion of generalized contact forces along the robot structure due to a collision. For simplicity, we assume that there is at most a single link involved in the collision. Let

$$V_c = \begin{bmatrix} v_c \\ \omega_c \end{bmatrix} = \begin{bmatrix} \dot{x}_c \\ \dot{\omega}_c \end{bmatrix} = \begin{bmatrix} J_{c, \text{lin}}(q) \\ J_{c, \text{ang}}(q) \end{bmatrix} \dot{q} = J_c(q)\dot{q} \in \mathbb{R}^6 \quad (3)$$

be the stacked (screw) vector of linear velocity v_c at the contact point x_c and angular velocity ω_c of the considered robot link, with an associated (geometric) contact Jacobian $J_c(q)$. Accordingly, the Cartesian collision wrench at x_c (with its force and moment) is denoted by

$$\mathcal{F}_{\text{ext}} = \begin{bmatrix} f_{\text{ext}} \\ m_{\text{ext}} \end{bmatrix} \in \mathbb{R}^6. \quad (4)$$

Note that the contact location x_c and the contact Jacobian are typically unknown. When a collision occurs, the robot dynamics (1) becomes

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + \tau_F = \tau_m + \tau_{\text{ext}} = \tau_{\text{tot}} \quad (5)$$

where $\tau_{\text{ext}} \in \mathbb{R}^n$ is the external joint torque given by

$$\tau_{\text{ext}} = J_c^T(q) \mathcal{F}_{\text{ext}}. \quad (6)$$

The two active (nonconservative) terms on the right-hand side of (5) have been collectively denoted as $\tau_{\text{tot}} \in \mathbb{R}^n$, while we moved the dissipative term τ_F to the left-hand side.

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

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

with $g(q) = (\partial U_g(q) / \partial q)^T$. From (5) and the skew-symmetry of $\dot{M} - 2C$, it follows that

$$\dot{E} = \dot{q}^T \tau_{\text{tot}} - \dot{q}^T \tau_F \quad (8)$$

which represents the power balance in the system, including the external power $P_{\text{ext}} = \dot{q}^T \tau_{\text{ext}}$.

The *generalized momentum* p of the robot is defined as

$$p = M(q)\dot{q}. \quad (9)$$

From (5), the time evolution of p can be written as

$$\dot{p} = \tau_{\text{tot}} - \tau_F + \dot{M}(q)\dot{q} - C(q, \dot{q})\dot{q} - g(q) \quad (10)$$

$$= \tau_{\text{tot}} - \tau_F + C^T(q, \dot{q})\dot{q} - g(q), \quad (11)$$

where (2) has been used. In particular, the i th component of \dot{p} can also be written as

$$\dot{p}_i = \tau_{\text{tot}, i} - \tau_{F, i} - \frac{1}{2} \dot{q}^T \frac{\partial M(q)}{\partial q_i} \dot{q} - g_i(q) \quad (12)$$

for $i = 1, \dots, n$. Thus, each component of the generalized momentum is affected only by the associated component of the nonconservative (active and dissipative) torques. This decoupling property will be further exploited later.

B. Robots with Flexible Joints

If flexibility of the transmission and reduction components has to be considered, the generalized coordinates need to be doubled, since there is a dynamic displacement between the motor position $\theta \in \mathbb{R}^n$ and the link position $q \in \mathbb{R}^n$. Moreover, an additional potential energy term U_e due to joint deflection $\delta = \theta - q$ appears. An associated joint torque τ_J couples link and motor dynamics, which dependence on δ may take various forms (from linear to nonlinear) and may also be time varying or independently modified by an additional control input, such as in VSA [51]. No matter how complex the functional expression of τ_J is, the developments in this paper will remain unchanged as long as the link dynamics is affected in practice only by τ_J directly. For the sake of clarity, we consider in the following the case of linear viscoelasticity at each robot joint only. For a complete overview of this class, refer to [52].

For a robot with n viscoelastic joints, we consider the so-called reduced model of Spong [53], which assumes no inertial couplings between the motor and link bodies. The *link dynamics* replacing (1) is

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + K_J(q - \theta) = D_J(\dot{\theta} - \dot{q}) - \tau_{F, q} \quad (13)$$

where $\tau_{F, q}$ denotes the friction terms acting on the link side of the joint. We define the *elastic torque* transmitted through the joints as

$$\tau_J = K_J(\theta - q). \quad (14)$$

The joint stiffness matrix $K_J = \text{diag}\{K_{J, i}\} \in \mathbb{R}^{n \times n}$ is diagonal and positive definite, while the joint damping matrix $D_J = \text{diag}\{D_{J, i}\} \in \mathbb{R}^{n \times n}$ is diagonal and positive semidefinite. All the dissipative terms have been collected on the right-hand side of (13). The value of τ_J in (14) is also the output of joint torque sensing devices, when available.

For the *motor dynamics*, we assume a decoupled second-order system

$$B\ddot{\theta} + K_J(\theta - q) = \tau_m - D_J(\dot{\theta} - \dot{q}) - \tau_{F, \theta} \quad (15)$$

where $\mathbf{B} = \text{diag}\{B_i\} \in \mathbb{R}^{n \times n}$ is the diagonal positive-definite motor inertia matrix. $\boldsymbol{\tau}_{F,\theta}$ contains friction terms acting on the motor side of the joint.

In many cases, the mechanical design of the transmission/reduction elements is such that one can neglect the joint damping, $\mathbf{D}_J \simeq \mathbf{0}$, as well as the influence of friction on the link side, $\boldsymbol{\tau}_{F,q} \simeq \mathbf{0}$. In particular, this is true for the DLR/KUKA Lightweight Robot series. On the other hand, the motor friction $\boldsymbol{\tau}_{F,\theta}$ is usually not negligible. However, note that for, e.g., intrinsically elastic systems like the DLR Hand/Arm System HASY, damping is explicitly modeled, since it improves the joint torque estimation based on motor- and link-side position and velocity sensing.

Therefore, when also including the presence of joint torques due to contact forces (acting on the link dynamics), we shall consider the following dynamic model of robots with flexible joints:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}_J + \boldsymbol{\tau}_{\text{ext}} = \boldsymbol{\tau}_{\text{tot},J} \quad (16)$$

$$\mathbf{B}\ddot{\boldsymbol{\theta}} + \boldsymbol{\tau}_J = \boldsymbol{\tau}_m - \boldsymbol{\tau}_{F,\theta}. \quad (17)$$

Equation (16) has basically the same properties as (5), except for $\boldsymbol{\tau}_J$ now being—together with $\boldsymbol{\tau}_{\text{ext}}$ —the driving torque of the dynamics. By analogy, we have collectively denoted the two terms on the right-hand side of (16) as $\boldsymbol{\tau}_{\text{tot},J} \in \mathbb{R}^n$.

For later use, and similarly to the rigid case, we derive next energy and momentum equations for robots with flexible joints. The *total energy* E_J of a flexible joint robot also includes the energy U_e stored in the elasticities and takes the form

$$\begin{aligned} E_J = T + U_g + U_e &= \frac{1}{2}\dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q})\dot{\mathbf{q}} + \frac{1}{2}\dot{\boldsymbol{\theta}}^T \mathbf{B}\dot{\boldsymbol{\theta}} \\ &+ U_g(\mathbf{q}) + \frac{1}{2}(\boldsymbol{\theta} - \mathbf{q})^T \mathbf{K}_J(\boldsymbol{\theta} - \mathbf{q}). \end{aligned} \quad (18)$$

From (16) and (17), it follows that the power \dot{E}_J exchanged between the environment and the system is given by

$$\dot{E}_J = \dot{\mathbf{q}}^T \boldsymbol{\tau}_{\text{ext}} + \dot{\boldsymbol{\theta}}^T (\boldsymbol{\tau}_m - \boldsymbol{\tau}_{F,\theta}). \quad (19)$$

The torque $\boldsymbol{\tau}_J$ is only acting within the system and not between system and environment. While (19) is perfectly valid, using the *link energy* only [namely, (7)] is more convenient for detection purposes even in the flexible joint case. This is because (7) does not contain the motor energy terms, which are of no relevance for collisions with the environment. Note that neither friction terms nor motor positions appear when evaluating the time derivative of E with (16)

$$\dot{E} = \dot{\mathbf{q}}^T \boldsymbol{\tau}_{\text{tot},J}. \quad (20)$$

Also note that the input torque of the system described by (16) is now given by $\boldsymbol{\tau}_J + \boldsymbol{\tau}_{\text{ext}} = \boldsymbol{\tau}_{\text{tot},J}$.

The *generalized momentum* vector \mathbf{p}_J of a flexible joint robot has $2n$ components defined as

$$\mathbf{p}_J = \begin{bmatrix} \mathbf{p}_q \\ \mathbf{p}_\theta \end{bmatrix} = \begin{bmatrix} \mathbf{M}(\mathbf{q})\dot{\mathbf{q}} \\ \mathbf{B}\dot{\boldsymbol{\theta}} \end{bmatrix}. \quad (21)$$

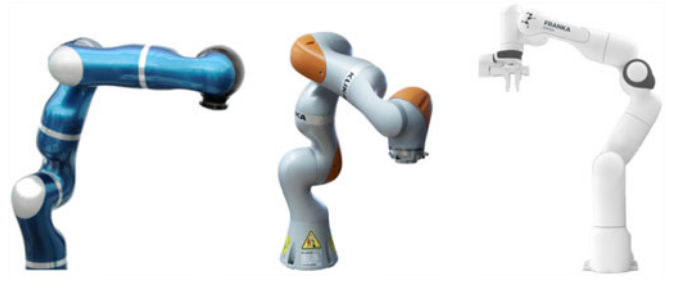


Fig. 2. From left: the DLR LWR-III (by courtesy of DLR), the KUKA LBR iiwa, and FRANKA EMIKA (by courtesy of FRANKA EMIKA GmbH).

Accordingly, the time evolution of the two n -dimensional momentum components is

$$\dot{\mathbf{p}}_q = \boldsymbol{\tau}_{\text{tot},J} + \dot{\mathbf{M}}(\mathbf{q})\dot{\mathbf{q}} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}) \quad (22)$$

$$= \boldsymbol{\tau}_{\text{tot},J} + \mathbf{C}^T(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}) \quad (23)$$

and

$$\dot{\mathbf{p}}_\theta = \boldsymbol{\tau}_m - \boldsymbol{\tau}_J - \boldsymbol{\tau}_{F,\theta}. \quad (24)$$

Note again that (23) is independent from friction terms.

C. Lightweight Robot Family

Since the DLR/KUKA family of lightweight robots is used as reference platform throughout this paper, we shortly summarize some key features of these manipulators [54], [55].

Fig. 2 shows the DLR LWR-III, the KUKA LBR iiwa (intelligent industrial work assistant), and the new fully soft-robotics-controlled lightweight robot FRANKA EMIKA [47] of FRANKA EMIKA GmbH. These lightweight robots have a slender arm design, with seven revolute joints (a spherical shoulder, an elbow joint, and a spherical wrist for the DLR LWR-III and the KUKA LBR iiwa), and are human-like in size. The LWR-III weighs 13.5 kg and is able to handle loads up to 15 kg, so an approximate unitary payload-to-weight ratio is achieved. In contrast to standard torque controlled robots such as the UR or ABB Yumi, this lightweight robot family is driven by DC brushless motors and can be fully joint torque or joint impedance controlled.

The drive trains contain harmonic drive gears with large reduction ratios (ranging between 100:1 and 160:1). Together with the presence of joint torque sensors, these gearboxes induce significant elasticity at the joints, but with intrinsic low joint damping and low friction on the link side. All manipulators used in our experiments were either of the DLR LWR-III or KUKA LWR4 type, which are almost equivalent from a dynamics point of view. Therefore, we refer to them collectively as DLR/KUKA LWR. All robots in the family are equipped with motor position encoders (measuring $\boldsymbol{\theta}$) and joint torque sensors (measuring $\boldsymbol{\tau}_J$), while the link position \mathbf{q} is estimated as $\hat{\mathbf{q}}$ from these measurements using (14). Ideally, the following relation holds:

$$\hat{\mathbf{q}} = \boldsymbol{\theta} - \hat{\mathbf{K}}_J^{-1} \boldsymbol{\tau}_J = \mathbf{q}. \quad (25)$$

From now on, an estimate of a generic scalar quantity x (or vector \mathbf{x}) will be denoted by \hat{x} (or $\hat{\mathbf{x}}$).

III. COLLISION MONITORING METHODS

Collision monitoring methods are dynamic or algebraic in nature and range from scalar monitoring of robot energy to momentum-based observers. In this section, emphasis is given on the properties of each of the reviewed methods and on their possible use for collision detection, isolation, and identification. In addition, we elaborate on computational issues, which were not considered in previous publications so far. All schemes are described here for rigid robots, while their extension to robots with flexible joints is considered in Section VI.

In this section, it is assumed that joint friction $\boldsymbol{\tau}_F = \mathbf{0}$, i.e., it is negligible or compensated by control. We come back to this issue in Section IV.

A. Estimation of the Power P_{ext} Associated with the External Joint Torque via Energy Observer

Since a collision is expected to change the energy level of the robotic system, an intuitive choice of a monitoring function is to resort to an energy argument. For this, define the scalar quantity

$$r(t) = k_O \left(\hat{E}(t) - \int_0^t (\dot{\mathbf{q}}^T \boldsymbol{\tau}_m + r) ds - \hat{E}(0) \right) \quad (26)$$

with initial value $r(0) = 0$, gain $k_O > 0$, and $\hat{E}(t)$ being the estimate of the total robot energy at time $t \geq 0$, as defined in (7) [34]. The monitoring signal r can be computed using the commanded motor torque $\boldsymbol{\tau}_m$, the measured link position \mathbf{q} , and the velocity $\dot{\mathbf{q}}$ (possibly obtained through numerical differentiation). No acceleration measurement or estimation is needed. Under ideal conditions, where $\hat{E}(t) = E(t)$, and using (8) together with the definition $\boldsymbol{\tau}_{\text{tot}} = \boldsymbol{\tau}_m + \boldsymbol{\tau}_{\text{ext}}$, the dynamics of r is

$$\dot{r} = k_O (\dot{\mathbf{q}}^T \boldsymbol{\tau}_{\text{ext}} - r) = k_O (P_{\text{ext}} - r). \quad (27)$$

This represents a first-order stable linear filter driven by the power P_{ext} associated with the external joint torque due to collision. During free motion, $r = 0$ holds. In response to a generic collision, r exponentially approaches the external joint torque with time constant $1/k_O$. When contact is lost, r decays similarly to zero.

For reducing the computational effort, we introduce a useful equivalent form of (26), which defines the monitoring signal based on the robot kinetic energy only, i.e.,

$$r(t) = k_O \left(\hat{T}(t) - \int_0^t (\dot{\mathbf{q}}^T (\boldsymbol{\tau}_m + \hat{\mathbf{g}}(\mathbf{q})) + r) ds - \hat{T}(0) \right) \quad (28)$$

where $\hat{\mathbf{g}}(\mathbf{q})$ is the estimate of the gravity vector $\mathbf{g}(\mathbf{q})$. Equation (28) is equivalent to (26), since the additional term

$$\int_0^t \dot{\mathbf{q}}^T \hat{\mathbf{g}}(\mathbf{q}) ds = \int_0^t \frac{\partial \hat{U}_g}{\partial \mathbf{q}} \dot{\mathbf{q}} ds = \int_0^t \dot{U}_g ds = \hat{U}_g(t) - \hat{U}_g(0) \quad (29)$$

provides exactly the missing potential energy terms. Therefore, the dynamics of the monitoring signal remains the one given

by (27). As a matter of fact, when moving from (26) to (28), we replaced the evaluation of the potential energy \hat{U}_g with that of the gravity vector $\hat{\mathbf{g}}$. The latter can be obtained using an efficient recursive numerical method (see also Section III-F), and in addition, it may have already been computed within the robot control law. Along the same line, note that (28) applies directly to cases when a gravity cancellation term is included in the motion control law. Then, the torque $\boldsymbol{\tau}_m$ in (28) will contain only the additional control action, whatever this is (e.g., a PD feedback law).

Not all possible collision situations can be detected by this simple scheme. When the robot is at rest, we have $\dot{\mathbf{q}} = \mathbf{0}$, and the instantaneous value of $\boldsymbol{\tau}_{\text{ext}}$ will not affect r until the robot will move (by the collision itself or otherwise). As a result, if the robot is initially not moving, a true impulsive collision cannot be reliably detected. Furthermore, when the robot is in motion, collisions are detected only if the Cartesian collision wrench \mathcal{F}_{ext} produces motion at the contact. In fact, from (3) and (6), the following property holds:

$$\dot{\mathbf{q}}^T \boldsymbol{\tau}_{\text{ext}} = \dot{\mathbf{q}}^T \mathbf{J}_c^T(\mathbf{q}) \mathcal{F}_{\text{ext}} = \mathbf{V}_c^T \mathcal{F}_{\text{ext}} = 0 \Leftrightarrow \mathbf{V}_c \perp \mathcal{F}_{\text{ext}} \quad (30)$$

i.e., wrenches orthogonal to the contact velocity cannot be detected. Despite the above drawbacks, it should be noted that the signal $r(t)$ approximately defines the power transferred during the collision to the robot, as long as contact is maintained. This information can also be useful for later decisions in the collision pipeline.

B. Direct Estimation of $\boldsymbol{\tau}_{\text{ext}}$

Using (5), the algebraic estimation of the external joint torque $\boldsymbol{\tau}_{\text{ext}}$ is

$$\hat{\boldsymbol{\tau}}_{\text{ext}} = \hat{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}} + \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \hat{\mathbf{g}}(\mathbf{q}) - \boldsymbol{\tau}_m. \quad (31)$$

This is indeed the most direct monitoring method, which uses the motor torque and the link position, velocity, and acceleration. Unfortunately, this approach is not applicable in practice, since typically only \mathbf{q} is measured and its double differentiation (to be performed on line) leads to the inclusion of nonnegligible noise in the estimation process. One might think of introducing acceleration sensors at the joints or on the links [56], but this increases system complexity significantly.

C. Monitoring $\boldsymbol{\tau}_{\text{ext}}$ via Inverse Dynamics

This method can be used when the robot is controlled by a high-performance (well-tuned) position controller that executes sufficiently smooth desired trajectories $\mathbf{q}_d(t) \in \mathbb{R}^n$ [40]. Then, it can be assumed that

$$\mathbf{q} \approx \mathbf{q}_d, \quad \dot{\mathbf{q}} \approx \dot{\mathbf{q}}_d, \quad \ddot{\mathbf{q}} \approx \ddot{\mathbf{q}}_d. \quad (32)$$

An estimate of the external joint torque due to collision is obtained by computing first the inverse dynamics (the so-called *feedforward* term) associated with (32), using (5) evaluated for $\boldsymbol{\tau}_{\text{ext}} = \mathbf{0}$:

$$\hat{\boldsymbol{\tau}}_{m, \text{ff}} = \hat{\mathbf{M}}(\mathbf{q}_d)\ddot{\mathbf{q}}_d + \hat{\mathbf{C}}(\mathbf{q}_d, \dot{\mathbf{q}}_d)\dot{\mathbf{q}}_d + \hat{\mathbf{g}}(\mathbf{q}_d). \quad (33)$$

The hat symbol on a robot dynamic term denotes its estimated version. Then, (33) is compared with the applied motor torque yielding

$$\hat{\tau}_{\text{ext}} = \hat{\tau}_{m, \text{ff}} - \tau_m. \quad (34)$$

This method may work reasonably well for stiff position control and smooth desired motion. However, it is not independent from the executed trajectories and from the specific control law and its parameters. Moreover, when using a computed torque controller, the above model-based computations need to be performed twice, once on the desired state trajectory and another on the actual state (q, \dot{q}) . Furthermore, from the instant of collision on, (32) is no longer valid. As a result, the method is suitable for monitoring but not for estimating τ_{ext} .

D. Estimation of τ_{ext} via Joint Velocity Observer

The underlying idea of this method is to use a reduced observer for the dynamic estimation of the joint velocity \dot{q} , and then to fit this scheme as a disturbance observer of the unknown external joint torque τ_{ext} [57]. The observer uses a reduced state of dimension n (rather than $2n$, the state dimension of a mechanical system with n generalized coordinates), and thus, it reacts faster to changes of the external joint torque (in fact, as a first-order system).

First, the actual acceleration \ddot{q} is expressed as

$$\ddot{q} = M^{-1}(q) (\tau_m - n(q, \dot{q}) + \tau_{\text{ext}}) \quad (35)$$

with the notation $n(q, \dot{q}) := C(q, \dot{q})\dot{q} + g(q)$. The observer output r represents an estimate $\hat{\tau}_{\text{ext}}$ of the true disturbance joint torque τ_{ext} . According to [58], we assume a constant disturbance and the following elementary model for the observation of the external collision joint torque:

$$\hat{\tau}_{\text{ext}} = r \in \mathbb{R}^n, \quad \dot{r} = 0. \quad (36)$$

Such a model is used since no further information on the expected behavior of τ_{ext} is available. The observer dynamics is then defined as

$$\begin{aligned} \dot{\hat{q}} &= \hat{M}^{-1}(q) (\tau_m - \hat{n}(q, \dot{q}) + r) \\ \dot{r} &= K_O (\hat{q} - \dot{q}) \end{aligned} \quad (37)$$

where $K_O = \text{diag}\{k_{O,i}\} > 0$ is the diagonal gain matrix of the observer. Fig. 3 shows the block diagram of the robot and the observer. The method provides as observer output

$$r(t) = K_O \left(\hat{q}(t) - \int_0^t \hat{M}^{-1}(q) (\tau_m - \hat{n}(q, \dot{q}) + r) ds - \hat{q}(0) \right). \quad (38)$$

The observer dynamics are given by

$$\dot{r}(t) = K_O \left(\hat{q}(t) - \hat{M}^{-1}(q) (\tau_m - \hat{n}(q, \dot{q}) + r) \right) \quad (39)$$

$$= K_O M^{-1}(q) (\tau_m - n(q, \dot{q}) + \tau_{\text{ext}}) \quad (40)$$

$$- K_O \hat{M}^{-1}(q) (\tau_m - \hat{n}(q, \dot{q}) + r).$$

Therefore, under ideal conditions, the dynamic relation between the external joint torque τ_{ext} and the observed disturbance

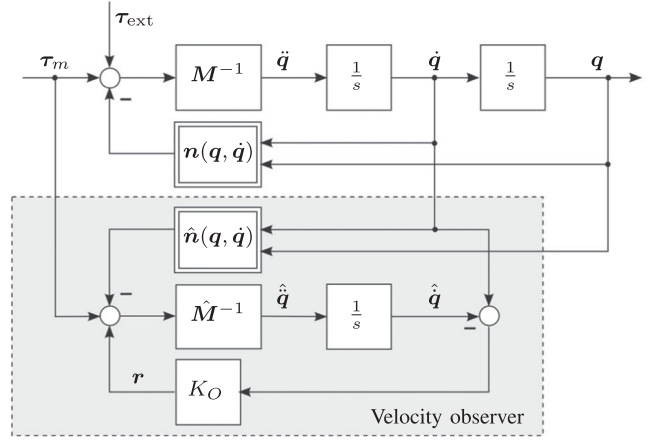


Fig. 3. Block diagram of the reduced observer (37) for \dot{q} .

r is obtained by replacing $\hat{M} = M$ and $\hat{n} = n$ in (40). It follows that

$$\dot{r} = K_O M^{-1}(q) (\tau_{\text{ext}} - r) \quad (41)$$

i.e., r is a filtered estimation of τ_{ext} but, due to the presence of the inverse inertia matrix, the filter equation is nonlinear and coupled.

E. Estimation of τ_{ext} via Momentum Observer

The monitoring method based on the generalized momentum observer introduced in [33], [34], and [44] was motivated by the desire of avoiding the inversion of the robot inertia matrix, decoupling the estimation result, and also eliminating the need of an estimate of joint accelerations. For compactness, define the quantity

$$\beta(q, \dot{q}) := g(q) + C(q, \dot{q})\dot{q} - \dot{M}(q)\dot{q} \quad (42)$$

$$= g(q) - C^T(q, \dot{q})\dot{q} \quad (43)$$

where property (2) has been used.

Based on the expression of the dynamics of p , as given by (10) or, equivalently, (11), the momentum observer dynamics is defined as

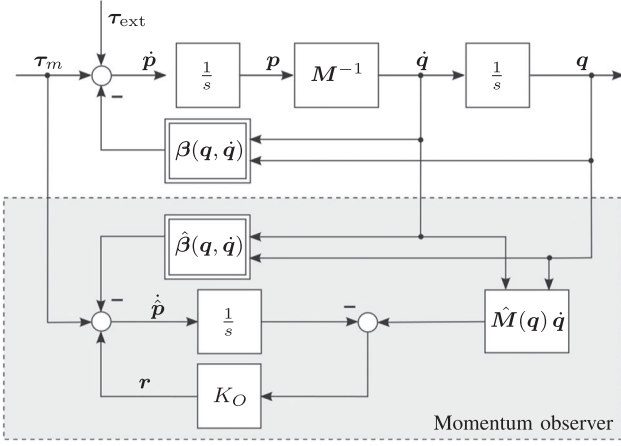
$$\begin{aligned} \dot{p} &= \tau_m - \hat{\beta}(q, \dot{q}) + r \\ \dot{r} &= K_O (\dot{p} - \hat{p}) \end{aligned} \quad (44)$$

where $K_O = \text{diag}\{k_{O,i}\} > 0$ is the diagonal gain matrix of the observer. The observer output $r(t)$ follows from integrating the second equation in (44) and then using the first one as

$$r = K_O \left(p(t) - \int_0^t \dot{p}(s) ds - p(0) \right) \quad (45)$$

$$= K_O \left(p(t) - \int_0^t (\tau_m - \hat{\beta}(q, \dot{q}) + r) ds - p(0) \right) \quad (46)$$

with $p = \hat{M}(q)\dot{q}$. Fig. 4 illustrates the block diagram of the momentum observer. The monitoring signal r is also called *residual* vector (see [34] and [35]). Under ideal conditions, $\hat{M} = M$ and $\hat{\beta} = \beta$, the dynamic relation between the external joint torque

Fig. 4. Block diagram of the reduced observer (44) for \dot{p} .

τ_{ext} and r is

$$\dot{r} = K_O (\tau_{\text{ext}} - r). \quad (47)$$

In other words, the filter equation is a stable, linear, decoupled, first-order estimation of the external collision joint torque τ_{ext} . Also, the monitoring vector r is sensitive to collisions even when $\dot{q} = 0$.

In the Laplace domain, working componentwise, it is

$$r_i = \frac{k_{O,i}}{s + k_{O,i}} \tau_{\text{ext},i} = \frac{1}{1 + T_{O,i}s} \tau_{\text{ext},i}, \quad i = 1, \dots, n. \quad (48)$$

Large values of $k_{O,i}$ give small time constants $T_{O,i} = 1/k_{O,i}$ in the transient response of that component of r , which is associated with the same component of the external joint torque τ_{ext} . In the limit, we obtain

$$K_O \rightarrow \infty \Rightarrow r \approx \tau_{\text{ext}}. \quad (49)$$

This nice feature makes the momentum observer a kind of virtual sensor for external joint torques acting along the robot structure. Note finally that the velocity observer as well as the momentum observer can also be used to compute an estimate \hat{q} of the joint acceleration \ddot{q} .

F. Computational Issues

We shortly comment on computational issues for the momentum observer introducing an approach for generally rating the computational effort of other monitoring methods as well (see Table I). Let $\text{NE}_{g_0}(q, \dot{q}, \ddot{q})$ denote your preferred implementation of the recursive Newton–Euler (NE) algorithm, which takes three vectors as inputs and the base acceleration g_0 as vector parameter (for a fixed-base manipulator, this is the constant gravity vector expressed in the base frame). For instance, the gravity vector in (28) can be evaluated as $g(q) = \text{NE}_{g_0}(q, 0, 0)$. By $\text{MNE}_{g_0}(q, \dot{q}, \ddot{q})$, we denote the modified NE algorithm proposed in [59], which takes four vectors as inputs to provide more flexibility. Let e_i be the i th vector of the canonical base of \mathbb{R}^n . Then, with the single call $\text{MNE}_0(q, \dot{q}, e_i, 0)$, the algorithm outputs the i th column of the Christoffel’s matrix $C(q, \dot{q})$. Every

run of the NE or MNE algorithm has computational complexity $O(n)$, being n the number of robot joints.

The momentum observer requires the computation of vectors p and β , for which the following alternatives are available.

- 1) The standard NE algorithm and numerical approximation of the directional derivative of the inertia matrix at control rate, with sampling time T_s

$$\begin{aligned} \beta &= g(q) + C(q, \dot{q})\dot{q} - \dot{M}(q)\dot{q} \\ &= \text{NE}_{g_0}(q, \dot{q}, 0) - \frac{M_k - M_{k-1}}{T_s} \dot{q} \\ p &= M(q)\dot{q} = \text{NE}_0(q, 0, \dot{q}). \end{aligned}$$

- 2) Use of customized Lagrange dynamics in symbolic form.
- 3) Mixed use of the standard and modified NE algorithms

$$\begin{aligned} \beta &= g(q) - C^T(q, \dot{q})\dot{q} \\ &= \text{NE}_{g_0}(q, \dot{q}, 0) \\ &\quad - [\text{MNE}_0(q, \dot{q}, e_1, 0) \dots \text{MNE}_0(q, \dot{q}, e_n, 0)]^T \dot{q} \\ p &= M(q)\dot{q} = \text{NE}_0(q, 0, \dot{q}). \end{aligned}$$

All three methods have been implemented and can run efficiently in real time. The choice strongly depends on the robotic system, in particular on the size of n , which has different influence on the various methods.

G. Summary

The advantages and drawbacks of the presented monitoring methods together with the required measurements and the computational effort are summarized in Table I. The use of a certain method depends on the number of joints, the reliability of the available measurements, and the available computational power.

IV. DETECTION, ISOLATION, AND IDENTIFICATION

In this section, we review and expand methods for generating a collision monitoring function and derive a new concept for the reconstruction of the contact wrench. Depending on the monitoring method of choice, a generalized collision monitoring signal $\mu(t)$ (possibly a scalar $\mu(t)$) is produced, which encapsulates exploitable information for the detection, isolation, and identification phases of the collision event pipeline. Due to its basic properties (i.e., linear error dynamics, no need of $M^{-1}(q)$ or \ddot{q} , no dependence on the direction of the external contact force), the momentum observer, which generates useful information for all three phases, is treated as the reference collision monitoring algorithm in this section, with $\mu := r$ according to (46). In comparison, the other algorithms show lower performance in more than one of the phases. Note that the following analysis for the momentum-based monitoring can also be carried out for the other algorithms.

A. Detection

Solving the collision detection problem means to decide whether a physical collision is present or not, limiting as much as possible the occurrence of false positives or false negatives.

TABLE I
COLLISION MONITORING METHODS FOR RIGID ROBOTS

Monitoring method		Computational effort	Measurement quantities	Advantages	Disadvantages
Absolute motor torque	$ \tau_m \geq \tau_{m,\max}$	complexity : * memory : * dual use : no overall : *	$\tau_m = K_i \dot{i}_m$	– extremely simple – needs only \dot{i}_m	– heuristic – friction dependent – low sensitivity – no isolation/identification
Motor torque change	$ \dot{\tau}_m \geq \dot{\tau}_{m,\max}$	complexity : * memory : * dual use : no overall : *	$\tau_m = K_i \dot{i}_m$ $\dot{\tau}_m = K_i \ddot{i}_m$	– extremely simple – needs only \dot{i}_m	– heuristic – friction dependent – low sensitivity – \dot{i}_m required – no isolation/identification
Motor torque tube	$\hat{\tau}_{\text{ext}} \approx \tau_{m,\text{ref}} - \tau_m$	complexity : * memory : * * * * dual use : no overall : *	$\tau_m = K_i \dot{i}_m$	– simple – model free – friction independent	– controller/trajectory dependent – only suitable for accurate tracking – needs prior execution – large data storage
Energy observer	Eq. (26); property: Eq. (27)	$\frac{1}{2} \dot{q}^T \text{NE}_0(q, 0, \dot{q})$ complexity : ** memory : ** dual use : partly overall : **	$\tau_m = K_i \dot{i}_m$ q, \dot{q}	– simple – only $M(q)\dot{q}$, no inversion – some identification	– no detection if – $\dot{q} \equiv 0$ – $\mathcal{F}_{\text{ext}} \perp V_c$ – no isolation
Direct estimation	Eq. (31)	$\text{NE}_{g_0}(q, \dot{q}, \ddot{q})$ complexity : * * * * * memory : * dual use : yes overall : * * *	$\tau_m = K_i \dot{i}_m$ q, \dot{q}, \ddot{q}	– ideal model – moderately complex – only $M(q)$, no inversion – isolation/identification possible	– \ddot{q} required – full dynamics necessary – friction dependent
Inverse dynamics	$\hat{\tau}_{\text{ext}} \approx \hat{\tau}_{m,\text{ff}} - \tau_m$; $\hat{\tau}_{m,\text{ff}}$ from Eq. (33)	$\text{NE}_{g_0}(q_d, \dot{q}_d, \ddot{q}_d)$ complexity : * * * * * memory : ** dual use : no overall : * * * * *	$\tau_m = K_i \dot{i}_m$	– no position sensor dependency – isolation/identification possible – feedforward computed anyway	– controller/trajectory dependent – calculate dynamics twice
Velocity observer	Eq. (38); property: Eq. (41)	$\text{NE}_{g_0}(q, \dot{q}, 0)$ $n \times \text{NE}_0(q, \dot{q}, e_i)$ \hat{M}^{-1} complexity : * * * * * memory : * dual use : partly overall : *	$\tau_m = K_i \dot{i}_m$ q, \dot{q}	– \ddot{q} not needed – isolation/identification possible – further use for state recovery	– sensitivity depends on q – full dynamics required
Momentum observer	Eq. (46); property: Eq. (47)	$\text{NE}_{g_0}(q, \dot{q}, 0)$ $n \times \text{NE}_0(q, 0, \dot{q}_i e_i / T_s)$ $\text{NE}_0(q, 0, \dot{q})$ OR customized dynamics OR $\text{NE}_{g_0}(q, \dot{q}, 0)$ $n \times \text{MNE}_0(q, \dot{q}, e_i, 0)$ $\text{NE}_0(q, 0, \dot{q})$ complexity : * * * * * memory : * dual use : partly overall : * * * *	$\tau_m = K_i \dot{i}_m$ q, \dot{q}	– \ddot{q} not needed – isolation possible – only $M(q)$, no inversion – sensitivity independent of q	– full dynamics necessary – $\hat{M}(q)$ required – (or n calls of modified NE for $C^T(q, \dot{q})$)

The following approach is an extension of our previous works [34], [40] by systematically deriving a threshold with higher robustness against model uncertainties and disturbances. A collision detection function $cd(\cdot)$ maps the monitoring signal $\mu(t)$ into the two disjoint collision classes *TRUE* or *FALSE*:

$$cd : \mu(t) \rightarrow \{\text{TRUE}, \text{FALSE}\}. \quad (50)$$

Ideally, this classification is obtained by

$$cd(\mu(t)) = \begin{cases} \text{TRUE}, & \text{if } \mu(t) \neq 0 \\ \text{FALSE}, & \text{if } \mu(t) = 0. \end{cases} \quad (51)$$

In practice, the accurate detection of a collision requires appropriate thresholding for robustness. The following nonidealities should be taken into account:

- 1) torque/current measurement noise \mathbf{n}_{τ_m} causing a motor torque error $\Delta\tau_m(\mathbf{n}_{\tau_m}) = \mathbf{K}_i \Delta\mathbf{i}_m(\mathbf{n}_{\tau_m})$, with the diagonal gain matrix $\mathbf{K}_i > \mathbf{0}$ containing the current-to-torque motor constants;
- 2) position and velocity sensor noise $\mathbf{n}_q, \mathbf{n}_{\dot{q}}$ causing joint position and velocity measurement errors $\Delta\mathbf{q}(\mathbf{n}_q), \Delta\dot{\mathbf{q}}(\mathbf{n}_{\dot{q}})$;
- 3) modeling errors in the estimated robot dynamics $\hat{\mathbf{D}} := \{\hat{\mathbf{M}}(\mathbf{q}), \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}), \hat{\mathbf{g}}(\mathbf{q})\}$ with respect to the true dynamics \mathbf{D} ;
- 4) friction torque $\tau_F(\mathbf{q}, \dot{\mathbf{q}}, \tau_{\text{tot}}, \vartheta, t)$, which in general depend on the robot state, load torque, temperature, and time.

Due to the aforementioned errors in measurement, modeling, and disturbances (friction, hysteresis, etc.), the monitoring signal becomes in practice $\boldsymbol{\mu}(t) \neq \mathbf{0}$ even when no contact is present. Assuming that all the above effects act at the torque level, introduce the unknown but bounded generalized disturbance vector

$$\boldsymbol{\tau}_{d,\text{tot}} = \Delta\tau_m(\mathbf{n}_{\tau_m}) + \tau_F(\mathbf{q}, \dot{\mathbf{q}}, \tau_{\text{tot}}, \vartheta, t) + \Delta\tau_d(\mathbf{q}, \dot{\mathbf{q}}, \hat{\mathbf{D}}) \quad (52)$$

where $\Delta\tau_d(\mathbf{q}, \dot{\mathbf{q}}, \hat{\mathbf{D}})$ denotes the torque disturbance that originates from position and velocity noise, plus dynamic modeling errors. Due to the presence of disturbance, the dynamics of the monitoring signal in the contact-free case becomes [see also (47)]

$$\dot{\boldsymbol{\mu}} = \mathbf{K}_O(\boldsymbol{\tau}_{d,\text{tot}} - \boldsymbol{\mu}). \quad (53)$$

Thanks to the stability of system (53), a bounded input $\boldsymbol{\tau}_{d,\text{tot}}(t)$ will always generate a bounded output $\boldsymbol{\mu}(t)$. Thus, the observation over a suitable set of robot motions for a sufficiently long time interval $[0, T]$ without any external disturbances leads to a convenient definition of $\boldsymbol{\mu}_{\text{max}} := \max\{|\boldsymbol{\mu}(t)|, t \in [0, T]\}$, where the maximum is to be taken componentwise. Taking into account a small constant robustness margin $\epsilon_{\text{safe}} > \mathbf{0}$, the collision detection problem can be decided using the conservative symmetric threshold $\epsilon_{\boldsymbol{\mu}} = \boldsymbol{\mu}_{\text{max}} + \epsilon_{\text{safe}} > \mathbf{0}$:

$$cd(\boldsymbol{\mu}(t)) = \begin{cases} \text{TRUE}, & \text{if } |\boldsymbol{\mu}(t)| > \epsilon_{\boldsymbol{\mu}} \\ \text{FALSE}, & \text{if } |\boldsymbol{\mu}(t)| \leq \epsilon_{\boldsymbol{\mu}} \end{cases} \quad (54)$$

This thresholding scheme robustly limits false positives. At the same time, it may produce false negatives for light contacts and possibly delayed positives for softer contacts. In order to overcome this limitation, suitable techniques such as friction compensation, model-based adaptive thresholding [60], or learning techniques can be applied to increase detection sensitivity [61].

The largely nonlinear error dynamics (52) is hard to be fully captured over the entire state and parameter spaces. Nonetheless, rather simple actions may improve significantly the collision detection performance.² First, by choosing a smaller (i.e., slower) filter constant \mathbf{K}_O in (46), the high-frequency measurement noise may be reduced at the price of some detection delay. Second, it is possible to exploit the frequency information of

the collision monitoring signal. Compared to typical collision events, the robot dynamics associated with nominal motion velocities has a frequency content in a lower range. Therefore, a simple yet effective possibility to cope with dynamic modeling errors is to high-pass filter componentwise the signals in (48), which leads to

$$\mu'_i := T_{O,i} s \mu_i = \frac{s}{s + k_{O,i}} \tau_{\text{ext},i} \quad (55)$$

with $T_{O,i} = 1/k_{O,i}$. Using μ'_i instead of μ_i for computing cd following (54), a more sensitive monitoring scheme for high-frequency torque components, i.e., to fast and stiff impacts, is obtained. Indeed, using (55) implies to ignore the very low frequency content of external joint torques. However, these may still be estimated in parallel by (48), possibly with smaller gains \mathbf{K}_O . In fact, one might even use an entire filter bank that consists of appropriately designed low-pass, high-pass, and bandpass filters for precise impact frequency analysis.

Alternatively to the high-pass filtered signal (55), one can also use as monitoring signal the difference between (34) and (48). Since dynamic modeling errors are equally present in both signals, their difference under ideal conditions would be almost independent from such errors. Due to the algebraic nature of (34), such a scheme would again lead to a componentwise high-pass filtered version of $\boldsymbol{\tau}_{\text{ext}}$

$$\mu''_i := \hat{\tau}_{\text{ext},i} - \mu_i \approx \tau_{\text{ext},i} - \frac{1}{1 + T_{O,i} s} \tau_{\text{ext},i} = \frac{s}{s + k_{O,i}} \tau_{\text{ext},i}. \quad (56)$$

However, note that this relation holds only under assumption (32), which is in general no longer true at the onset of a collision.

B. Isolation

Solving the isolation problem means to find the collision link i_c along the robot and the associated contact point \mathbf{x}_c . A systematic and formal solution to this problem did not exist so far and will be derived in the following assuming that only one collision occurs at a time. More expensive approaches for multiple contacts are given in [62, 4].

For a simpler notation, the threshold $\epsilon_{\boldsymbol{\mu}}$ is assumed to be zero. Suppose that a single contact occurs on the i_c th link of the open kinematic chain structure of the robot. Then, we have the property

$$\boldsymbol{\mu} = [\mu_1 \quad \dots \quad \mu_{i_c} \quad 0 \quad 0 \quad \dots \quad 0]^T \quad (57)$$

where the last $n - i_c$ zeros are structural. In fact, for a collision on link i_c , the last $n - i_c$ columns of the contact Jacobian $\mathbf{J}_c(\mathbf{q})$ are identically zero. Thus, the vector $\boldsymbol{\tau}_{\text{ext}} = \mathbf{J}_c^T(\mathbf{q}) \mathcal{F}_{\text{ext}}$, where $\mathbf{J}_c(\mathbf{q})$ is generally unknown, will have the last $n - i_c$ components identically zero. Thanks to the decoupled structure of (47), this is also the case for the last $n - i_c$ components of $\boldsymbol{\mu}$. On the other hand, the first i_c components of vector $\boldsymbol{\mu}$ will be generally different from zero, at least for the contact duration, and will start decaying exponentially toward zero as soon as contact is lost. Obviously, the collision link $i_c \in \{1 \dots n\}$ can be found as

$$i_c = \max\{i \in \{1 \dots n\} : \mu_i \neq 0\}. \quad (58)$$

²For the time being, assume that τ_F has negligible effects, which usually means that it is either compensated for very well or that joint torque sensing after the gearbox is used.

In general, μ will be affected only by a Cartesian collision wrench \mathcal{F}_{ext} that performs work on admissible robot motion, i.e., those net wrenches that do not belong to the kernel of $\mathbf{J}_c^T(\mathbf{q})$. More, in general, the sensitivity to \mathcal{F}_{ext} of each of the affected residual components (those up to index i_c) will vary with the arm configuration. Thanks to the properties of the generalized momentum, this dynamic analysis can be carried out based only on static considerations, using the transformation matrix $\mathbf{J}_c^T(\mathbf{q})$ from Cartesian wrenches to joint torques. In fact, the residual dynamics in (47) is unaffected by the robot velocity or acceleration.

Isolating the contact area on the colliding link i_c needs a more careful treatment, for which suitable methods are derived in the following. However, an isolated contact area can, in general, only be obtained for certain types of collisions. For a general collision wrench \mathcal{F}_{ext} acting on link $i = i_c$, we can write the rigid body transformation with respect to a frame attached in a known position to link i (e.g., the Denavit–Hartenberg one) and by using the adjoint matrix as

$$\mathcal{F}_i = \begin{bmatrix} \mathbf{f}_i \\ \mathbf{m}_i \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{O} \\ \mathbf{S}^T(\mathbf{r}_{c,i}) & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{f}_{\text{ext}} \\ \mathbf{m}_{\text{ext}} \end{bmatrix} = \mathbf{J}_{c,i}^T \mathcal{F}_{\text{ext}}. \quad (59)$$

The skew-symmetric matrix $\mathbf{S}(\cdot)$ for the vector product is built with the components of its vector argument, and $\mathbf{r}_{i,c} = -\mathbf{r}_{c,i} \in \mathbb{R}^3$ is the vector from the origin of frame i to the point of application of the contact force. For the position of the contact point, we have also that $\mathbf{x}_c = \mathbf{r}_i + \mathbf{r}_{i,c}$, being $\mathbf{r}_i = \mathbf{r}_i(\mathbf{q})$ the known position of the origin of frame i . Note that matrix $\mathbf{J}_{c,i}$ is constant when all quantities are expressed in the local frame i (assuming that the contact point remains the same). Moreover, the yet unknown contact Jacobian can be factorized as

$$\mathbf{J}_c(\mathbf{q}) = \mathbf{J}_{c,i} \mathbf{J}_i(\mathbf{q}) \quad (60)$$

where the known $6 \times n$ (geometric) Jacobian matrix $\mathbf{J}_i(\mathbf{q})$ is associated with the linear and angular velocity of frame i .

Once $i = i_c$ has been determined from (58), by using (60), an estimate of the effect of the Cartesian collision wrench at frame i can be obtained as

$$\hat{\mathcal{F}}_i = \begin{bmatrix} \hat{\mathbf{f}}_i \\ \hat{\mathbf{m}}_i \end{bmatrix} = (\mathbf{J}_i^T(\mathbf{q}))^\# \mu. \quad (61)$$

When $\mathbf{J}_i(\mathbf{q})$ has full row rank, which requires necessarily that $i_c \geq 6$, there is no loss of information in the obtained 6-D estimate $\hat{\mathcal{F}}_i$ when using (61). At this stage, note that it is not possible to determine \mathbf{x}_c (or, equivalently, $\mathbf{r}_{i,c}$) for a general contact wrench consisting of forces and moments. To proceed, we make the assumption that only wrenches of the form

$$\mathcal{F}_{\text{ext}} = \begin{bmatrix} \mathbf{f}_{\text{ext}} \\ \mathbf{0} \end{bmatrix} \quad (62)$$

with negligible contact moments occur, which is the typical case for impulsive collisions as well as the most relevant contact situation in practice. Assuming only contact wrenches of the form (62) allows the localization of the contact point where the (estimated) external force $\hat{\mathbf{f}}_i = \hat{\mathbf{f}}_{\text{ext}} \approx \mathbf{f}_{\text{ext}}$ is applied, thanks to (59) and (61). Simple manipulations provide the following

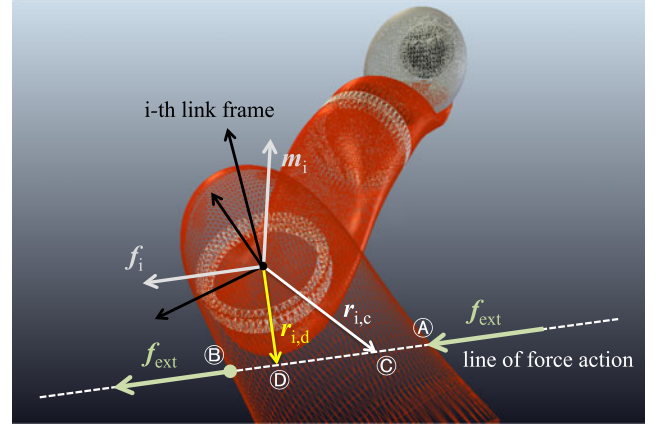


Fig. 5. Geometry for isolation of the contact point, when a pure contact force \mathbf{f}_{ext} is applied to link i at a generic point of its surface. Any generic location C along the line of force action is a candidate for being the application point of the external force. Point D is defined by the pseudoinverse solution (64). The additional knowledge of the link surface restricts the possibilities only to point A (pushing the link) or B (pulling the link).

linear system to be solved for the unknown vector $\mathbf{r}_{i,c}$:

$$\mathbf{S}^T(\hat{\mathbf{f}}_i) \mathbf{r}_{i,c} = \hat{\mathbf{m}}_i \quad (63)$$

where the 3×3 skew-symmetric coefficient matrix $\mathbf{S}^T(\hat{\mathbf{f}}_i)$ is always of rank 2 unless $\hat{\mathbf{f}}_i \equiv \mathbf{0}$. With reference to Fig. 5, (63) can be used to determine the *line of action* of \mathbf{f}_{ext} . The minimum distance vector from the origin of frame i to this line is given by the pseudoinverse solution to (63)

$$\mathbf{r}_{i,d} = (\mathbf{S}^T(\hat{\mathbf{f}}_i))^\# \hat{\mathbf{m}}_i. \quad (64)$$

All points along the line of force action can then be described by $\mathbf{r}_{i,d} + \lambda(\hat{\mathbf{f}}_i / \|\hat{\mathbf{f}}_i\|)$, for a varying scalar λ . Considering the links as convex bodies and assuming complete knowledge of the surface \mathcal{S}_{i_c} of the colliding link, the contact point \mathbf{x}_c can be found by intersecting the line of force action with \mathcal{S}_{i_c} . In Fig. 5, two final contact points can be isolated in principle (for two different values of λ , say λ_A and λ_B). Therefore, $\mathbf{r}_{c,i}$ can be determined in this way and so $\mathbf{J}_{c,i}$ in (60).

C. Identification

Solving the identification problem aims at estimating the collision joint torques τ_{ext} , as well as the contact wrenches \mathcal{F}_{ext} acting on the robot structure.

We have already shown in Section III-E that the momentum observer outputs an accurate estimation of τ_{ext} , i.e., full identification of the joint torques due to collision is always possible. In contrast, estimating the contact wrench \mathcal{F}_{ext} can only be done once $\mathbf{J}_c(\mathbf{q})$ is known. In order to calculate the corresponding external wrench we use (6), yielding

$$\hat{\mathcal{F}}_{\text{ext}} = (\mathbf{J}_c^T(\mathbf{q}))^\# \mu. \quad (65)$$

Three main sources could provide the required information about the contact Jacobian \mathbf{J}_c :

- 1) For manipulation tasks, where no other contact is expected, the task context specifies the contact location by

its tool center point (TCP) frame S_{TCP} . Thus, $\mathbf{J}_c = \mathbf{J}_{\text{TCP}}$ for a general contact wrench $\mathcal{F}_{\text{ext}} = [\mathbf{f}_{\text{ext}}^T \mathbf{m}_{\text{ext}}^T]^T$. For example, the TCP frame S_{TCP} could be associated with the tip of a tool that is mounted on the end-effector of a robot with $n \geq 6$ joints.

- 2) Assuming that (62) holds, one can obtain \mathbf{J}_c from (60) and the procedure outlined in Section IV-B.
- 3) The use of (possibly binary) tactile sensors allows us to localize directly the contact area. Then, \mathcal{F}_{ext} can be exactly computed if the Jacobian is full rank.

Of course, (65) leads to a complete solution only for a full row rank $\mathbf{J}_c(\mathbf{q})$, in which case $(\mathbf{J}_c^T)^\# = (\mathbf{J}_c \mathbf{J}_c^T)^{-1} \mathbf{J}_c$. Since the contact Jacobian may be interpreted as a sensitivity matrix, wrenches acting along singular directions cannot be sensed, as these are balanced by the reaction forces and moments internal to the manipulator structure. However, within the regular workspace, (65) may have interesting applications. For instance, it can be used in place of a wrist force-torque sensor for measuring TCP wrenches in the *virtual sensor* frame S_{vs}

$$\hat{\mathcal{F}}_{\text{vs}} = (\mathbf{J}_{\text{vs}}^T(\mathbf{q}))^\# \boldsymbol{\mu}. \quad (66)$$

For this, $\mathbf{J}_{\text{vs}} = \mathbf{J}_n(\mathbf{q})$ for a suitable definition of Denavit–Hartenberg parameters. This is useful for all tasks defined in the operational space that require wrist wrench control, e.g., in assembly processes. Such a virtual sensing strategy is especially interesting for industrial tasks, where it is desirable to limit the addition of external sensing devices. This reduces initial and maintenance costs and eliminates the risk of potential sensor failures.

On the other hand, estimating the full external wrench when the contact occurs at a *generic* location along the robot structure is in many cases impossible, even though the location \mathbf{x}_c is known. Causes for this are 1) the structural lack of rank of the contact Jacobian for collisions occurring at links proximal to the base, or 2) closeness to a singular configuration. Practically, this would lead to a transfer of some of the contact wrench components into the mechanical structure. Thus, parts of \mathcal{F}_{ext} cannot be identified anymore, no matter which joint torque based method (measurement or estimate) is used. Similarly, the estimation of multiple contact forces cannot be resolved with proprioceptive sensing only. In this context, its fusion with tactile sensors [63] and/or with exteroceptive sensing [64] are interesting recent research directions, together with the independent use of more expensive machine learning techniques [62].

V. COMPARISON OF METHODS FOR RIGID ROBOTS

This section provides a new systematic comparison of the essential five model-based collision monitoring algorithms from Section III in terms of their detection, isolation, and identification properties for the case of rigid robots.

For completeness, we consider also three known industrial schemes, namely monitoring the *absolute motor torque*, the *motor torque change*, or the *motor torque tube*. The first two methods based on absolute motor torque and on motor torque change simply check the motor torques and, respectively, their deriva-

tives against threshold values. These schemes are very simple to implement and have minimal computational cost. However, they have low detection sensitivity and are purely heuristic. Obviously, no isolation and identification are possible.

The main characteristics of all methods are depicted in Table I. For other aspects of the five model-based collision monitoring algorithms, refer to the respective descriptions in that section. For each monitoring method, we classify the *computational effort* in terms of *complexity*, *memory*, a possible *dual use* of the results (e.g., when the computed quantity is also used for the controller as well as for the *direct estimation* scheme), and a resulting *overall* rating (each from * = “very low” to ***** = “very high”). Moreover, we list the required *measurement quantities* and summarize the most significant *advantages* and *disadvantages*.

The complexity rating of the first two industrial schemes that are only based on a comparison of the motor torque to a threshold is * (“very low”). The third industrial scheme (*motor torque tube*) coarsely estimates the external joint torque, which requires only a little more computational effort but much more memory. The other analytical schemes are rated based on the effort required for the standard recursive NE algorithm. The widely used Big-O notation for computational effort is not really informative here, because all algorithms are $O(n)$, i.e., their complexity grows linearly with the number n of joints, except for the momentum observer, which is $O(n^2)$ when $\dot{\mathbf{M}}$ is computed exactly, but still $O(n)$ when an approximate derivation is used for this matrix. Note that the dynamics calculated for the *inverse dynamics* method cannot, in general, be used for control, leading to a higher overall rating in comparison to the *direct estimation* scheme.

The motor torque tube method is essentially a model-free equivalent of the inverse dynamics approach. By recording, possibly repeatedly, the motor torques along a given trajectory, a time-dependent reference profile $\tau_{m,\text{ref}}(t)$ is created, together with a robustness margin. These serve as dynamic thresholds next time the trajectory will be executed. This very simple, yet effective strategy has two main disadvantages: a large memory demand and little flexibility in terms of trajectory and control law modifications. In particular, the method does not apply to online generated trajectories.

Note that the monitoring performance of all schemes, except for the torque tube, is inherently bounded by the presence of unknown friction τ_F in (1). In turn, depending on the quality of a possibly implemented friction correction $\hat{\tau}_F$, as already mentioned in Section IV, monitoring performance can be improved equally over all the affected methods. We also remark that it is possible to run in parallel multiple algorithms for redundant monitoring, including also different variants of the same algorithm.

When combining the evaluations from Section III and Table I, we can conclude the following. The energy observer, being quite simple on the one hand, is not able to detect collisions in specific directions or when the robot is at rest on the other hand (see Section VII and Fig. 13). Moreover, it does not provide the possibility of isolation. The two schemes relying on the inverse dynamics of the robot require the measurement of

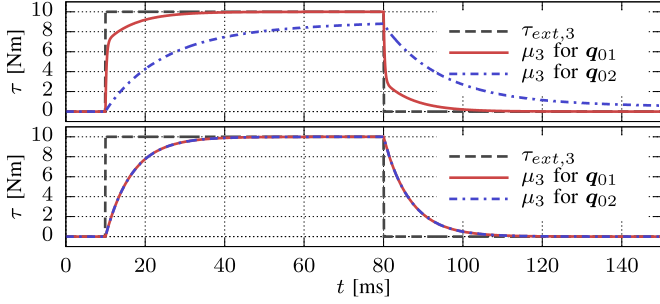


Fig. 6. External joint torque $\tau_{\text{ext},3}$ and its estimation μ_3 for the velocity observer (top, $k_{O,i} = 15 \text{ s}^{-1}$) and the momentum observer (bottom, $k_{O,i} = 150 \text{ s}^{-1}$) at two different initial configurations: $\mathbf{q}_{01} = (0, 0, 0, 0, 0, 0, 0)$ rad and $\mathbf{q}_{02} = (0, 1.3, 0, 1.3, 0, 1, 0)$ rad.

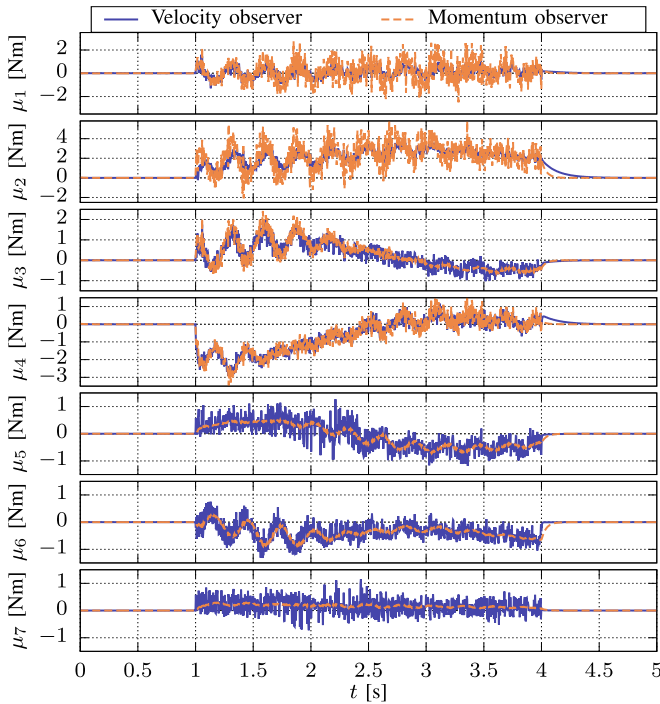


Fig. 7. External joint torque estimation in all seven joints during sinusoidal motion of the LWR for $\tau_{\text{ext}} = 0$ under ideal conditions for $t < 1 \text{ s}$ and $t > 4 \text{ s}$ and under perturbed conditions for $1 \text{ s} < t < 4 \text{ s}$.

the joint accelerations (when using real joint positions) or depend on the used controller (when using desired joint positions). Instead, the remaining two monitoring schemes based on velocity and momentum observation emerge for the following major advantages:

- 1) simultaneous collision detection, isolation, and identification;
- 2) complete independence from the reference trajectory and control law in execution.

Consequently, due to the nonlinear observer error dynamics of the velocity observer, which depends on $M(\mathbf{q})$, the *momentum observer* turns out to be the best choice for being used in the complete collision event pipeline (see Section VII and Figs. 6, 8, 12, and 15–19).

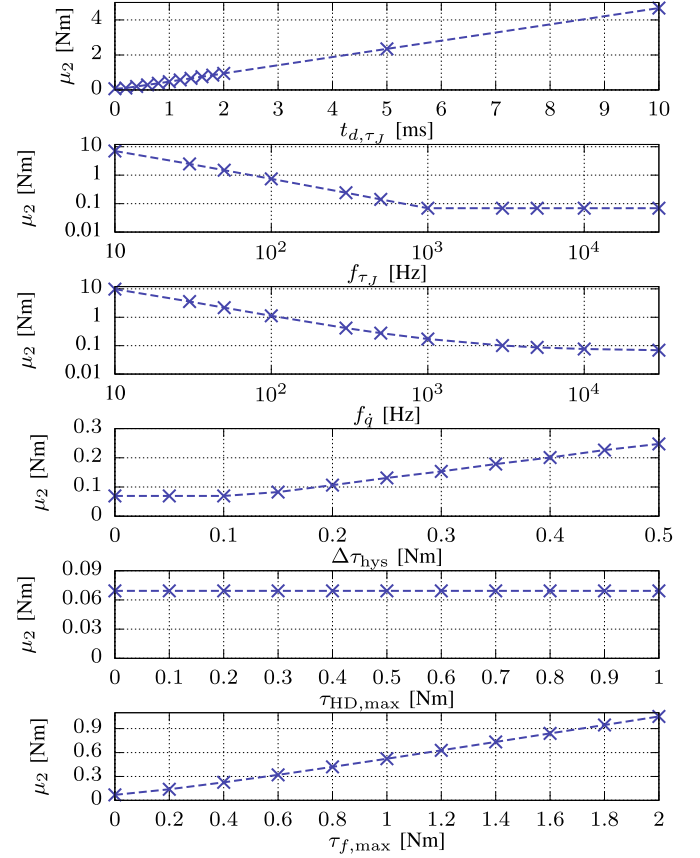


Fig. 8. Influence of different disturbing effects on μ_2 .

VI. EXTENSION TO ROBOTS WITH FLEXIBLE JOINTS

Extending the collision monitoring schemes to robots with flexible joints is rather straightforward. In general, we discriminate between two different types of flexible joint robot implementations, which differ significantly in terms of their respective sensorization.

- 1) *Elastic joint* robots have finite but typically rather high joint stiffness. This is often considered as a parasitic effect, like when strain wave gears are used as reduction elements. The joint deflection $\delta = \theta - \mathbf{q}$ is very small, and thus, τ_J cannot be estimated accurately from position measurements via (14). Therefore, these robots are equipped with motor encoders measuring θ , and with stiff joint torque sensors measuring τ_J after the reduction gears, and thus independently from $\tau_{F,\theta}$. Then, \mathbf{q} is estimated from (25). An alternative is the iterative scheme from [65], which computes the link position \mathbf{q} from θ based on a steady-state approximation. Alternatively, a second position sensor on the link side of the elastic joint measures \mathbf{q} directly. However, achieving the necessary resolution for obtaining $\dot{\mathbf{q}}$ via numerical differentiation is quite difficult in this case.
- 2) *Intrinsically flexible joint* robots are intentionally equipped with a very compliant element (typically, a mechanical spring) between motor and link. Thus, δ becomes significant and can be measured with sufficient accuracy. In these robots, two position sensors (e.g., encoders)

measure θ and \mathbf{q} , respectively. For the linear elastic case, i.e., \mathbf{K}_J constant, the joint torque is then estimated as $\hat{\tau}_J = \mathbf{K}_J(\theta - \mathbf{q})$. This case is commonly referred to as Series Elastic Actuation (SEA) [66]. However, the same arguments also hold for nonlinear joint torque/deflection characteristics and variable stiffness joints, where the joint torque estimate takes forms such as $\hat{\tau}_J = \hat{\mathbf{k}}_J(\theta, \mathbf{q}, \mathbf{u}_m)$. There, \mathbf{u}_m is an additional control input that serves the modification of the elastic behavior of the joint via a second actuator.

Despite the significant differences in the mechanical design and in the sensing principle used to obtain full state measurements, the extension of the collision monitoring schemes from rigid robots to the above classes of flexible joint robots is rather straightforward, thanks also to the physically motivated approaches that we have taken.

As a matter of fact, all methods from Section III can be directly applied to flexible joint robots with joint torque sensing (or estimation), by just substituting τ_m with τ_J (or $\hat{\tau}_J$) and using everywhere the link position \mathbf{q} (or $\hat{\mathbf{q}}$)—and *not* the motor position θ . An essential advantage of measuring τ_J directly (or accurately estimating it) is to decouple the collision monitoring problem from the friction $\tau_{F,\theta}$ acting on the motor side. Moreover, since the link dynamics (16) and the motor dynamics (17) are coupled only via τ_J , it is sufficient to consider the link dynamics alone for monitoring collisions. Stated differently, τ_J can be regarded as the system input in the link dynamics (16).

As an example, the *momentum observer* for rigid robots becomes the *link-side momentum observer* of the flexible joint case, taking the form

$$\mathbf{r} = \mathbf{K}_O \left(\hat{\mathbf{p}}_q(t) - \int_0^t (\tau_J - \hat{\beta}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{r}) dt - \hat{\mathbf{p}}_q(0) \right) \quad (67)$$

$$\dot{\mathbf{r}} = \mathbf{K}_O(\tau_{\text{ext}} - \mathbf{r}) \quad (68)$$

with $\hat{\mathbf{p}}_q = \hat{\mathbf{M}}(\mathbf{q})\dot{\mathbf{q}}$. Note that (67) has exactly the same form as (46), except for substituting τ_m with τ_J , whereas (68) is identical to (47).

Table II compares all collision monitoring schemes for flexible joint robots. In particular, the column with *measurement quantities* shows the user's choices for measuring and/or estimating the required signals for robots with elastic or intrinsically flexible joints (\vee stands for logical OR). Note that the possibility of estimating the link position, velocity, and acceleration via the iterative scheme from [65] is omitted for brevity.

The only method that does not require joint torque sensing (or estimation) is the *total momentum observer*. This is an extension of the monitoring method based on the link momentum: instead of observing \mathbf{p}_q only, it observes the *total momentum* \mathbf{p}_{tot} of the robot, which is defined from (21) as

$$\mathbf{p}_{\text{tot}} = \mathbf{p}_q + \mathbf{p}_\theta = \mathbf{M}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{B}\dot{\theta}. \quad (69)$$

Defining

$$\mathbf{r} = \mathbf{K}_O \left(\hat{\mathbf{p}}_{\text{tot}} - \int_0^t (\tau_m - \hat{\beta}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{r}) ds - \hat{\mathbf{p}}_{\text{tot}}(0) \right) \quad (70)$$

leads again to the same first-order behavior with constant filter frequency as in (68). However, the estimation does not depend on τ_J , i.e., a joint torque sensor is no longer required. In turn, this method is again sensitive to $\tau_{F,\theta}$ and would thus be a suboptimal choice for highly geared robots. Nonetheless, we point out that the residual-based approach can be independently applied also to the motor dynamics (17), so as to provide a model-less estimate also of $\tau_{F,\theta}$. This has been proposed in [67] and used therein to compensate online the motor-side friction.

In summary, due to reasons analogous to the rigid case, the link momentum observer is the best choice for handling collision detection, isolation, and identification with proprioceptive sensing only also for flexible joint robots. An accurate estimation of the external joint torques is obtained, without depending on particular controllers or trajectories nor on motor-side friction. Furthermore, no acceleration measures nor estimates or inversion of the robot mass matrix are required. Indeed, a rather accurate robot dynamic model is needed, and its terms have to be evaluated in real time. However, evaluating the full robot dynamics, which can be easily done nowadays at kilohertz rates, is required anyway for accurate model-based motion control.

Finally, it should be noted that the extension to robots having intrinsically flexible joints with variable stiffness is straightforward as well and is not considered here for brevity (see, e.g., [57], [68], and [69]).

VII. EXPERIMENTAL EVALUATION

Having now determined the best method to solve the context-independent phases of the collision event pipeline for both rigid and flexible robots, we will further elaborate the properties of the velocity and momentum observers. For this, we first compare the dynamics and the impact of parasitic effects on the detection performance. Then, we discuss some performance and timing properties of the *link momentum observer* in detail by analyzing suitable simulations and experiments. Thereafter, collision detection with the *energy observer* and, respectively, collision detection and isolation with the *link momentum observer* are validated in new experiments compared to our previous works. Despite that the presented experiments were carried out with elastic joint robots that are equipped with joint torque sensing, similar conclusions can be drawn for other robot classes.

A. Observer Dynamics Velocity and Link Momentum Observer

The influence of $\mathbf{M}(\mathbf{q})$ on the dynamics of the velocity observer (37) is demonstrated by the following exemplary simulation. Fig. 6 (top) depicts the results for μ_3 of the LWR (flexible joint dynamics) with observer gains $k_{O,i} = 15 \text{ s}^{-1}$ and an unknown external joint torque. The observer is affected by $\mathbf{M}(\mathbf{q})$ resulting in a configuration-dependent behavior and thus in a difficult tuning of the observer dynamics. The plot depicts the results for $\tau_{\text{ext},3} = 10 \text{ Nm}$ when $10 \leq t \leq 80 \text{ ms}$ at initial conditions $\mathbf{q}_{01} = (0, 0, 0, 0, 0, 0, 0)$ rad and $\mathbf{q}_{02} = (0, 1.3, 0, 1.3, 0, 1, 0)$ rad. On the one hand, the current configuration clearly influences the observer dynamics. On the other hand, the nonlinear coupling by $\mathbf{M}(\mathbf{q})$ leads to a deformation of the transient behavior. In contrast, the dy-

TABLE II
COLLISION MONITORING METHODS FOR ROBOTS WITH FLEXIBLE JOINTS

Monitoring method		Measurement quantities
Absolute joint torque	$ \tau_J \geq \tau_{J,\max}$	$\tau_J \vee \hat{\tau}_J = \hat{K}_J(\theta - q)$
Joint torque change	$ \dot{\tau}_J \geq \dot{\tau}_{J,\max}$	$\tau_J \vee \hat{\tau}_J = \hat{K}_J(\theta - q) \quad \dot{\tau}_J \vee \hat{\dot{\tau}}_J = \hat{K}_J(\dot{\theta} - \dot{q})$
Joint torque tube	$\hat{\tau}_{\text{ext}} \approx \tau_{J,\text{ref}}(t) - \tau_J(t)$	$\tau_J \vee \hat{\tau}_J = \hat{K}_J(\theta - q)$
Energy observer	Subst. τ_m with τ_J in Eq. (26); property: Eq. (27)	$q \vee \hat{q} = \theta - \hat{K}_J^{-1} \tau_J \quad \tau_J \vee \hat{\tau}_J = \hat{K}_J(\theta - q)$ $\dot{q} \vee \hat{\dot{q}} = \dot{\theta} - \hat{K}_J^{-1} \dot{\tau}_J$
Direct estimation	Subst. τ_m with τ_J in Eq. (31)	$q \vee \hat{q} = \theta - \hat{K}_J^{-1} \tau_J \quad \ddot{q} \vee \hat{\ddot{q}} = \ddot{\theta} - \hat{K}_J^{-1} \ddot{\tau}_J$ $\dot{q} \vee \hat{\dot{q}} = \dot{\theta} - \hat{K}_J^{-1} \dot{\tau}_J \quad \tau_J \vee \hat{\tau}_J = \hat{K}_J(\theta - q)$
Inverse dynamics	$\hat{\tau}_{\text{ext}} \approx \hat{\tau}_{J,\text{ff}} - \tau_J$ Eq. (33) yields $\hat{\tau}_{J,\text{ff}}$, not $\hat{\tau}_{m,\text{ff}}$	$\tau_J \vee \hat{\tau}_J = \hat{K}_J(\theta - q)$
Velocity observer	Subst. τ_m with τ_J in Eq. (38); property: Eq. (41)	$q \vee \hat{q} = \theta - \hat{K}_J^{-1} \tau_J \quad \tau_J \vee \hat{\tau}_J = \hat{K}_J(\theta - q)$ $\dot{q} \vee \hat{\dot{q}} = \dot{\theta} - \hat{K}_J^{-1} \dot{\tau}_J$
Link momentum observer	Eq. (67); property: Eq. (68)	$q \vee \hat{q} = \theta - \hat{K}_J^{-1} \tau_J \quad \tau_J \vee \hat{\tau}_J = \hat{K}_J(\theta - q)$ $\dot{q} \vee \hat{\dot{q}} = \dot{\theta} - \hat{K}_J^{-1} \dot{\tau}_J$
Total momentum observer	Eq. (70); property: Eq. (68)	$\theta, \dot{\theta} \quad q \vee \hat{q} = \theta - \hat{K}_J^{-1} \tau_J$ $\dot{q} \vee \hat{\dot{q}} = \dot{\theta} - \hat{K}_J^{-1} \dot{\tau}_J$

namics of the momentum observer (44) does not depend on the robot configuration [see Fig. 6(bottom)].

B. Observer Errors and Thresholding

In general, one of the challenges when implementing collision detection algorithms is tuning of the detection thresholds. In observer-based methods, model uncertainties such as unknown friction torques, quantization effects, measurement noise, or delay are generally included in the torque/joint torque estimations and thus result in $\mu \neq 0$ even for $\tau_{\text{ext}} = 0$. In order to compare the robustness of the velocity and the momentum observer, a simulation of robot movements was carried out for the LWR III (flexible joint dynamics), taking into consideration the following effects.

- 1) The measurement of θ is quantized with a resolution of 400 increments on the motor side and filtered with a first-order filter (cutoff frequency $f_\theta = 300$ Hz).
- 2) The measurement of τ_J is affected by uniformly distributed noise of ± 0.3 Nm, a hysteresis (width $\Delta\tau_{\text{hys}} = 0.2$ Nm), quantization of 12-bit resolution and filtered with a first-order filter (cutoff frequency $f_{\tau_J} = 300$ Hz).
- 3) The harmonic drive gears are assumed to produce a ripple torque with amplitude $\tau_{\text{HD,max}} = 0.2$ Nm at a frequency, which is twice that of the motor-side velocity $\dot{\theta}$.
- 4) The link-side position q is estimated via $\hat{q} = \theta - \hat{K}_J^{-1} \tau_J$.
- 5) The link-side velocity \dot{q} is calculated by numerical differentiation of the estimated position \hat{q} and filtered with a first-order filter (cutoff frequency $f_{\dot{q}} = 300$ Hz).
- 6) The link-side Coulomb friction torque $\tau_{f_{\dot{q}}}$ is considered with $\tau_{f_{\dot{q}}} = \pm 0.5$ Nm.
- 7) The estimated end-effector mass considered in $\hat{M}(q)$, $\hat{c}(q, \dot{q})$, and $\hat{g}(q)$ is only 50 % of the real mass.
- 8) The measurements of θ and τ_J are delayed by $t_{d,\theta} = 1$ ms and $t_{d,\tau_J} = 1$ ms, respectively.

Fig. 7 depicts the results for $k_{O,i} = 25 \text{ s}^{-1}$, an initial position $q_0 = (0, 0.79, 0, 1.57, 0, 0.61, 0.79)$ rad, and a sinusoidal desired position around q_0 with an amplitude of 0.6 rad and frequencies in the range of $0.75 \leq \omega \leq 1.6$ rad/s varying for the different axes. The effects mentioned above are enabled for $1 \text{ s} < t < 4 \text{ s}$. It can be seen that both observers work well under ideal conditions, whereas considerable differences appear under realistic conditions. The behavior of both observers is comparable for axes 3 and 4, whereas the velocity observer shows slightly better performance in axes 1 and 2. In axes 5–7, however, the momentum observer allows much smaller thresholds for detecting external joint torques.

Further evaluations have shown that the effect of \hat{q} can be neglected compared to the parasitic effects on the measurements of τ_J and \dot{q} . Fig. 8 depicts a detailed analysis by simulating the estimation μ_2 of the momentum observer for axis 2, when only one effect at a time is taken into consideration. Moreover, all simulations were carried out without any measurement noise, since its effect depends strongly on the used filters. Thus, it cannot be distinguished from other effects. In summary, f_{τ_J} and $f_{\dot{q}}$ as well as t_{d,τ_J} have the most significant influence and require special care in the system design and practical implementation.

C. Performance of the Link Momentum Observer

1) *Desired Monitoring Bandwidth:* In order to ensure the highest quality of estimation of the collision joint torque τ_{ext} , the largest possible K_O should obviously be chosen. However, noise deteriorates the monitoring signal and in practice K_O is limited from above. To assess the best tradeoff, we should analyze the behavior of μ , i.e., r in (67), with respect to the observer gain K_O , and verify how well the true external joint torques can be estimated in a realistic (worst-case) rigid collision.

For evaluating the influence of K_O on the transient behavior of the observer for a very wide range of values of K_O , a sim-

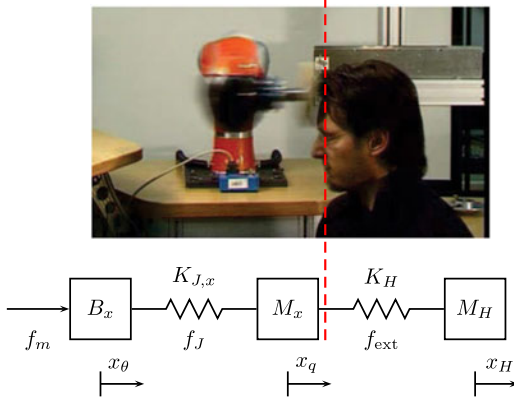


Fig. 9. Hard collision between robot and human head (top) and dynamic model in the operational space (bottom).

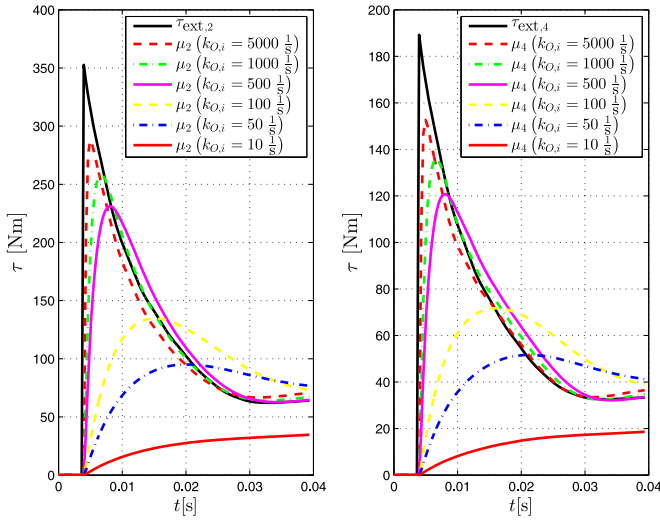


Fig. 10. Comparison between the external joint torque τ_{ext} and the momentum-based monitoring signals μ for different values of $K_O = \text{diag}\{k_{O,i}\}$ at joint 2 (left) and 4 (right) in the flexible joint robot–human head collision simulation of Fig. 9.

ulation study of impacts between the DLR/KUKA LWR and a human head has been carried out, using the complete flexible joint dynamic model for the robot and the following data for the human head: frontal head stiffness $K_H = 10^5$ N/m, head mass $M_H = 4.5$ kg. Fig. 9 depicts the simplified operational space model along the instantaneous unit collision direction \mathbf{u} [70]. The robot is in the stretched configuration ($\mathbf{q} = \mathbf{0}$) when it hits the head with its last link ($i_c = 7$) at a collision velocity $\|\dot{\mathbf{x}}_c\| \approx 2$ m/s. Along the impact direction \mathbf{u} , the reflected link and motor inertias are $M_x \approx 4$ kg and $B_x \approx 13$ kg, respectively. The reflected joint stiffness is $K_{J,x} \approx 7000$ N/m. The maximum collision force is $\|\mathbf{f}_{ext}\|_{\max} \approx 500$ N. Fig. 10 shows the time evolution (in the first 40 ms after the collision) of the two components of μ associated with the second and fourth robot joints, for different values of K_O and in comparison with the actual τ_{ext} . For a fast and almost ideal collision detection, assuming a nonzero but small threshold ϵ_μ , the gain K_O should be set to at least 500 s^{-1} in order to make the measure μ increase almost immediately after the collision. However, for the



Fig. 11. Collision experiment (DLR/KUKA LWR and Hybrid III dummy).

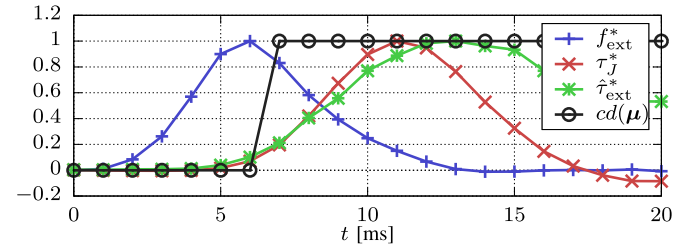


Fig. 12. Timing of relevant normalized (*) signals for an impact experiment between a DLR/KUKA LWR robot and a Hybrid III Dummy at 1-kHz recording time.

robot considered in the experiments that have been carried out, the diagonal entries of K_O may take in practice values between 25 s^{-1} and a maximum of 75 s^{-1} . While simulations were performed under ideal conditions (i.e., exact model and no noise), noisy sensors, model uncertainties, as well as communication delays induce the upper bound in real applications. In turn, heavier filtering would lead to lower (delayed) detection, isolation, and identification performance.

2) *Signal Timing*: Further important characteristics of collision handling based on joint torque sensing and link position estimation in flexible joint robots can be unveiled when taking a closer look at the timing of relevant signals. For this, a real collision between the DLR/KUKA LWR robot and the frontal head area of a Hybrid III crash-test dummy has been conducted emulating a high-speed collision between robot and human (see Fig. 11).

For an impact at $t = 1$ ms with an end-effector velocity of 2 m/s, Fig. 12 shows the time evolution of some relevant signals in the first few milliseconds. They are all normalized to their maximum value attained during the collision experiment (normalized quantities are labeled by an asterisk): the contact force f_{ext} measured via a high-speed 20-kHz bandwidth force sensor; the measured joint torque τ_J at joint 4; the estimation of the external torque τ_{ext} at joint 4 caused by f_{ext} ; and finally, the binary collision detection signal $cd(\mu)$.

The robot-head contact lasts for about 10 ms. Most noticeably, $\hat{\tau}_{ext}^*$ starts raising significantly only about 5 ms later than f_{ext}^* , which leads also to $cd(\mu) = \text{TRUE}$ with the same delay. Ideally, if no structural elasticities were present, one could indeed detect a collision instantaneously by setting $\epsilon_\mu \approx 0$. However, joint torque measurement (with designated joint torque

sensors or by evaluating the joint deformation) and lightweight design introduce structural joint and link flexibility. The unmodeled link flexibility would lead to false detection positives. Therefore, nonnegligible positive collision thresholds ϵ_μ have to be used and exceeded before a collision can be reliably detected. Furthermore, the reflected joint stiffness $K_{J,x}$ is significantly lower than the stiffness K_H of the dummy frontal area. Thus, the robot links will act as a mechanical low-pass filter placed before the position and joint torque sensor locations. In turn, τ_J^* , which is the input to the observer, will react in a delayed fashion to a change in f_{ext}^* . Since link position and velocity are usually estimated via $\hat{q} = \theta - K_J^{-1}\tau_J$ and $\dot{\hat{q}} = \dot{\theta} - K_J^{-1}\dot{\tau}_J$, they will suffer from the same delayed response. In fact, the measured joint torque and its derivative, both scaled by the inverse of the large joint stiffness, are the dominant factors in the rate of change of link position and velocity estimates, respectively. In combination with the presence of a positive detection threshold, this causes a significant delay in detecting the collision, since the monitoring signal activates only as soon as the measurements start reacting to the impact.

Before the collision is detected, the joint torque τ_J^* is almost zero because the robot moves at constant velocity without acceleration. After the collision, $cd(\mu)$ remains true even after the contact force vanished. Indeed, $cd(\mu)$ will return to false when normal robot operation is resumed, according to the reaction strategy. The estimated external joint torque $\hat{\tau}_{\text{ext}}^*$ represents a sort of impulse response of the sensor and observer filter characteristics and does not significantly decrease (following f_{ext}^*) within the shown, extremely short time interval. The joint torque τ_J^* does not immediately return to zero (following f_{ext}^*) but indicates the reaction of the mechanical structure to the impact, in this case a damped oscillation.

D. Collision Detection with an Energy Observer

In order to showcase in a single experiment all aspects of the energy-based observer, as elaborated in Section III-A and Table II, we have considered the effects of collisions, while a DLR/KUKA LWR was at rest for most of the time or was moving along a straight line in the (x, y) -plane under a Cartesian impedance control law during specific time intervals. Fig. 13 shows the obtained results, reporting the desired and actual robot Cartesian motion (the two components of the Cartesian position x and velocity \dot{x} in the horizontal plane), the external contact force f_{ext} , the scalar collision monitoring signal μ , as given by (28) with τ_J replacing τ_m , and, finally, the detection sensitivity measure $s = |\hat{f}_{\text{ext}}^T \dot{\hat{x}}|$.

The series of experimental phases indicated by A–F are characterized by the following behaviors.

- 1) Phase A: The external force is correctly recognized by the detection scheme, as it acts against the motion direction while the robot moves. The minimal delay between external force and collision detection is not visible in the used time scale.
- 2) Phase B: No external force is applied; the robot is at rest.
- 3) Phase C: The external force applied in the (x, y) plane increases gradually during standstill, but it cannot be detected since the robot does not move, thus $s \approx 0$.

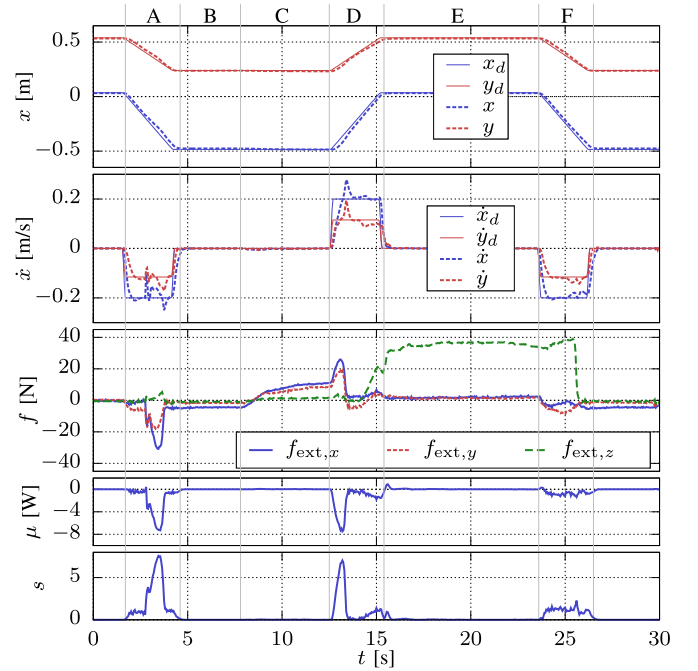


Fig. 13. Experimental behavior of the collision detection based on the energy observer. From the top: desired \mathbf{x}_d (continuous line) and actual Cartesian collision position (dashed line) \mathbf{x} ; desired $\dot{\mathbf{x}}_d$ and actual Cartesian collision velocity $\dot{\mathbf{x}}$; external contact force \mathbf{f}_{ext} measured via a wrist force/torque sensor; monitoring signal μ ; sensitivity s of the detection scheme.

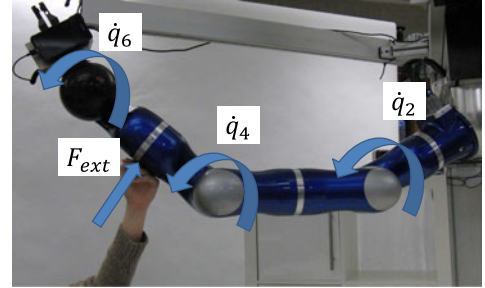


Fig. 14. Collision detection and isolation experiment with the link momentum observer (67).

- 4) Phase D: The robot starts moving again, which immediately leads to an increase in sensitivity s and consequently in the monitoring signal μ .
- 5) Phase E: While the robot is at rest, a strong force is applied in the z -direction, without being detected.
- 6) Phase F: The robot moves again, while the z -force is still being applied. However, since it acts orthogonally to the motion vector, it cannot be recognized ($s \approx 0$).

E. Collision Detection/Isolation with a Link Momentum Observer

The experimental setup of the collision detection and isolation experiment using the link momentum observer (67) is shown in Fig. 14. The DLR/KUKA LWR robot moves joints 2, 4, and 6, under full state feedback joint position control law with gravity compensation at maximum speeds between 50 and 80 °/s. The desired trajectory is a rest-to-rest fifth-order polynomial with

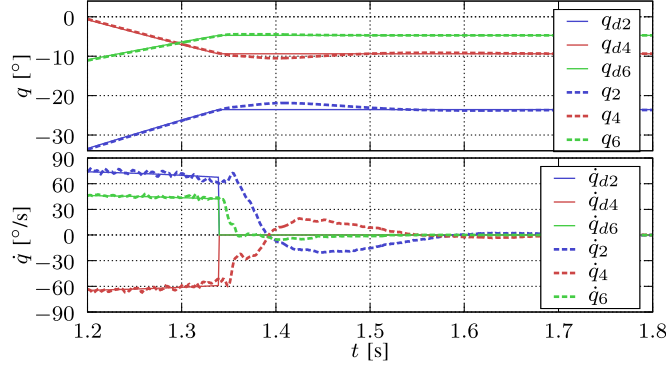


Fig. 15. Experimental behavior of the collision detection and isolation scheme based on the link momentum observer (67) for the experiment in Fig. 14. Desired and actual position (top) and velocity (bottom) of joints 2, 4, and 6 (the first 1.2 s of the rest-to-rest motion trajectory are omitted).

total motion time $T = 2$ s. The remaining four joints are all kept fixed at $q_i = 0$, for $i = 1, 3, 5, 7$.

Fig. 15 shows the desired and measured joint position and velocity during the time interval $1.2 \leq t \leq 1.8$ s. Around the instant of maximum trajectory speed at $t \approx 1.3$ s, the human blocks the motion with his hand by hitting the robot at link 4. As soon as the collision is detected, the desired velocity switches to zero, i.e., a controlled emergency stop is activated (in fact the simplest possible reaction strategy; see [40]). The robot instantaneously executes the desired stop and brings itself to rest in about 200 ms.

Fig. 16 illustrates the monitoring joint torques μ , together with the binary collision detection signals $cd(\mu)$ and constant thresholds ϵ_μ . The latter were set to be $\epsilon_\mu = 0.05 \cdot \tau_{J,\max}$, where $\tau_{J,\max} = [180 \ 180 \ 80 \ 80 \ 35 \ 35 \ 35]^T$ Nm. This is a rather sensitive threshold, which, however, does not lead to any false positives over the entire workspace. The monitoring signals at joints 2 and 4 indicate a collision, while those at joints 1 and 3 remain silent, since the Cartesian contact force does not produce action on these joint axes. Equation (58) provides $i_c = 4$, so the correct colliding link has been isolated.

F. Estimating the External Force and Contact Point

The following experiment aims at showing the performance of the contact force isolation and identification schemes from Sections IV-B and IV-C. Specifically, a mass of 2.4 kg fixed at the end effector (and unknown to the robot controller) had to be identified during motion. The resulting external wrench

$$\mathcal{F}_{\text{ext}} = \begin{bmatrix} \mathbf{f}_{\text{ext}} \\ \mathbf{0} \end{bmatrix} \quad (71)$$

is estimated based on $\mu = \hat{\tau}_{\text{ext}}$ from the momentum observer (67) and on the pseudoinversion (61). Fig. 17 shows that the force components $f_{\text{ext},x} = 0$, $f_{\text{ext},y} = 0$, and $f_{\text{ext},z} = 2.4g \approx 23.54$ N are correctly reconstructed by $\hat{\mathbf{f}}$ even during stepwise changes of the end-effectors velocity. The estimated moment $\hat{\mathbf{m}}$ is not zero because the given $\hat{\mathbf{f}}$ and $\hat{\mathbf{m}}$ are related to the base frame of the robot. The moment \hat{m}_y , therefore, varies with the x -position of the robot, which directly influences the

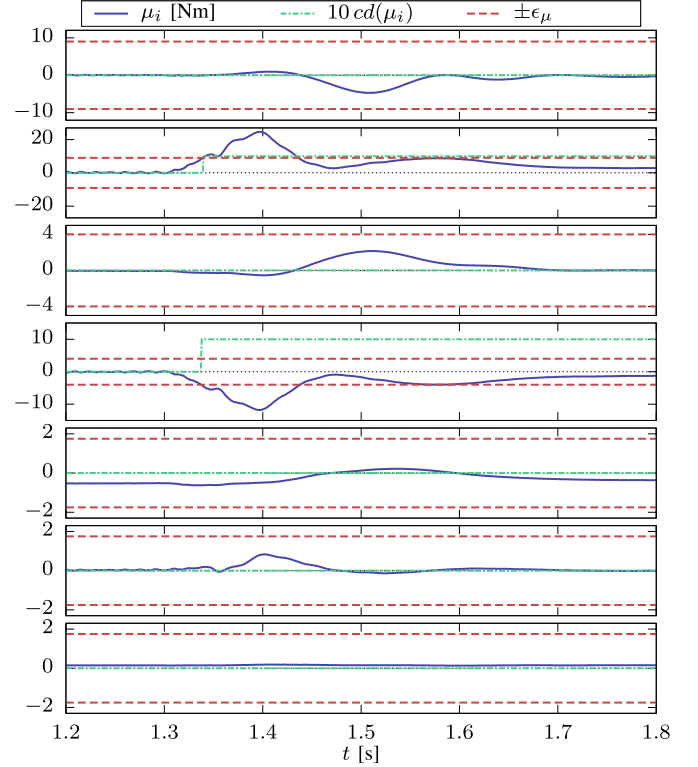


Fig. 16. Monitoring joint torques μ , collision detection signals $cd(\mu)$, and detection thresholds ϵ_μ for joints 1–7 in the experiment of Figs. 14 and 15. According to (58), the collision is correctly isolated at link $i_c = 4$.

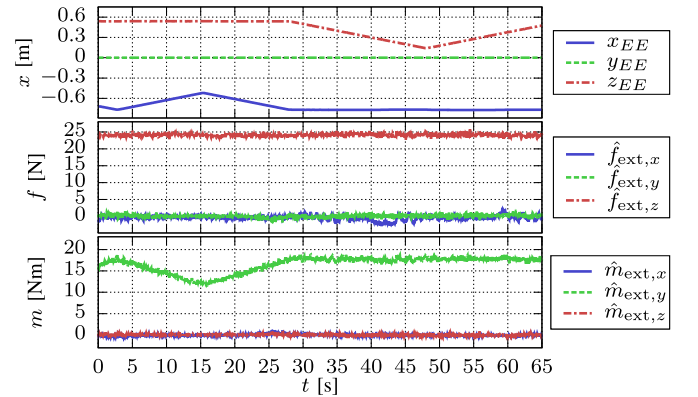


Fig. 17. End-effector position (top), reconstructed constant force (center), reconstructed moment (bottom) caused by a mass of 2.4 kg. Force and moment are expressed with respect to the robot base frame.

lever arm of the external force around the robot base. The z -position is orthogonal to the lever arm and has no impact on the moment.

Subsequently, in order to demonstrate the ability to reconstruct the collision point, the experiment was extended. A mass of 1 kg was successively placed at different distances $x_{6,c}$ from the center of the robot spherical wrist (see Fig. 18). The following values of $x_{6,c}$ were tested during the experiment: 0.187 m, 0.237 m, 0.287 m, 0.337 m, 0.387 m, and 0.437 m. During the experiments, the robot was at rest.

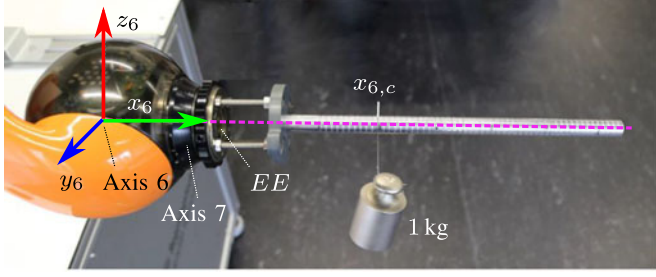


Fig. 18. Mass of 1 kg is placed on the end-effector (joint 7) at different distances $x_{6,c}$ from the joint axis 6.

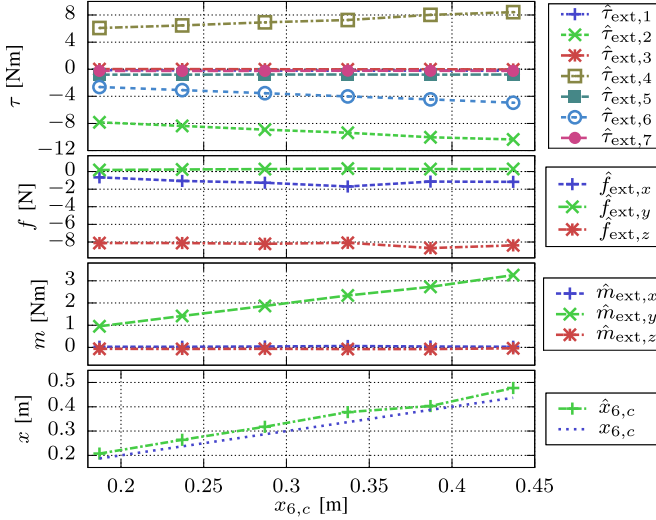


Fig. 19. Estimated external joint torques $\mu = \hat{\tau}_{\text{ext}}$ (top), estimated wrench components in frame 6 (middle plots), and estimated versus actual distance of the mass from the joint axis 6 (bottom), for six different placements of the 1-kg mass on the robot end-effector.

As in the previous experiments, the resulting wrench was computed from the estimated external joint torque $\mu = \hat{\tau}_{\text{ext}}$, which was obtained by the generalized momentum observer [see Fig. 19(top)]. It can be seen that for the given configuration of the robot, only joints 2, 4, and 6 are affected. Consequently, the contact location lies between joint 6 and 7. Alternatively, it lies in positive x_6 -direction because wrenches of the form (71) do not have any influence on the joint torque in joint 7 when acting on the x_6 -axis. Therefore, the estimation of the contact locations is based on

$$\hat{\mathcal{F}}_6 = \begin{bmatrix} \hat{\mathbf{f}}_6 \\ \hat{\mathbf{m}}_6 \end{bmatrix} = (\mathbf{J}_6^T(\mathbf{q}))^\# \mu. \quad (72)$$

From (64), the position vector of the line of force action is found as

$$\mathbf{r}_{6,d} = (\mathbf{S}^T(\hat{\mathbf{f}}_6))^\# \hat{\mathbf{m}}_6 \quad (73)$$

where

$$\mathbf{S}^T(\hat{\mathbf{f}}_6) = \begin{bmatrix} 0 & \hat{f}_{6,z} & -\hat{f}_{6,y} \\ -\hat{f}_{6,z} & 0 & \hat{f}_{6,x} \\ \hat{f}_{6,y} & -\hat{f}_{6,x} & 0 \end{bmatrix}. \quad (74)$$

Finally, the intersection of the line of force action (represented in the frame of joint 6) $\mathbf{r}_{6,d} + \lambda \hat{\mathbf{f}}_6$ with the robot structure is the estimated contact point. In our case, a contact between joint 6 and 7 can be ruled out by the length of the position vector $\mathbf{r}_{6,d}$. Consequently, the intersection of the line of force action with the robot structure is obtained by solving

$$\mathbf{r}_{6,d} + \lambda \hat{\mathbf{f}}_6 = \nu \mathbf{e}_{x_6} \quad (75)$$

where \mathbf{e}_{x_6} is the unit vector along the axis x_6 . This finally leads to

$$\begin{bmatrix} \lambda^* \\ \nu^* \end{bmatrix} = [-\hat{\mathbf{f}}_6 \quad \mathbf{e}_{x_6}]^\# \mathbf{r}_{6,d}. \quad (76)$$

The results are depicted in Fig. 19. The estimated force in the z -direction corresponds to the unknown mass of 1 kg. The contact point estimation works well, although it provides slightly larger values for the distances between the axis of joint 6 and the contact point. The estimation error

$$e_{6,c} = \|\mathbf{r}_{6,d} + \lambda^* \hat{\mathbf{f}}_6 - \nu^* \mathbf{e}_{x_6}\| \quad (77)$$

is $0.42 \text{ cm} \leq e_{6,c} \leq 1.16 \text{ cm}$, i.e., reasonably small and practically useful.

VIII. CONCLUSION

The collision event pipeline has been introduced as a formal abstraction for the seven phases of robot collision handling. In particular, we elaborated in this paper on the theoretical foundations of the three context-independent phases, namely collision detection, isolation, and identification. In this single framework, we described common industrial as well as advanced research algorithms for generating collision monitoring signals. All methods rely on proprioceptive information only and were developed both for rigid and flexible joint robots. The state-of-the-art algorithms resort to energy arguments or to the robot generalized momentum. We provided a systematic overview and comparison of all schemes in their use during collision detection, isolation, and identification. Furthermore, we have discussed the computational demands and general advantages and disadvantages of each method. The presented experimental and simulation results support the theoretical findings and provide insight into collision dynamics and its implications on detection speed, as well as the effects of relevant nonidealities.

It should be noted that within a technology transfer, two of our methods were commercialized. In fact, the collision detection method based on the link momentum observer, which was found to be the best performing one, and the end-effector contact wrench identification algorithm have found their way into the KUKA LBR product family [55] and most recently into FRANKA EMIKA [47]. Our approaches made it possible to apply this new generation of lightweight robots to a large variety of novel tasks in automobile and electronics production and assembly. Processes could be automated, which were fully manual before, and our collision handling technology is regarded as the key element for enabling safe physical human–robot interaction in real-world applications.

ACKNOWLEDGMENT

The authors would like to thank R. Weitschat, K. Krieger, and L. Johannsmeier for their help with the experiments. They would also like to particularly thank T. Lilge for his support with the revision of the paper.

REFERENCES

- [1] A. Bicchi, M. A. Peshkin, and J. E. Colgate, "Safety for physical human-robot interaction," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Berlin, Germany: Springer, 2008, pp. 1335–1348.
- [2] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi, "An atlas of physical human-robot interaction," *Mech. Mach. Theory*, vol. 43, no. 3, pp. 253–270, 2008.
- [3] J. Frémy, F. Michaud, and M. Lauria, "Pushing a robot along—A natural interface for human-robot interaction," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 3440–3445.
- [4] J. Vorndamme, M. Schappler, and S. Haddadin, "Collision detection, isolation and identification for humanoids," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 4754–4761.
- [5] T. Tomic and S. Haddadin, "Simultaneous estimation of aerodynamic and contact forces in flying robots: Applications to metric wind estimation and collision detection," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 5290–5296.
- [6] F. Flacco, A. Paolillo, and A. Kheddar, "Residual-based contacts estimation for humanoid robots," in *Proc. 16th IEEE Int. Conf. Humanoid Robots*, 2016, pp. 409–415.
- [7] J.-C. Latombe, *Robot Motion Planning*. Norwell, MA, USA: Kluwer, 1991.
- [8] R. Diankov and J. Kuffner, "OpenRAVE: A planning architecture for autonomous robotics," *Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA*, Tech. Rep. CMU-RI-TR-08–34, Jul. 2008.
- [9] I. A. Sucan and S. Chitta, MoveIt!, Aug. 31, 2016. [Online]. Available: <http://moveit.ros.org>
- [10] E. A. Sisbot and R. Alami, "A human-aware manipulation planner," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1045–1057, Oct. 2012.
- [11] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 299–306.
- [12] D. Ebert and D. Henrich, "Safe human-robot-cooperation: Image-based collision detection for industrial robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2002, pp. 239–244.
- [13] S. Kuhn and D. Henrich, "Fast vision-based minimum distance determination between known and unknown objects," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot Syst.*, 2007, pp. 2186–2191.
- [14] F. Flacco, T. Kröger, A. De Luca, and O. Khatib, "A depth space approach to human-robot collision avoidance," in *Proc. IEEE Conf. Robot. Autom.*, 2012, pp. 338–345.
- [15] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1985.
- [16] C. Warren, "Global path planning using artificial potential fields," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1989, pp. 316–321.
- [17] O. Brock and O. Khatib, "Elastic strips: A framework for motion generation in human environments," *Int. J. Robot. Res.*, vol. 21, no. 12, pp. 1031–1052, 2002.
- [18] H. Minoura, R. Kijima, and T. Ojika, "Collision avoidance using a virtual electric charge in the electrostatic potential field," in *Proc. Int. Conf. Virtual Syst. MultiMedia*, 1996, pp. 289–294.
- [19] J. Roßmann, "On-line collision avoidance for multi-robot systems: A new solution considering the robots' dynamics," in *Proc. IEEE/SICE/RSJ Int. Conf. Multisens. Fusion Integr. Intell. Syst.*, 1996, pp. 249–256.
- [20] P. Ögren, N. Egerstedt, and X. Hu, "Reactive mobile manipulation using dynamic trajectory tracking," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2000, pp. 3473–3478.
- [21] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "CHOMP: Gradient optimization techniques for efficient motion planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 489–494.
- [22] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic trajectory optimization for motion planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 4569–4574.
- [23] C. Park, J. Pan, and D. Manocha, "Real-time optimization-based planning in dynamic environments using GPUs," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 4090–4097.
- [24] K. Suita, Y. Yamada, N. Tsuchida, K. Imai, H. Ikeda, and N. Sugimoto, "A failure-to-safety "kyozon" system with simple contact detection and stop capabilities for safe human - Autonomous robot coexistence," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1995, pp. 3089–3096.
- [25] Y. Yamada, Y. Hirasawa, S. Huang, Y. Umetani, and K. Suita, "Human-robot contact in the safeguarding space," *IEEE/ASME Trans. Mechatronics*, vol. 2, no. 4, pp. 230–236, Dec. 1997.
- [26] S. Takakura, T. Murakami, and K. Ohnishi, "An approach to collision detection and recovery motion in industrial robot," in *Proc. 15th Annu. Conf. IEEE Ind. Electron. Soc.*, 1989, pp. 421–426.
- [27] S. Morinaga and K. Kosuge, "Collision detection system for manipulator based on adaptive impedance control law," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, pp. 1080–1085.
- [28] K. Kosuge, T. Matsumoto, and S. Morinaga, "Collision detection system for manipulator based on adaptive control scheme," *Trans. Soc. Instrum. Control Eng.*, vol. 39, pp. 552–558, 2003.
- [29] V. Lumelsky and E. Cheung, "Real-time collision avoidance in teleoperated whole-sensitive robot arm manipulators," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 1, pp. 194–203, Jan./Feb. 1993.
- [30] M. Strohmayer, "Artificial skin in robotics," Ph.D. dissertation, Karlsruhe Inst. Technol., Karlsruhe, Germany, 2012.
- [31] G. De Maria, C. Natale, and S. Pirozzi, "Force/tactile sensor for robotic applications," *Sens. Actuators A, Phys.*, vol. 175, pp. 60–72, 2012.
- [32] R. Dahiya, P. Mittendorfer, M. Valle, G. Cheng, and V. Lumelsky, "Directions toward effective utilization of tactile skin: A review," *IEEE Sens. J.*, vol. 13, no. 11, pp. 4121–4138, Nov. 2013.
- [33] A. De Luca and R. Mattone, "Sensorless robot collision detection and hybrid force/motion control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2005, pp. 1011–1016.
- [34] A. De Luca, A. Albu-Schäffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the DLR-III lightweight manipulator arm," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 1623–1630.
- [35] A. De Luca and R. Mattone, "Actuator fault detection and isolation using generalized momenta," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, pp. 634–639.
- [36] H.-B. Kuntze, C. Frey, K. Giesen, and G. Milighetti, "Fault tolerant supervisory control of human interactive robots," in *Proc. IFAC Workshop Adv. Control Diagnosis*, 2003, pp. 55–60.
- [37] S. Golz, C. Osendorfer, and S. Haddadin, "Using tactile sensation for learning contact knowledge: Discriminate collision from physical interaction," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 3788–3794.
- [38] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "The role of the robot mass and velocity in physical human-robot interaction—Part I: Unconstrained blunt impacts," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 1331–1338.
- [39] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "The role of the robot mass and velocity in physical human-robot interaction - Part II: Constrained blunt impacts," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 1339–1345.
- [40] S. Haddadin, A. Albu-Schäffer, A. De Luca, and G. Hirzinger, "Collision detection and reaction: A contribution to safe physical human-robot interaction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 3356–3363.
- [41] A. De Luca and L. Ferrajoli, "Exploiting robot redundancy in collision detection and reaction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 3299–3305.
- [42] S. Parusel, S. Haddadin, and A. Albu-Schäffer, "Modular state-based behavior control for safe human-robot interaction: A lightweight control architecture for a lightweight robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 4298–4305.
- [43] A. De Luca and R. Mattone, "An adapt-and-detect actuator FDI scheme for robot manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, New Orleans, LA, USA, 2004, pp. 4975–4980.
- [44] S. Haddadin, "Evaluation criteria and control structures for safe human-robot interaction," Master thesis, Tech. Univ. Munich/German Aerosp. Center (DLR), Cologne, Germany, 2005.
- [45] J. Vorndamme, M. Schappler, A. Tödtheide, and S. Haddadin, "Collision detection with a hydraulically actuated robot arm," in *Proc. 8th Int. Workshop Human-Friendly Robot.*, 2015. [Online]. Available: <http://www.hri.ei.tum.de/fileadmin/w00bsw/www/hfr2015/HFR2015Proceedings.pdf>
- [46] KUKA. LBR iiwa, Jul. 15, 2016. [Online]. Available: <http://www.kuka-lbr-iiwa.com/>

- [47] FRANKA EMIKA. This is Franka, Jan. 23, 2017. [Online]. Available: <https://www.franka.de/>
- [48] 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 *Proc. IEEE Int. Conf. Autom. Sci. Eng.*, 2015, pp. 1230–1235.
- [49] D. P. Le, J. Choi, and S. Kang, "External force estimation using joint torque sensors and its application to impedance control of a robot manipulator," in *Proc. 13th Int. Conf. Control, Autom. Syst.*, 2013, pp. 1794–1798.
- [50] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL, USA: CRC Press, 1994.
- [51] B. Vanderborght *et al.*, "Variable impedance actuators: A review," *Robot. Auton. Syst.*, vol. 61, no. 12, pp. 1601–1614, 2013.
- [52] A. De Luca and W. Book, "Robots with flexible elements," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Berlin, Germany: Springer, 2008, pp. 287–319.
- [53] M. W. Spong, "Modeling and control of elastic joint robots," *ASME J. Dyn. Syst. Meas. Control*, vol. 109, no. 4, pp. 310–319, 1987.
- [54] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, and G. Hirzinger, "The DLR lightweight robot—Lightweight design and soft robotics control concepts for robots in human environments," *Ind. Robot.*, vol. 34, no. 5, pp. 376–385, 2007.
- [55] R. Bischoff *et al.*, "The KUKA-DLR lightweight robot arm: A new reference platform for robotics research and manufacturing," in *Proc. 41st Int. Symp. Robot.*, 2010, pp. 1–10.
- [56] A. De Luca, D. Schröder, and M. Thümmel, "An acceleration-based state observer for robot manipulators with elastic joints," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 3817–3823.
- [57] S. Haddadin, *Towards Safe Robots - Approaching Asimov's 1st Law (ser. Springer Tracts in Advanced Robotics)*, vol. 90. Berlin, Germany: Springer, 2014.
- [58] O. Foellinger, *Regelungstechnik (in German)*. Heidelberg, Germany: Huethig, 1992.
- [59] A. De Luca and L. Ferrajoli, "A modified Newton-Euler method for dynamic computations in robot fault detection and control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 3359–3364.
- [60] V. Sotoudehnejad, A. Takhar, M. Kermani, and I. Polushin, "Counteracting modeling inaccuracies in torque observer-based collision detection using time-variant thresholds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 4315–4320.
- [61] R. Isermann, *Fault-Diagnosis Systems*. Berlin, Germany: Springer, 2006.
- [62] L. Manuelli and R. Tedrake, "Localizing external contact using proprioceptive sensors: The contact particle filter," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 5062–5069.
- [63] M. Fumagalli *et al.*, "Force feedback exploiting tactile and proximal force/torque sensing," *Auton. Robots*, vol. 33, no. 4, pp. 381–398, 2012.
- [64] E. Magrini, F. Flacco, and A. De Luca, "Estimation of contact forces using a virtual force sensor," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 2126–2133.
- [65] A. Albu-Schäffer, C. Ott, and G. Hirzinger, "A unified passivity-based control framework for position, torque and impedance control of flexible joint robots," *Int. J. Robot. Res.*, vol. 26, pp. 23–39, 2007.
- [66] G. Pratt and M. Williamson, "Series elastic actuators," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1995, pp. 399–406.
- [67] L. Le Tien, A. Albu-Schäffer, A. De Luca, and G. Hirzinger, "Friction observer and compensation for control of robots with joint torque measurement," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 3789–3795.
- [68] A. De Luca, F. Flacco, A. Bicchi, and R. Schiavi, "Nonlinear decoupled motion-stiffness control and collision detection/reaction for the VSA-II variable stiffness device," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 5487–5494.
- [69] S. Haddadin, A. Albu-Schäffer, O. Eiberger, and G. Hirzinger, "New insights concerning intrinsic joint elasticity for safety," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 2181–2187.
- [70] O. Khatib, "Inertial properties in robotic manipulation: An object-level framework," *Int. J. Robot. Res.*, vol. 14, no. 1, pp. 19–36, 1995.



Sami Haddadin (M'11) received degrees in electrical engineering, computer science, and technology management from Technical University of Munich, Munich, Germany, and Ludwig Maximilian University of Munich, Munich. He then received the Ph.D. degree in safety in robotics from RWTH Aachen University, Aachen, Germany, in 2011.

He is a Full Professor and Director of the Institute of Automatic Control, Leibniz University of Hannover, Hannover, Germany. He organized/edited several international robotics conferences and journals and authored/coauthored more than 120 scientific articles. His research interests include physical human–robot interaction, nonlinear robot control, real-time motion planning, real-time task and reflex planning, robot learning, optimal control, human motor control, variable impedance actuation, and safety in robotics.

Dr. Haddadin received numerous awards at the top international robotics conferences and journals, including the 2015 IEEE/RAS Early Career Award, the 2015 RSS Early Career Spotlight, and the 2015 Alfried Krupp Award. In 2015 and 2016, he was selected as a Capital Young Elite Leader under 40 in Germany for the domain "Politics, State & Society."



Alessandro De Luca (F'07) received the Ph.D. degree in systems engineering in 1987 from Sapienza Università di Roma, Rome, Italy.

Since 2000, he has been a Full Professor of robotics and automatic control with the Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma. In 2005–2006, he spent a sabbatical period with Institute of Robotics and Mechatronics of DLR—German Aerospace Center, Oberpfaffenhofen, with a Helmholtz-Humboldt Research Award for foreign scientists, where the research in this paper was initiated. His research interests include dynamic modeling, motion planning, and control of robots, in particular of manipulators with elastic joints or flexible links and of kinematically redundant robotic systems, and safe control of physical human–robot interaction.

Dr. De Luca was an Editor (1998–2003) and Editor-in-Chief (2003–2004) of IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION and the first Editor-in-Chief of IEEE TRANSACTIONS ON ROBOTICS (2004–2008). He was the General Chair of the 2007 IEEE International Conference on Robotics and Automation and the Program Chair of the 2016 IEEE International Conference on Robotics and Automation. He was the Coordinator of the EU FP7 project SAPHARI in 2011–2015.



Alin Albu-Schäffer (F'16) received the Engineering degree in electrical engineering from Technical University of Timișoara, Timișoara, Romania, in 1993 and the Ph.D. degree in control systems from Technical University of Munich, Munich (TUM), Germany, in 2002.

Since 1995, he has been with the Institute of Robotics and Mechatronics, German Aerospace Center—DLR, Weßling, Germany. In 2012, he became the Head of this institute and a Full Professor with the Technical University of Munich. His research interests include robot design, modeling and control, flexible joint and variable compliance robots for manipulation and locomotion, physical human–robot interaction, and bioinspired robot design.

Dr. Albu-Schäffer, and his team, received numerous awards, including the IEEE King-Sun Fu Best Paper Award of IEEE TRANSACTIONS ON ROBOTICS in 2012 and 2014, and several IEEE International Conference on Robotics and Automation and IEEE/RSJ International Conference on Intelligent Robots and Systems Best Paper Awards as well as the DLR Science Award.