IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

Aim-Aware Collision Monitoring: Discriminating between Expected and Unexpected Post-Impact Behaviors

Benn Proper[1], Alexander Kurdas[2], Saeed Abdolshah[2], Sami Haddadin[2], Alessandro Saccon[1]

***Abstract*— To speed up and reduce power consumption per cycle in robotic manipulation, one option is to exploit intentional collisions with the surrounding environment and objects, an approach referred to as impact-aware manipulation. Within this context, this paper focuses on developing an online collision monitoring framework for distinguishing between expected and unexpected post-impact behaviors. The classification is based on a desired post-impact motion created via an idealized rigid robot-object-environment model. To generate a classification error bound, it employs a causal envelop filter that is needed due to the unavoidable joint and environment flexibility. In this way, it becomes possible to compare a desired idealized rigid response, which is straightforward to obtain with existing tools, with a measured impact response, which is affected by difficult-to-model post-impact oscillations. The classifier can be used for single-contact as well as multi-contact impact scenarios, such as those occurring in surface-to-surface impacts, and allows for tuning of the sensitivity between expected and unexpected post-impact behaviors. The monitoring framework fuses a (bandpass) momentum observer with impact-aware control to extend the classical collision event pipeline. As a proof of concept, we show the effectiveness of the approach through numerical simulations as well as with preliminary experimental results.**

***Index Terms*— Failure Detection and Recovery, Contact Modeling, Perception for Grasping and Manipulation**

I. Introduction

STATE-OF-THE-ART physical interaction control typically resorts to enforcing zero or near-zero relative contact velocity between a robot and its environment [1] and treats collisions as unexpected and undesired behaviors [2]. However, the development of robots with flexible joints [3], [4] combined with validation of robot-environment impact laws [5], [6] is opening the door to the exploitation of intentional dynamic contact transitions for manipulation and locomotion. This exciting field of research, that we refer to as *impact-aware robotics*, requires the development of a new holistic framework comprising modeling, learning, planning, sensing, and control aspects, supported by collision-tolerant hardware.

In this context, we focus on the sensing aspect by proposing a strategy capable of distinguishing between expected and unexpected robot-environment impacts. We name this strategy *aim-aware collision monitoring* to emphasize that classification is performed based on the knowledge that the robot has an explicit intention of impacting the environment as well as a notion of what is expected to happen following the impact. Successfully distinguishing between expected and unexpected impact scenarios gives the robot the knowledge to attempt a recovery action, which can include stopping the execution or activating a reflex motion. This allows for more robust task execution while potentially reducing damage to the robot and environment. Schematically, Fig. 1 depicts the proposed aim-aware collision monitoring method, where classification is based on analyzing a post-impact velocity error signal passed through a signal envelope filter.

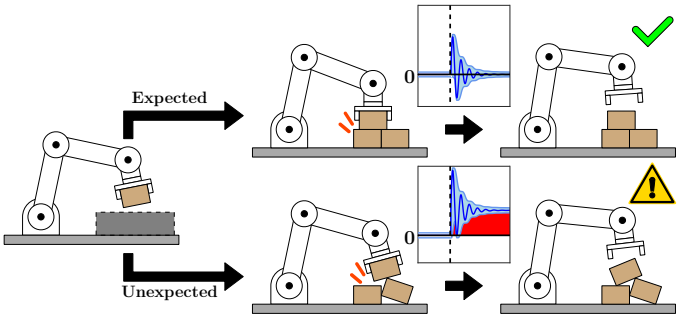


Fig. 1: The proposed aim-aware collision monitoring method in action. Using the error between the predicted and measured impact response, a signal envelope is generated. If the region inside the envelope (shaded light blue) contains 0, the impact sequence is classified as expected. If it does not (the difference is shaded red), the impact sequence is classified as unexpected.

Related works. Originally intended for, but not limited to, physical human-robot interaction (pHRI), the well-known *collision event pipeline* [7] systematizes the contact handling problem by dividing it into *seven* phases (namely, pre-collision, detection, isolation, identification, classification, reaction, and post-collision). State-of-the-art approaches for the detection, isolation, and identification phases work regardless of the impact source [8], [9], [10], [11], [12]. However, for the classification and reaction phases, state-of-the-art approaches focus on pHRI [8], [13], [2], [14], resulting in robot-environment impacts being classified as unintentional or as intentional to stop the robot from, e.g., initiating a hand-guiding teaching phase. Furthermore, current classification methods rely on machine learning to discriminate among different types of collisions [8], [9], [15], [16], [17], where the pHRI intention is captured implicitly. For the emerging field of impact-aware robotics, the focus must instead be placed on comparing the planned post-impact behavior with the observed post-impact behavior of the robot-object-environment interactions. Within this context, the post-impact velocity jump, requiring some a priori knowledge of the environment, can be predicted either through analytical or learning approaches [5], [6].

Paper contribution. In this paper, we propose an extension to the collision event pipeline for intentional robot-environment and robot-object interactions and a first-of-its-kind classifier

Manuscript received: January, 20, 2023; Revised April, 26, 2023; Accepted May, 20, 2023.

This paper was recommended for publication by Editor Clément Gosselin upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the European Union's H2020 research and innovation program as part of the project I.AM. under grant no. 871899.

[1]B. Proper and A. Saccon are with the Department of Mechanical Engineering, Eindhoven University of Technology (TU/e), 5600MB Eindhoven, The Netherlands {b.w.b.proper, a.saccon}@tue.nl

[2]A. Kurdas, S. Abdolshah, and S. Haddadin are with the Chair of Robotics and Systems Intelligence, MIRMI - Munich Institute of Robotics and Machine Intelligence, Technical University of Munich (TUM), 80992 Munich, Germany, {alexander.kurdas, saeed.abdolshah, haddadin}@tum.de

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

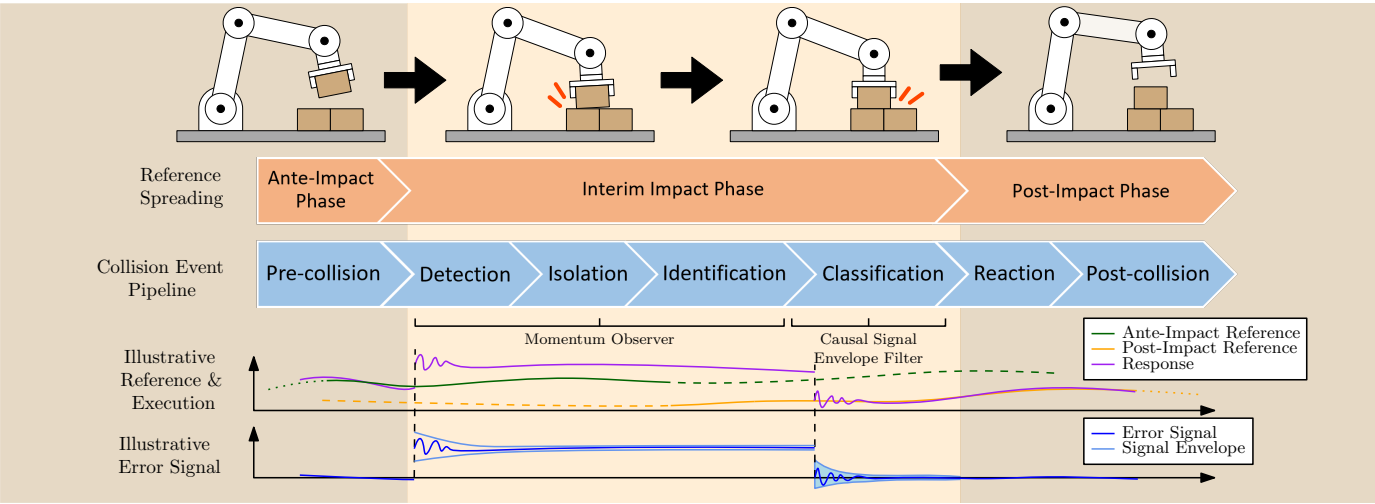


Fig. 2: The proposed aim-aware collision monitoring method. For context, the top row shows an impact sequence for a simultaneous impact. The second row shows the impact phases used in reference spreading and the coinciding collision event pipeline phases are shown in the third row. The fourth and fifth rows show an illustrative example of a reference signal with an impact response and the error signal with a signal envelope, respectively. The data from the bottom row is used to assess whether an impact occurred as expected. Illustrated damped oscillations, mostly due to joint flexibility, commonly last for about 100 ms.

for classifying post-impact behavior. We show that rigid object and environment models with instantaneous and inelastic nonsmooth impact laws [18] allow for effective robot-environment impact classification with an adjustable classification threshold. Classification is performed in two steps, (1) through static conditions that provide an initial guess for the classification at the time of impact, and (2) through post-impact analysis inspired by [5], where the post-impact oscillations in the order of 100 ms are interpreted as fast dynamics superimposed to the rigid nonsmooth impact response. Emphasis is placed on classifying intentional *simultaneous* impacts, where multiple contact points ideally make contact at the same time. We employ the notion that an inelastic impact law is capable of capturing the gross rigid velocity jump when contact is fully established [19]. Our classifier is supported by extensions to methods to encode impact tasks through reference spreading [20], [21] and detect impacts regardless of the current contact state of the robot using a momentum observer [10], [22], [23]. For illustrative purposes, in Fig. 2, the state of the robotic manipulator is linked to the impact phases used in reference spreading [10], [22], [23] and the collision event pipeline [7] combined with an example showing the reference and execution of the task and the error signal used for classification.

Paper structure. Sec. II presents preliminary material. The aim-aware collision monitor is described in Sec. III. Numerical and experimental validation of the approach is presented in Sec. IV. Concluding remarks are provided in Sec. V.

II. Preliminaries

The monitoring method uses the robot dynamics (Sec. II-A) and impact dynamics (Sec. II-B) to predict the impact behavior, classification is then initiated using impact detection derived from the momentum observer (Sec. II-C).

A. Dynamical Model

For the numerical simulations, we adopt the standard robot with flexible joints (RFJ) model [24]

$$\begin{aligned} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) &= \boldsymbol{\tau}_J + \mathbf{D}\mathbf{K}^{-1}\dot{\boldsymbol{\tau}}_J + \boldsymbol{\tau}_{ext}, \\ \mathbf{B}\ddot{\boldsymbol{\theta}} + \mathbf{D}\mathbf{K}^{-1}\dot{\boldsymbol{\tau}}_J + \boldsymbol{\tau}_J &= \boldsymbol{\tau}_{act}, \end{aligned} \tag{1}$$

where $\mathbf{q} \in \mathbb{R}^n$ are the link-side joint positions, $\boldsymbol{\theta} \in \mathbb{R}^n$ are the motor positions adjusted with gear ratio, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) \in \mathbb{R}^{n\times n}$ is the Coriolis matrix, $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$ is the gravity vector, $\boldsymbol{\tau}_J \in \mathbb{R}^n$ and $\boldsymbol{\tau}_{ext} \in \mathbb{R}^n$ are the spring and external joint torques, respectively, $\mathbf{B} = diag(\mathbf{b}) \in \mathbb{R}^{n\times n}$ is the reflected motor inertia matrix after the transmission, and $\mathbf{D} = diag(\mathbf{d}) \in \mathbb{R}^{n\times n}$ is the joint damping matrix. Here, the $diag$ operation turns a vector of size n into an $n \times n$ diagonal matrix. Angles $\boldsymbol{\theta}$ and $\mathbf{q}$ both contribute to the joint torques via $\boldsymbol{\tau}_J = \mathbf{K}(\boldsymbol{\theta} - \mathbf{q})$, where $\mathbf{K} = diag(\mathbf{k}) \in \mathbb{R}^{n\times n}$ is the joint stiffness matrix. We assume motor friction to be compensated in the motor controller. Finally, $\boldsymbol{\tau}_{act} \in \mathbb{R}^n$ are the actuation torques obtained from a low-level torque controller [24].

B. Simultaneous Inelastic Impact Map

Rigid impact maps consider impact events as instantaneous events and are used to describe the post-impact motion including friction and switching contact for a given impact configuration using nonsmooth mechanics [18], [25], [26]. We derive the simultaneous rigid impact map with the assumptions:

- The robot has rigid dynamics.
- The impact is inelastic.
- Joint displacements remain constant during impact.

Using these assumptions, the impact equation

$$\mathbf{M}_r(\dot{\mathbf{q}}^+ - \dot{\mathbf{q}}^-) = \sum_{i=1}^{n_c} \mathbf{J}_{i,N}^\top \Lambda_{N,i} + \mathbf{J}_{i,T}^\top \Lambda_{T,i}, \tag{2}$$

can be derived from (1). Here, for each contact point i, $\mathbf{J}_{i,N}(\mathbf{q}) \in \mathbb{R}^{1\times n}$ and $\mathbf{J}_{i,T}(\mathbf{q}) \in \mathbb{R}^{1\times n}$ are the normal and tangential components of the Jacobian $\mathbf{J}_i(\mathbf{q})$ as defined in [20], respectively, $\Lambda_{N,i}$ and $\Lambda_{T,i}$ denote the normal and tangential impulsive contact force, respectively, superscripts $(\cdot)^-$ and $(\cdot)^+$ denote an ante- and post-impact quantity, and $\mathbf{M}_r(\mathbf{q}) \in \mathbb{R}^{n\times n}$ is the rigid joint equivalent of $\mathbf{M}(\mathbf{q})$ defined as $\mathbf{M}_r(\mathbf{q}) = \mathbf{M}(\mathbf{q}) + \mathbf{B}_\theta$ [27], with $\mathbf{B}_\theta \in \mathbb{R}^{n\times n}$ apparent reflected motor inertia matrix after gear reduction. Furthermore, Newton's impact law can be written as

$$\mathbf{J}_{i,N}\dot{\mathbf{q}}^+ = \mu_N \mathbf{J}_{i,N}\dot{\mathbf{q}}^-. \tag{3}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

In the simplest case, assuming that the impact is frictionless ($\Lambda_{T,i} = \mathbf{0}$) and inelastic ($\mu_N = 0$), (2) and (3) are used to derive the simultaneous impact map as

$$\begin{bmatrix} \dot{\mathbf{q}}^+ \\ \Lambda_{N,1} \\ \vdots \\ \Lambda_{N,n_c} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_r & -\mathbf{J}_{1,N}^\top \dots -\mathbf{J}_{n_c,N}^\top \\ \mathbf{J}_{1,N} & 0 \quad \dots \quad 0 \\ \vdots & \vdots \quad \ddots \quad \vdots \\ \mathbf{J}_{n_c,N} & 0 \quad \dots \quad 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{M}_r \dot{\mathbf{q}}^- \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \quad (4)$$

Note, (4) is applicable to the simple case considered in this paper because the (unilateral) constraints are linearly independent, thus constraining different degrees of freedom. Suitable regularization for more complex impact scenarios considering more than two contact points in 2D and three in 3D and friction is found in available solvers[1]. Additionally, while we assume a frictionless impact for simplicity, the accuracy of the prediction for real systems can be improved using models that consider impacts with friction [28], [29].

C. Momentum Observer

The state-of-the-art impact detection methods are based on external joint torque estimators such as the momentum observer [10], [22], [23]. The momentum observer can isolate the impacted link and identify the external torque using the generalized momentum $\boldsymbol{p} \in \mathbb{R}^n$ defined as

$$\boldsymbol{p} = \mathbf{M}(\mathbf{q})\dot{\mathbf{q}}. \quad (5)$$

Using (1), (5), and the known property that $\dot{\mathbf{M}}(\mathbf{q}) = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})^\top$, the momentum observer is defined as

$$\mathbf{r}(t) = \mathbf{K}_O \Bigg[\boldsymbol{p}(t) - \int_0^t (\boldsymbol{\tau}_J + \mathbf{D}\mathbf{K}^{-1}\dot{\boldsymbol{\tau}}_J + \mathbf{C}^\top(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}) + \mathbf{r}(s))ds - \boldsymbol{p}(0) \Bigg], \quad (6)$$

with $\mathbf{r}$ satisfying the dynamic constraint

$$\dot{\mathbf{r}} = \mathbf{K}_O \left(\boldsymbol{\tau}_{ext} - \mathbf{r} \right). \quad (7)$$

To detect an impact, a threshold can be defined for the residual $\mathbf{r} \in \mathbb{R}^n$: if an entry of $\mathbf{r}$ surpasses this threshold during operation, then that link is affected by an impact.

III. Aim-aware collision monitoring

The proposed aim-aware collision monitoring method is a framework that requires three main components to function. These components are (1) encoding of an impact task, (2) detecting impacts regardless of the current contact state, and (3) relevant classification criteria. For the first two components, we provide extensions to existing methods for use with our classifier in Sec. III-A and Sec. III-B. For the classification criteria, we propose the distinction between an instantaneous and time-window impact classification strategy. Sec. III-C describes an approach to defining static conditions that can be assessed at the time of impact and provides an initial guess on whether the impact was expected. Sec. III-D then describes a method that refines the classification by analyzing post-impact oscillations against the predicted motion using an online signal envelope filter. All of these components are compiled into a complete classifier in Sec. III-E.

[1]For example nonsmooth.gricad-pages.univ-grenoble-alpes.fr/siconos

A. Encoding the Intentional Impact Task

Model-based planners for control systems undergoing impacts is an active field of research [30]. For the robust execution against modeling and perception errors, we use an impact-aware control strategy called reference spreading [20], [21], [31]. Here, we generate ante- and post-impact trajectories that are extended so that overlap in time is present before and after the nominal impact time [32], [33]. The intended impact motion is encoded into these trajectories by having the velocity jump from ante- to post-impact trajectory at the intended impact time t_d satisfy an impact law such as (4).

For illustration purposes, in this work, we consider a planar motion of the end effector on a vertical plane orthogonal to an impact surface as sketched in Fig. 2. For the end-effector longitudinal and vertical displacements (x,y) and orientation (ϕ) we define desired ante- and post-impact positions/orientations and their first- and second-order time derivatives for t_0, t_d, and t_f, where t_d is the desired impact time. The post-impact velocities at t_d are determined using the simultaneous impact map (4) with the ante-impact velocities at t_d as input as presented in [20]. With these desired parameters, fifth-order polynomials can be generated for $t_0 \leq t \leq t_d$ and $t_d \leq t \leq t_f$. An example of a trajectory is shown in Fig. 3.

The manipulator is commanded via an impact-aware impedance controller that follows an ante-impact trajectory until an impact is detected, after which the reference switches to the post-impact trajectory [31]. To take into account small deviations from the expected impact surface, the post-impact motion in the y direction is controlled by applying a constant force on the impact surface.

B. Simultaneous Impact Detection

In Sec. II-C, impacts are detected by defining a threshold against $\mathbf{r}$, but this does not guarantee an impact detection if the system is already in contact with the environment. Applying a high-pass filter to $\mathbf{r}$ would allow for detecting fast changes in torques [2], but this implementation was not designed for discrete time. To allow for detecting impacts in simultaneous impact scenarios in discrete time, a threshold against the numerical time derivative of (6) defined as

$$\frac{\Delta \mathbf{r}_k}{\Delta t} = \frac{\mathbf{r}_k - \mathbf{r}_{k-1}}{t_k - t_{k-1}}, \quad (8)$$

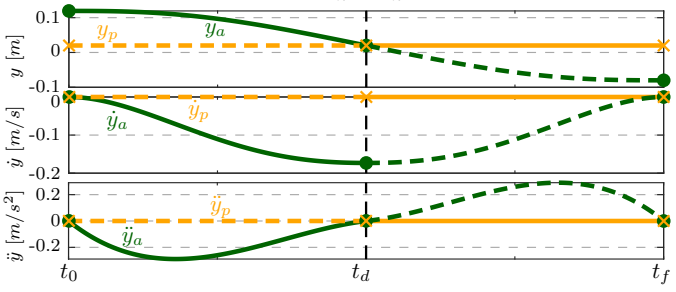


Fig. 3: Schematic illustration of the ante- and post-impact trajectories in relation to the desired trajectory parameters. Ante- and post-impact trajectories are denoted by $(\cdot)_a$ (green) and $(\cdot)_p$ (ocra), respectively. Solid lines are the desired impact trajectories, while dashed lines are extended trajectories. Green and ocra markers represent the desired trajectory parameters.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

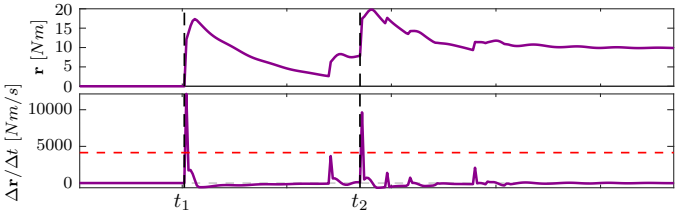


Fig. 4: The momentum observer residual and its numerical derivative during a simulated simultaneous impact sequence. The dashed red line is the impact detection threshold $\mathbf{r}_b$. By adjusting $\mathbf{r}_b$, smaller impacts that are of lesser interest for the classification method can be included or excluded.

is used instead. The differences between $\mathbf{r}$ and $\Delta\mathbf{r}/\Delta t$ are shown in Fig. 4 for a simulated simultaneous impact scenario. The threshold against $\Delta\mathbf{r}/\Delta t$ allows for detecting impacts regardless of the current contact state. The detection criterion with detection bound $\mathbf{r}_b$ is then formulated as:

Time instant $t_k = k\Delta t$ is considered an impact time if and only if there exists an entry $i \in \{1,\dots,n\}$ of $\Delta\mathbf{r}_k$ such that

$$\left|\frac{\Delta\mathbf{r}_{k,i}}{\Delta t}\right| \geq \mathbf{r}_{b,i}, \qquad \textit{and} \qquad \left|\frac{\Delta\mathbf{r}_{k-1,i}}{\Delta t}\right| < \mathbf{r}_{b,i}. \tag{9}$$

C. Instantaneous Impact Classification

Instantaneous impact classification seeks to provide an initial guess at the classification result at the time of impact with low computational burden through data available at the time of impact, the expected behavior, and user-defined thresholds. In this paper, the impact trajectory is designed around one (ideally) simultaneous impact. We can then define a threshold for the error between the detected impact times and the desired impact time t_d as

$$\left\|t_j - t_d\right\| \leq \zeta, \quad \text{for } j \in \left\{1,\dots,j_f\right\} \tag{10}$$

where j_f is the index of the final detected impact and, similarly, for the measured and predicted configurations

$$\left\|\mathbf{q}(t_j) - \mathbf{q}_d(t_d)\right\| \leq \epsilon, \quad \text{for } j \in \left\{1,\dots,j_f\right\}, \tag{11}$$

where t_j is the j-th detected impact time, t_d is the desired impact time for a single expected impact, $\zeta > 0$ is the user-defined time difference threshold, $\mathbf{q}_d \in \mathbb{R}^n$ is the expected joint configuration, and $\epsilon > 0$ is the user-defined impact configuration error threshold. For simultaneous impact classification performance, the time difference between individual impacts is also considered

$$\left\|t_j - t_{j-1}\right\| \leq \iota, \quad \text{for } j \in \left\{2,\dots,j_f\right\}, \tag{12}$$

where $\iota > 0$ is the admissible time difference between two consecutive impacts.

D. Time-Window Impact Classification

Time-window impact classification focuses on using the time evolution of the impact response for classification to verify that the post-impact behavior agrees with the intended motion. It occurs once contact has been completed, e.g., when the entire end-effector is in contact with the impact surface (after the impact time and configuration have been verified during the instantaneous phase described previously). Time-window classification is slower and more computationally expensive but it allows for enhancing considerably the sensitivity of the classifier by incorporating for the first time the intended post-impact motion. It is the core of our approach.

As was shown in [5], post-impact oscillations can be interpreted as fast dynamics superimposed to the rigid nonsmooth impact response before converging. For simultaneous impact scenarios, this can be formulated as

$$\mathbf{e}_{vel}(t) \to \mathbf{0}_{n\times 1}, \quad \text{for } t \in \left(t_{j_f}, t_{end,j_f}\right], \tag{13}$$

where

$$\mathbf{e}_{vel}(t) = \dot{\mathbf{q}}(t) - \dot{\mathbf{q}}_d(t). \tag{14}$$

Here, $\mathbf{e}_{vel}(t) \in \mathbb{R}^n$ is the error between the measured joint velocity $\dot{\mathbf{q}}(t) \in \mathbb{R}^n$ and the expected post-impact joint velocity $\dot{\mathbf{q}}_d(t) \in \mathbb{R}^n$ which is obtained from the post-impact reference trajectory discussed in Sec. III-A. Finally, $t_{end,j_f} = t_{j_f} + \xi$ where $\xi \in \mathbb{R}_{\geq 0}$ is the tunable time until final classification, which is dependent on the natural impact-induced oscillations.

Signal envelope detection [34] is used here to extract the envelope of the measured oscillatory impact response. The resulting envelope can then be related to the equivalent rigid response following the slow-fast post-impact dynamics philosophy [5]. The required envelope filter would need to be causal to allow for online post-impact classification. This causal signal envelope filter is defined using the element-wise minima and maxima operation on a vector signal over a prescribed window length

$$\begin{aligned} \mathbf{x}_{min,k} &= \min\left\{\mathbf{x}_{k-m},\dots,\mathbf{x}_k\right\}, \\ \mathbf{x}_{max,k} &= \max\left\{\mathbf{x}_{k-m},\dots,\mathbf{x}_k\right\}. \end{aligned} \tag{15}$$

Here, $\mathbf{x}_{min,k} \in \mathbb{R}^n$ and $\mathbf{x}_{max,k} \in \mathbb{R}^n$ are defined as a vector containing the minima and maxima of a measured response $\mathbf{x} \in \mathbb{R}^n$ at timestep k, determined from data from timestep $k-m$ to timestep k where $m \in \mathbb{N}$ is the desired window length. Afterward, (15) can be used to define an upper and lower envelope that can adjust to a newly measured extremum

$$\begin{aligned} \mathbf{x}_{lb,k} &= \min\left\{\mathbf{x}_{min,k}, (1-a\Delta t)\,\mathbf{x}_{lb,k-1} + a\Delta t\,\mathbf{x}_{min,k}\right\}, \\ \mathbf{x}_{ub,k} &= \max\left\{\mathbf{x}_{max,k}, (1-a\Delta t)\,\mathbf{x}_{ub,k-1} + a\Delta t\,\mathbf{x}_{max,k}\right\}, \end{aligned} \tag{16}$$

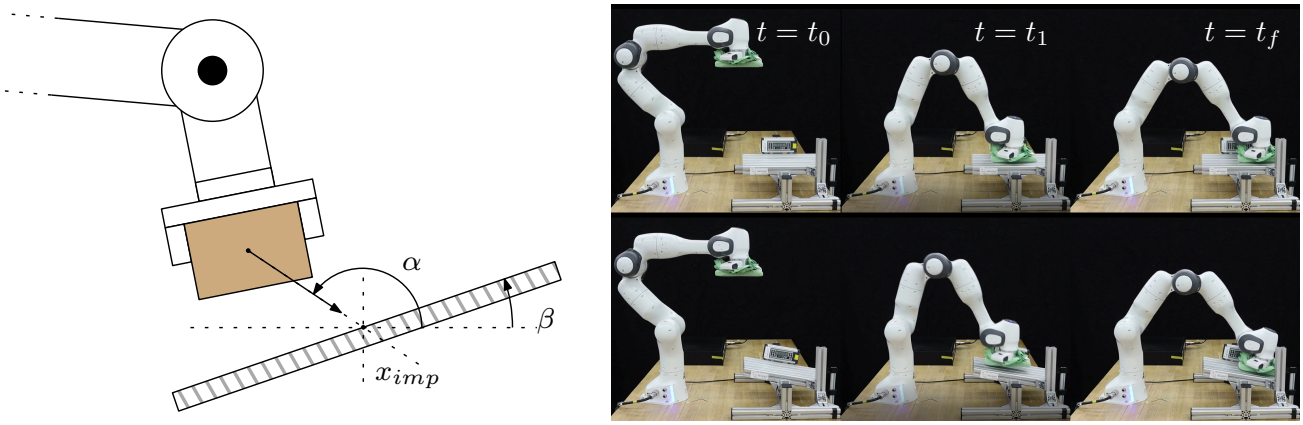
where $a \in \mathbb{R}_{>0}$ is the decay rate of the envelope that is chosen depending on the decay rate of the impact response, $\Delta t = t_k - t_{k-1}$ is the sampling time, $\mathbf{x}_{lb} \in \mathbb{R}^n$ and $\mathbf{x}_{ub} \in \mathbb{R}^n$ are the lower- and upper envelope of the signal, respectively.

The filter described in (15) and (16) can be used on an arbitrary oscillatory signal $\mathbf{x}$ to produce an envelope. In this work, $\mathbf{x}$ is substituted by $\mathbf{e}_{vel}$, and the time-window classification condition is defined as

$$\begin{aligned} \mathbf{e}_{vel,lb}(t) - e_m \leq \mathbf{0}_{n\times 1} \leq \mathbf{e}_{vel,ub}(t) + e_m, \\ \forall t \in \left(t_{j_f}, t_{end,j_f}\right], \end{aligned} \tag{17}$$

where $\mathbf{e}_{vel,lb} \in \mathbb{R}^n$ and $\mathbf{e}_{vel,ub} \in \mathbb{R}^n$ are the lower- and upper envelope encompassing signal $\mathbf{e}_{vel}$ as calculated using (15) and (16), respectively. Additionally, $e_m \in \mathbb{R}_{\geq 0}$ is an additional user-defined margin on the envelope, which can be changed depending on the noise level, the tracking error, and other user-defined requirements. For simplicity, we choose e_m to be uniform over all joints. However, intuitively and assuming the impact happens at the end-effector, the joints closer to the base will be perturbed less severely. For enhanced accuracy, e_m could be chosen to be smaller for joints closer to the base, but investigating the benefit is left for future work.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.



(a) **(b)**

Fig. 5: Defining various impact scenarios. (a) The impact approach angle α, impact surface inclination β, and desired impact location x_{imp} are defined. (b) A snapshot of experiments where the robot is impacting the impact platform for an expected (top) and unexpected (bottom) impact scenario at starting time t_0, first detected impact time t_1, and end time t_f.

The application of the causal signal envelope filter to $\mathbf{e}_{vel}$ for simulated and experimental expected and unexpected scenarios are shown in Fig. 6. In this figure, the area enclosed by the envelope shaded in blue is used for classification. If this envelope encompasses 0, as described in (17), the impact is classified as expected. If not, the impact is classified as unexpected. To make it visible when the envelope does not encompass 0, in Fig. 6 a red shade is explicitly drawn.

E. Classifier construction

For the full classifier, further conditions can be defined to reduce false positive classification. Firstly, with simultaneous impacts, the system's response between the first and final impact, also known as *the interim phase* [20], [21], is unpredictable and full contact needs to be established before full state feedback control can be applied. As a consequence of this unpredictability and to avoid the interim phase affecting the generated envelope, the generated envelope and the window length are reset after each detected impact. For each iteration afterward, the window length increases by 1, until it reaches the desired maximum window length $m_{max} \in \mathbb{N}$.

Secondly, the ideal post-impact trajectory based on a fully rigid robot model and rigid contact assumes an instant change in velocity after an impact while a realistic response is fast but non-instantaneous due to the robot and environment flexibility. To take this difference into account, an additional tuning parameter $m_c \in \mathbb{N}$, the number of timesteps after an impact before time-window classification starts, is defined. Using this parameter, classification only starts when

$$t_k \geq m_c \Delta t + t_j, \tag{18}$$

The resulting schematic representation of the classifier is shown in Algorithm 1.

Algorithm 1: Impact classifier

initialisation: Impact = YetUnknown
InstImpact$_0$ = YetUnknown
Post-Impact$_0$ = Off
$m_1 = 1$
input : t_k, t_j, $\mathbf{q}(t)$, $\mathbf{e}_{vel}$, $\mathbf{e}_{vel,ub,k-1}$, $\mathbf{e}_{vel,lb,k-1}$, m_k, InstImpact$_{k-1}$, Post-Impact$_{k-1}$
output : $\mathbf{e}_{vel,ub,k}$, $\mathbf{e}_{vel,lb,k}$, m_{k+1}, InstImpact$_k$, Post-Impact$_k$
parameter : a, ϵ, ζ, ξ, ι, e_m, m_c, m_{max}, t_d, $\mathbf{q}_d(t_d)$, Δt

if t_k = *impact time according to* (9) **then**
  $m_k = 1$
  if (10) *and* (11) *and* (12) **then**
    InstImpact$_k$ = Expected
    Post-Impact$_k$ = YetUnknown
  else
    InstImpact$_k$ = Unexpected
    Post-Impact$_k$ = Unexpected
  end
  $\mathbf{e}_{vel,lb,k} = \mathbf{e}_{vel,k}$
  $\mathbf{e}_{vel,ub,k} = \mathbf{e}_{vel,k}$
else
  Calculate $\mathbf{e}_{vel,lb,k}$ and $\mathbf{e}_{vel,ub,k}$ using (15) and (16) with $\mathbf{e}_{vel}$
  InstImpact$_k$ = InstImpact$_{k-1}$
  Post-Impact$_k$ = Post-Impact$_{k-1}$
end
if *Post-Impact*$_k$ = *YetUnknown* **then**
  if (¬ (17)) *and* (18) **then**
    Post-Impact$_k$ = Unexpected
  else
    if $t_k - t_j > \xi$ **then**
      Post-Impact$_k$ = Expected
    end
  end
end
if *Post-Impact*$_k \neq$ *YetUnknown and* $t_k - t_j > \xi$ **then**
  if *InstImpactt*$_k$ = *Expected and Post-Impact*$_k$ = *Expected* **then**
    Impact = Expected
  else
    Impact = Unexpected
  end
end
if $m_k < m_{max}$ **then**
  $m_{k+1} = m_k + 1$
end

IV. Validation

Validation of the proposed aim-aware collision monitoring method is performed numerically (Sec. IV-A) and through a preliminary experimental validation (Sec. IV-B).

A. Numerical validation

A graphical illustration of the chosen validation impact scenario and snapshots taken from experiments are presented in Fig. 5. The simulation[2] uses the robot dynamics described in Sec. II-A and simulates realistic frictionless contact using the exponentially extended Hunt-Crossley model [35], [36].

The robot is tasked to impact a surface with a flat end-effector with an approach angle α and impact location x_{imp},

[2]Found at gitlab.tue.nl/robotics-lab-public/aim-aware-collision-monitoring

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

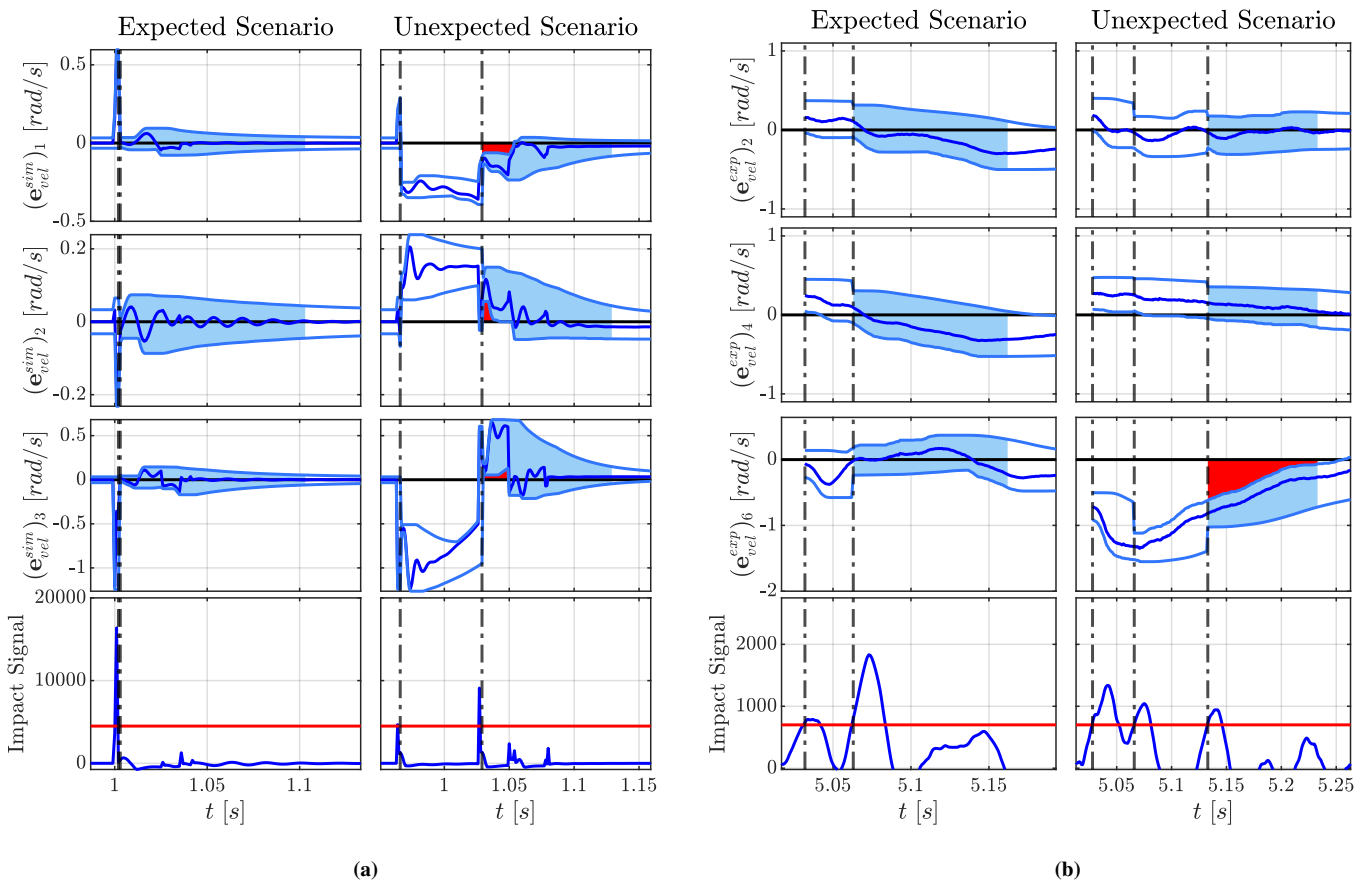


Fig. 6: Post-impact behavior classification using the causal signal envelope filter applied to the joint-velocity error for expected and unexpected impact scenarios. (a) Simulated simultaneous impact scenarios. (b) Experimental simultaneous impact scenarios. In these figures, the bright blue line represents the joint velocity error $\mathbf{e}_{vel}$, which is encompassed by the envelope generated by the causal signal envelope filter. The interval used for classification is represented with the blue shaded area between the envelope bounds. If condition (17) was not satisfied, the difference is shaded in red.

measured horizontally. The impact approach angle α ranges from 60 and 120 deg with a 2 deg resolution and the desired impact location x_{imp} from 0.2 to 0.4 m with a 0.02 m resolution. Without loss of generality, the desired impact motion assumes the surface orientation β, to be zero and will generate a trajectory as described in Sec. III-A in all experiments. In the simulation, the surface orientation with respect to the horizontal line ranges from -10 to 10 deg with a resolution of 1 deg. We expect the collision monitor to be able to discriminate between an expected and unexpected value of β, depending on user-adjustable thresholds.

Classification results for simulated simultaneous impacts are presented in Fig. 7 for three different sets of tuning parameters, which are found in Tab. 1. The different blue shades in Fig. 7 illustrate that the sensitivity of the classifier can be tuned to change the range of inclinations that are classified as expected. Additionally, a quite thin area of about 2 deg, where the impact is not always classified as expected or unexpected, is present on both sides of the blue band for all sensitivities. Finally, beyond this ambiguous area, 100% of the experiments are classified as unexpected. Using the size of the ambiguous area as a performance measure, we can observe the effectiveness of the collision monitoring method and the consistent performance between sensitivities.

Currently, there is no systematic approach to tune the classifier and a particular range of inclinations that are always classified as expected can only be obtained by hand-tuning the classifier. The instantaneous classification conditions can be correlated to desired impact properties, as they refer to desired outcomes of the impact trajectory. Time-window classification parameters can be extracted from measured post-impact motion behavior, but extracting these values for classifier tuning is not the focus of this work.

B. Preliminary Experimental Validation

The aim-aware collision monitor has been tested on a Franka Emika Research robot arm [37] as illustrated in Fig. 5(b). The robot arm is equipped with a custom-made 3D-printed flat end-effector wrapped in a soft cloth to provide additional damping. The impact surface, made of Item profiles, can be rotated around a central axis and is held in place by tightening the screws on the guide rail, where the inclination is measured using a digital goniometer[3].

To assess the functioning of the classifier on an experimental setup, four impact trajectories are considered which differ in the approach angle α ranging from 100 to 130 deg, with a resolution of 10 deg, while keeping the impact location constant at $x_{imp} = 0.51$. The vertical height, measured with respect to the robot base, was set to $z_{imp} = 0.15$ m to ensure that the impact target was at the center of the impact platform. The impact platform inclinations β were varied between -20 to 20 deg with a resolution of 2 deg for two sets

[3]Stabila TECH 1000 DP

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

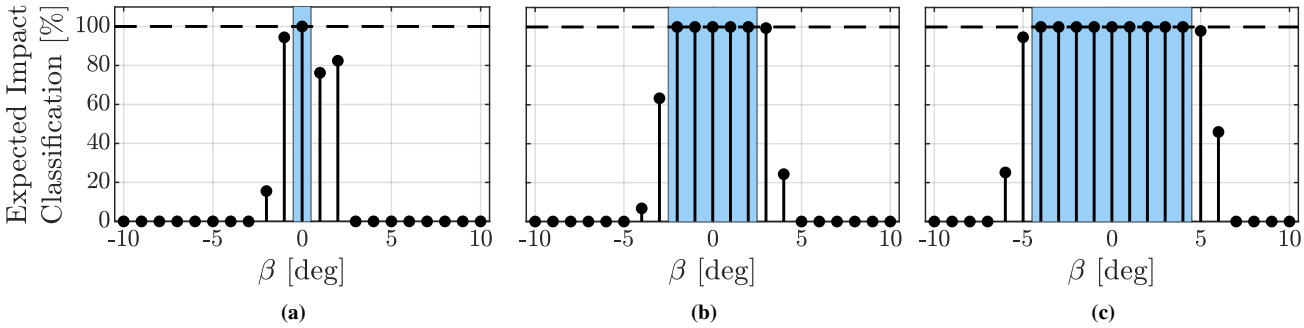


Fig. 7: Expected impact classification percentage for simulated simultaneous impact scenarios when tuned for varying sensitivities. The blue band represents the range of inclinations that are always classified as expected. (a) The tuned classifier always classifies $\beta = 0$ as expected. (b) The tuned classifier always classifies $\beta \in \{-2, \dots, 2\}$ as expected. (c) The tuned classifier always classifies $\beta \in \{-4, \dots, 4\}$ as expected.

Parameter	Fig. 7(a)	Fig. 7(b)	Fig. 7(c)	Unit
m_{max}	10	15	15	–
m_c	10	15	15	–
a	30	18	18	$1/s$
ϵ	0.042	0.067	0.115	rad
ζ	0.1	0.1	0.11	s
ξ	0.1	0.1	0.1	s
e_m	0.033	0.1	0.236	rad
ι	0.1	0.1	0.1	s

Tab. 1: Tuning parameters for simulated simultaneous impact classification. Each column is associated with its respective figure in Fig. 7

Parameter	Fig. 8(a)	Fig. 8(b)	Unit
m_{max}	13	13	–
m_c	5	5	–
a	19	19	$1/s$
ϵ	0.12	0.15	rad
ζ	0.1	0.1	s
ξ	0.1	0.1	s
e_m	0.18	0.2	rad
ι	0.1	0.1	s

Tab. 2: Tuning parameters for experimental simultaneous impact scenarios. Each column is associated with its respective figure in Fig. 8.

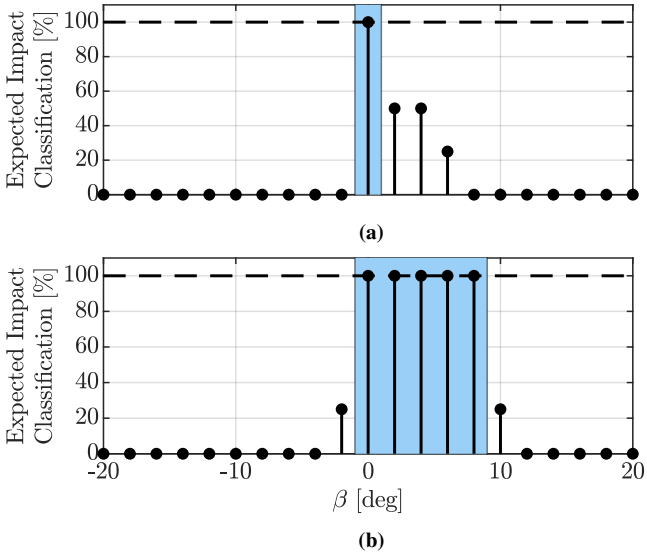


Fig. 8: Expected impact classification percentage for simultaneous impact scenarios on the experimental setup tuned for varying sensitivities. The blue band represents the range of inclinations that are always classified as expected. (a) The tuned classifier classifies $\beta = 0$ as expected. (b) The tuned classifier classifies $\beta \in \{0, \dots, 8\}$ as expected.

of tuning parameters to vary the classification sensitivity. The parameter values and associated results can be found in Tab. 2 and Fig. 8, respectively.

Currently, computational routines and validation of simultaneous impact maps on real robots such as those developed in [5], [6] and introduced in Sec. II-B are not yet readily available. Therefore, the post-impact velocity predictions used in Sec. III-A were substituted with data-driven predictions extracted from post-impact motion behavior of the four nominal impact scenarios in the nominal condition $\beta = 0$. These experiments intend to show that our classifier is effective as long as a reference trajectory that embeds the gross velocity jump induced by the desired impact is available. Such a gross velocity jump can be obtained both from a data-driven approach and a model-based approach, the second having our preference for scalability to arbitrary impact scenarios. Additionally, we make the underlined assumption that desired impact trajectories are such that a unique post-impact outcome is present (typically true with inelastic impact, with additional conditions on the constraints [25] and that small perturbations do not destroy this property but rather present a form of continuity that is discussed, e.g., in [19]).

Fig. 8 confirms that the sensitivity of the classifier (max inclination angle that is classified as expected) can be modified with the parameters. If desired, the classification band (shaded blue) can be quite selective, similar to what simulation results show. It can be thus concluded that the proposed framework has the potential to be used in practical applications.

In Fig. 8(a), a visible qualitative difference, when compared to Fig. 7, is that the uncertainty region is now asymmetric. Furthermore, as seen in Fig. 8(b), reducing the sensitivity results in a blue band that is not symmetric around $\beta = 0$. While the sensitivity curve should not be expected to be symmetric in general, the different shapes between the simulations and physical experiments are noticeably different considering they represent the same intentional impact scenario. While the reason for this qualitative difference has to be further investigated, we suspect that such a difference is due to both the lack of contact friction in the numerical simulation as well as a bias in the data-driven post-impact velocity prediction that was used to generate the post-impact reference motion. The latter in particular skews the definition of the post-impact error which is fed to the envelop filter as described in Sec. III.

V. Conclusion

In this paper, we proposed a collision monitoring framework able to discriminate between expected and unexpected post-impact behaviors using a prediction with a small computational

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

burden. The sensitivity of the classifier can be tuned, resulting in an adjustable range of impact scenarios that are always classified as expected. Scenarios deviating further from this range are eventually always classified as unexpected. The method has been developed and validated via numerical simulations and later tested on an experimental platform. Results are comparable, besides an asymmetry in the classification region which we ascribe to the current lack of suitable software to compute and properly validate the nominal inelastic impact map. In future work, software and validation of inelastic impact maps for simultaneous impacts should be developed along with a systematic approach to tune the classifier. The method should also be further evaluated experimentally also in different and more challenging desired impact scenarios. Additionally, expanding the classifier to allow for more fine-grained classification beyond expected/unexpected would allow for the design of a richer palette of reaction reflexes, each tailored to address a specific type of failure and making the execution more robust. Nevertheless, with the presented classification method, we claim that the first important step towards the realization of robust and useful monitoring for impact-aware robotics has been achieved.

REFERENCES

[1] S. S. M. Salehian and A. Billard, "A Dynamical-System-Based Approach for Controlling Robotic Manipulators During Noncontact/Contact Transitions," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2738–2745, 2018.

[2] S. Haddadin, A. Albu-Schaffer, A. De Luca, and G. Hirzinger, "Collision Detection and Reaction: A Contribution to Safe Physical Human-Robot Interaction," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 3356–3363.

[3] B. Vanderborght and others., "Variable impedance actuators: A review," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1601–1614, 2013.

[4] S. Monteleone, F. Negrello, M. G. Catalano, M. Garabini, and G. Grioli, "Damping in Compliant Actuation: A Review," *IEEE Robot. Autom. Mag.*, vol. 29, no. 3, pp. 47–66, 2022.

[5] I. Aouaj, V. Padois, and A. Saccon, "Predicting the Post-Impact Velocity of a Robotic Arm via Rigid Multibody Models: an Experimental Study," *2021 IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 2264–2271, 2021.

[6] Y. Wang, N. Dehio, and A. Kheddar, "Predicting Impact-Induced Joint Velocity Jumps on Kinematic-Controlled Manipulator," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6226–6233, 2022.

[7] S. Haddadin, A. D. Luca, and A. Albu-Schäffer, "Robot Collisions: A Survey on Detection, Isolation, and Identification," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292–1312, 2017.

[8] D. Popov, A. Klimchik, and N. Mavridis, "Collision detection, localization & classification for industrial robots with joint torque sensors," in *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2017, pp. 838–843.

[9] M. Karlsson, A. Robertsson, and R. Johansson, "Detection and Control of Contact Force Transients in Robotic Manipulation Without a Force Sensor," in *2018 IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2018, pp. 4091–4096.

[10] A. de Luca and R. Mattone, "Sensorless Robot Collision Detection and Hybrid Force/Motion Control," in *Proceedings of the 2005 IEEE Int. Conf. Robot. Autom.* IEEE, 2005, pp. 999–1004.

[11] J. Vorndamme, M. Schappler, and S. Haddadin, "Collision detection, isolation and identification for humanoids," in *2017 IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2017, pp. 4754–4761.

[12] H.-W. Park, S. Park, and S. Kim, "Variable-speed quadrupedal bounding using impulse planning: Untethered high-speed 3D Running of MIT Cheetah 2," in *2015 IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2015, pp. 5163–5170.

[13] C.-N. Cho, J.-H. Kim, Y.-L. Kim, J.-B. Song, and J.-H. Kyung, "Collision Detection Algorithm to Distinguish Between Intended Contact and Unexpected Collision," *Advanced Robotics*, vol. 26, no. 16, pp. 1–16, 2012.

[14] A. Kouris, F. Dimeas, and N. Aspragathos, "A Frequency Domain Approach for Contact Type Distinction in Human-Robot Collaboration," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 720–727, 2018.

[15] S. Golz, C. Osendorfer, and S. Haddadin, "Using tactile sensation for learning contact knowledge: Discriminate collision from physical interaction," in *2015 IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2015, pp. 3788–3794.

[16] N. Briquet-Kerestedjian, A. Wahrburg, M. Grossard, M. Makarov, and P. Rodriguez-Ayerbe, "Using Neural Networks for Classifying Human-Robot Contact Situations," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 3279–3285.

[17] S. Mikhel, D. Popov, S. Mamedov, and A. Klimchik, "Development of typical collision reactions in combination with algorithms for external impacts identification," in *IFAC-PapersOnLine*, vol. 52, no. 13. Elsevier, 2019, pp. 253–258.

[18] C. Glocker, *An Introduction to Impacts*. Springer, 2006, pp. 45–101.

[19] M. Rijnen, H. L. Chen, N. van de Wouw, A. Saccon, and H. Nijmeijer, "Sensitivity analysis for trajectories of nonsmooth mechanical systems with simultaneous impacts: a hybrid systems perspective," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 3623–3629.

[20] J. J. van Steen, N. van de Wouw, and A. Saccon, "Robot Control for Simultaneous Impact tasks via Quadratic Programming-based Reference Spreading," in *2022 American Control Conference (ACC)*. IEEE, 2022, pp. 3865–3872.

[21] ——, "Robot control for simultaneous impact tasks through time-invariant reference spreading," in *2023 American Control Conference (ACC)*. IEEE, 2023.

[22] A. De Luca, A. Albu-Schäffer, S. Haddadin, and G. Hirzinger, "Collision Detection and Safe Reaction with the DLR-III Lightweight Manipulator Arm," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 1623–1630.

[23] A. De Luca and R. Mattone, "Actuator failure detection and isolation using generalized momenta," in *2003 IEEE Int. Conf. Robot. Autom. (Cat. No.03CH37422)*, vol. 1. IEEE, 2003, pp. 634–639.

[24] A. Albu-Schäffer, C. Ott, and G. Hirzinger, "A Unified Passivity-based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots," *International Journal of Robotics Research*, vol. 26, no. 1, pp. 23–39, 2007.

[25] B. Brogliato, *Nonsmooth Mechanics: Models, Dynamics and Control*, 3rd ed. Springer, 2016.

[26] W. J. Stronge, *Impact Mechanics*, 2nd ed. Cambridge University Press, 2018.

[27] C. Ott, A. Albu-Schaffer, and G. Hirzinger, "Comparison of adaptive and nonadaptive tracking control laws for a flexible joint manipulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2. IEEE, 2002, pp. 2018–2024vol.2.

[28] Y. Hurmuzlu and D. B. Marghitu, "Rigid Body Collisions of Planar Kinematic Chains With Multiple Contact Points," *International Journal of Robotics Research*, vol. 13, no. 1, pp. 82–92, Feb. 1994.

[29] Y. Wang and M. T. Mason, "Two-Dimensional Rigid-Body Collisions With Friction," *Journal of Applied Mechanics*, vol. 59, no. 3, pp. 635–642, Sept. 1992.

[30] P. M. Wensing, M. Posa, Y. Hu, A. Escande, and A. Del Prete, "Optimization-Based Control for Dynamic Legged Robots," *under review (online draft available https://arxiv.org/abs/2211.11644)*, Nov. 2022.

[31] M. Rijnen, A. Saccon, and H. Nijmeijer, "Reference Spreading: Tracking Performance for Impact Trajectories of a 1DoF Setup," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 3, pp. 1124–1131, 2019.

[32] A. Saccon, N. van de Wouw, and H. Nijmeijer, "Sensitivity analysis of hybrid systems with state jumps with application to trajectory tracking," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 3065–3070.

[33] M. Rijnen, "Enabling motions with impacts in robotic and mechatronic systems," Ph.D. dissertation, TU/e: Department of Mechanical Engineering, 2018.

[34] A. Oppenheim and R. Schafer, *Discrete-Time Signal Processing*, 2nd ed. Prentice Hall, 1999.

[35] K. H. Hunt and F. R. E. Crossley, "Coefficient of Restitution Interpreted as Damping in Vibroimpact," *Journal of Applied Mechanics*, vol. 42, no. 2, pp. 440–445, 1975.

[36] A. S. Carvalho and J. M. Martins, "Exact restitution and generalizations for the Hunt-Crossley contact model," *Mechanism and Machine Theory*, vol. 139, pp. 174–194, 2019.

[37] S. Haddadin, S. Parusel, L. Johannsmeier, S. Golz, S. Gabl, F. Walch, M. Sabaghian, C. Jähne, L. Hausperger, and S. Haddadin, "The Franka Emika Robot: A Reference Platform for Robotics Research and Education," *IEEE Robot. Autom. Mag.*, vol. 29, no. 2, pp. 46–64, 2022.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.