IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Model-Adaptive High-Speed Collision Detection for Serial-Chain Robot Manipulators

Seyed Ali Baradaran Birjandi and Sami Haddadin

***Abstract*— In this paper, we introduce a novel regressor-based observer method to adapt an initially erroneous dynamics model of serial manipulators for improving collision detection sensitivity. Specifically, we assume that the robot joint velocity and acceleration can be accurately estimated via our previously introduced nonlinear estimator [1], [2] that fuses *Inertial measurement unit* (IMU) measurements with the robot proprioceptive sensing. Given the relatively high bandwidth of nowadays IMUs compared to a standard robot sensorization, the estimated kinematic joint variables support the prompt detection of unpredictable collisions. Compared to the state of the art, our algorithm notably improves collision detection accuracy and sensitivity, surpassing traditional methods such as the well established momentum based scheme. We support our claims and demonstrate the performance of our algorithm on a 7 degree of freedom (DoF) robot manipulator, both in simulation and experiment.**

I. Introduction and State of the Art

Safe human robot interaction is becoming more important as robotic systems pave their way into our daily lives. Detecting unpredictable collisions is an important phase in the collision event pipeline and, therefore, necessary for creating a safe framework for human-robot interaction [3]. Detected collisions can be considered as actuator faults in robotic systems [4], as it implicitly means that the actuator fails to behave as expected. The expected actuator behavior comes from an available dynamic system model. The most common state-of-the-art collision detection methods, such as the generalized momentum observer [4]–[8], the energy observer [3] or the disturbance observer from [9] compare some expected physical quantities such as the robot generalized momenta, energy or joint torque based on the available dynamics model to available measurements. Therefore, the robot dynamics models have direct influence on the collision detection precision.

These models, however, are never perfect because of time-varying coefficients such as motor coefficient and additional uncertainties such as load, backlash and friction [10], [11]. Specifically, both external forces applied to the robot links and modeling errors appear as the difference between the expected and measured physical quantities. Thus, the main challenge in today's collision detection approaches is to differentiate between modeling errors and externally applied forces. A common practice is to use thresholds, which is a simple strategy to avoid false collision detection. The main disadvantage of this approach is the decrease in the overall collision sensitivity and increase in detection delay. In other words, more severe modeling errors result in larger threshold selection and therefore, reduced sensitivity and responsiveness to collisions. In [12] a machine learning method is utilized to reduce model uncertainties in collision detection. As a result, smaller thresholds are required. The main limitation of the algorithm is that it can not be generally used for all robot manipulators in every situation. Time consuming data collection and training procedure is another drawback of such methods. Alternatively, time-variant thresholds are introduced in [13] in order to increase collision detection sensitivity. The algorithm contains several parameters, which need to be tuned. Since tuning the parameters requires large amount of collected data, it can be time consuming. Some of these parameters are related to the motor friction, which implicitly means that the parameters need to be constantly tuned. This, however, increase maintenance efforts.

Instead of using large thresholds, one can reduce model uncertainties in order to increase the collision detection precision. In [14] the physically-meaningful robot model is replaced with a neural network one, which shows less uncertainties. The algorithm is, however, dependent on the controller and therefore, large tracking errors are misclassified as faults or collisions. Moreover, due to the required heavy filtering, the method bandwidth is very limited.

Model adaptation, which was originally used to reduce trajectory tracking errors in robot manipulators [15], [16], is also a means to reduce the modeling errors or uncertainties. The robot dynamics model is adapted in [17] by estimating the model unknown parameters with assuming the availability of direct acceleration measurement. The monitoring signal is estimated based on the estimated robot manipulator generalized momentum. It is, however, well known that the momentum observers, despite clearly being the state-of-the-art scheme, has limited bandwidth [2].

In this paper, we introduce a novel method to adapt and improve the robot manipulator dynamics models in order to increase collision detection reliability, while maintaining high detection bandwidth and implementation simplicity. The underlying idea of the method lies on the fact that modeling errors and externally applied forces have different frequency characteristics. Robot modeling errors are mainly due to the simplification of dynamics expressions and/or the uncertainties in the inertial parameters such as link mass or inertia [18]. In this paper we account for the parametric uncertainties, which intrinsically have slow dynamics. For this, we introduce a Luenberger-like observer based on the robot regressor dynamics, which compares the estimated joint torques with the measured ones. First, in the transient time of the observer the parametric errors are estimated. Then, the estimated modeling errors are deducted from the monitoring signal, which is the difference between the model-adaptive computed torque and the measured one. As a result, high transient signals indicate potential dynamic impacts [19]. Furthermore, since different types of sensors are fused to generate high quality joint velocity and acceleration signals based on our most recent work [2], the proposed method shows high numerical stability, bandwidth and sensitivity compared to the state-of-the-art.

The authors are with the Chair of Robotics Science and Systems Intelligence, Munich School of Robotics and Machine Intelligence, Technical University Munich (TUM), Heßstr. 134, D-80797 Munich, GERMANY, firstname.lastname@tum.de

The remainder of the paper is organized as follows. In Section II, the problem of interest and the proposed solution are explained. In Section III, we introduce the required basics for solving the collision detection problem. In Section IV and V, we validate the theory in simulation and experiment with a 7-DoF robot, respectively. Finally, the paper concludes in Section VI.

II. Problem Statement and Contribution

Improving or adapting robot dynamics models for collision detection purposes has essentially been of lesser focus so far. Typically, the expected torque is computed based on the available robot dynamics model, which is essential to the quality of the estimated external torque. Since no robot model is perfect, the signal which carries the collision data (also called the remainder signal) is erroneous in nature. Moreover, joint velocity information is generally required in existing collision detection algorithms. As there exist no accurate measurement device for measuring joint velocity in manipulators, it is normally computed by numerical differentiation of the joint position. This causes additional noise to the remainder signal due to the quantized nature of the joint position measurement. Therefore, the traditional collision detection methods normally suffer from poor *signal to noise ratio* (SNR). In order to solve this issue, thresholds are traditionally used to distinguish between errors and real collisions in the remainder signal. This, however, will reduce the system sensitivity to the applied external forces. Moreover, as proprioceptive measurements have limited bandwidth, the state-of-the-art collision detection algorithms have limited bandwidth as well.

In our recent work [2], we fused IMUs with proprioceptive sensing to increase collision detection bandwidth. Furthermore, we introduced an observer which generates joint velocity and acceleration estimates that are more accurate than standard numerical differentiation and have large bandwidth. This increases the SNR and results in smaller required thresholds to detect collisions. In this paper, we intend to further improve the remainder signal by adapting the robot model online. Given that robot modeling errors are characterized by slow dynamics, they are intrinsically different from external forces. By exploiting this idea, we designed a Luenberger-like observer, which takes the difference between the expected and measured torques and separates its slow- and fast-dynamics components. The fast-dynamics components are then reported as collision. By doing so, the remainder signal becomes cleaner and consequently its SNR improves. As a result, smaller thresholds are required and the collision detection sensitivity increases. Furthermore, since the IMU data is fused with the available robot measurements, the bandwidth is higher than the traditional collision detection methods. Figure 1 summarizes the improvements and advantages of our proposed scheme in comparison to state-of-the-art collision detection algorithms.

III. Model-Adaptive Collision Detection

Let us assume the standard rigid body dynamics model of a robot with n joints as

$$\hat{\boldsymbol{M}}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \hat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{g}}(\boldsymbol{q}) = \boldsymbol{\tau}_{\mathrm{m}} - \hat{\boldsymbol{\tau}}_{\mathrm{f}} + \boldsymbol{\tau}_{\mathrm{ext}} - \boldsymbol{\tau}_{\mathrm{e}}, \quad (1)$$

where $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}} \in \mathbb{R}^n$ denote the link side joint position, velocity and acceleration, $\hat{\boldsymbol{M}}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ the available estimation of symmetric and positive definite inertia matrix, $\hat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} \in \mathbb{R}^n$ the estimation of centripetal and Coriolis vector, $\hat{\boldsymbol{g}}(\boldsymbol{q}) \in \mathbb{R}^n$ the estimation of gravity vector, $\boldsymbol{\tau}_{\mathrm{ext}} \in \mathbb{R}^n$ the external torque caused by unforeseen collisions, $\boldsymbol{\tau}_{\mathrm{m}} \in \mathbb{R}^n$ the active motor torque, $\hat{\boldsymbol{\tau}}_{\mathrm{f}} \in \mathbb{R}^n$ the estimation of friction torque and $\boldsymbol{\tau}_{\mathrm{e}} \in \mathbb{R}^n$ modeling errors. Please note that since the exact dynamics model ($\boldsymbol{M}(\cdot)$, $\boldsymbol{C}(\cdot)$, $\boldsymbol{g}(\cdot)$ and $\boldsymbol{\tau}_{\mathrm{f}}$) is not known in reality, the modeling error $\boldsymbol{\tau}_{\mathrm{e}}$ appears in the approximated dynamics model.

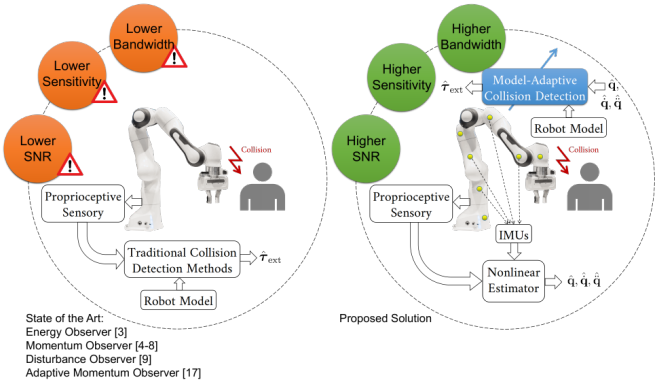


Fig. 1: Schematic comparison of the state-of-the-art with the proposed solution

In this section we assume that no collision occurs during model adaptation, i.e. $\boldsymbol{\tau}_{\mathrm{ext}} = \boldsymbol{0}$. Thus, system (1) can be reformulated to *linear in parameter* (LIP) form as [20]

$$\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\hat{\boldsymbol{\theta}} = \boldsymbol{\tau}_{\mathrm{m}} - \boldsymbol{\tau}_{\mathrm{e}}, \quad (2)$$

where $\boldsymbol{Y}(\cdot) \in \mathbb{R}^{n\times m}$ is called regressor matrix and contains the robot kinematics information. $\hat{\boldsymbol{\theta}} \in \mathbb{R}^m$ is the erroneous robot inertial parameters vector which consists of each link mass $\boldsymbol{m} \in \mathbb{R}^n$, center of mass $\boldsymbol{c} \in \mathbb{R}^{3n}$, elements of moment of inertia tensor $\boldsymbol{I} \in \mathbb{R}^{6n}$ and the coefficients of friction. $\hat{\boldsymbol{\theta}}$ comes with $\hat{\cdot}$ to express that this parameter is not exact, due to the modeling errors in (1). Therefore, if $\boldsymbol{\theta}$ was available, $\boldsymbol{\tau}_{\mathrm{e}} = \boldsymbol{0}$ would hold, i.e.

$$\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{\theta} = \boldsymbol{\tau}_{\mathrm{m}}. \quad (3)$$

Depending on the choice of linear joint friction model, which may include either or both of viscous and Coulomb friction, the number of corresponding coefficients in the inertial parameters vector varies. However, in simulation and experiment the friction effects are neglected for the sake of clarity. Moreover, following the procedure proposed in [20], there will normally appear linearly-dependent columns in the regressor matrix. There exist different approaches such as the ones in [21], [22] to remove these columns. Accordingly, the elements of $\boldsymbol{\theta}$ will turn into linear combinations of the inertial parameters mentioned above. Therefore, the length and structure of $\boldsymbol{\theta}$ will be determined by the structure of the regressor matrix.

Robot geometry information is normally provided with high accuracy by the manufacturers. Moreover, various algorithms for robot geometry calibration are known. Thus, we assume that the regressor matrix is known. The modeling errors are considered as parametric uncertainties in the inertial parameters, i.e

$$\boldsymbol{\tau}_{\mathrm{e}} = \boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\delta\boldsymbol{\theta}, \quad (4)$$

or

$$\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})(\hat{\boldsymbol{\theta}} + \delta\boldsymbol{\theta}) = \boldsymbol{\tau}_{\mathrm{m}}, \quad (5)$$

with $\delta\boldsymbol{\theta} \in \mathbb{R}^m$ being the *constant* (or slowly varying) and *bounded* inertial parameter error. Please note that the

availability of $\hat{\boldsymbol{\theta}}$ itself is not of this work's concern. In general, it is assumed that some non-accurate robot dynamics model with the unknown bounded error $\delta\boldsymbol{\theta}$ is available. From a practical standpoint, a model can be of the form (1), (2) (with unknown $\boldsymbol{\tau}_\mathrm{e}$ and $\boldsymbol{\tau}_\mathrm{ext}$) or even numerical values of $\hat{\boldsymbol{M}}(\cdot)$, $\hat{\boldsymbol{C}}(\cdot)$ and $\hat{\boldsymbol{g}}(\cdot)$ provided by the robot interface.

a) Observer Design and Model Adaptation: In order to estimate the robot modeling error $\boldsymbol{\tau}_\mathrm{e}$, we propose a Luenberger-like observer. Given that the inertial parameters deviation $\delta\boldsymbol{\theta}$ is considered to be constant, the dynamics of the modeling error is

$$\dot{\boldsymbol{\tau}}_\mathrm{e} = \dot{\boldsymbol{Y}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\delta\boldsymbol{\theta}, \tag{6}$$

where $\dot{\boldsymbol{Y}}(\cdot)$ is the derivative of the regressor with respect to time. Given that one has smooth joint velocity and acceleration such as [1], [2], $\dot{\boldsymbol{Y}}(\cdot)$ can be calculated by numerical differentiation. State space representation of the dynamic of the modeling error is then given by

$$\dot{\boldsymbol{Y}}(\cdot)\delta\boldsymbol{\theta} = \dot{\boldsymbol{Y}}(\cdot)\mathcal{I}_\mathrm{n}\delta\boldsymbol{\theta} = \dot{\boldsymbol{Y}}(\cdot)\boldsymbol{Y}^{+}(\cdot)\boldsymbol{Y}(\cdot)\delta\boldsymbol{\theta} \tag{7}$$

or more concretely

$$\dot{\boldsymbol{\tau}}_\mathrm{e} = \dot{\boldsymbol{Y}}(\cdot)\boldsymbol{Y}^{+}(\cdot)\boldsymbol{\tau}_\mathrm{e}, \tag{8}$$

with $\mathcal{I}_\mathrm{n}$ being the identity matrix of size n and $\boldsymbol{Y}^{+}(\cdot)$ the pseudo-inverse of the regressor matrix. As $\dot{\boldsymbol{Y}}(\cdot)$ and $\boldsymbol{Y}^{+}(\cdot)$ are both a posteriori known values, (8) is used to update the state $\boldsymbol{\tau}_\mathrm{e}$. If we denote the estimated inertial parameter error as $\delta\hat{\boldsymbol{\theta}} \in \mathbb{R}^m$, the observer dynamics are given by

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{Y}(\cdot)\delta\hat{\boldsymbol{\theta}}\right) = \dot{\boldsymbol{Y}}(\cdot)\delta\hat{\boldsymbol{\theta}} + \boldsymbol{Y}(\cdot)\dot{\delta\hat{\boldsymbol{\theta}}} \tag{9}$$

One can define the adaptation law for $\delta\hat{\boldsymbol{\theta}}$ as

$$\begin{aligned}\dot{\delta\hat{\boldsymbol{\theta}}} &= \boldsymbol{Y}^{+}(\cdot)\boldsymbol{K}\boldsymbol{Y}(\cdot)\left(\delta\boldsymbol{\theta} - \delta\hat{\boldsymbol{\theta}}\right)\\ \boldsymbol{Y}(\cdot)\dot{\delta\hat{\boldsymbol{\theta}}} &= \boldsymbol{K}\left(\boldsymbol{Y}(\cdot)\delta\boldsymbol{\theta} - \boldsymbol{Y}(\cdot)\delta\hat{\boldsymbol{\theta}}\right),\end{aligned} \tag{10}$$

where $\boldsymbol{K} = \mathrm{diag}(K_1 \cdots K_\mathrm{n}) \in \mathbb{R}^{n\times n}$ is a positive diagonal gain matrix. By replacing $\boldsymbol{Y}(\cdot)\delta\boldsymbol{\theta}$ in (10) with (5), we get

$$\boldsymbol{Y}(\cdot)\dot{\delta\hat{\boldsymbol{\theta}}} = \boldsymbol{K}\left(\boldsymbol{\tau}_\mathrm{m} - \boldsymbol{Y}(\cdot)(\hat{\boldsymbol{\theta}} + \delta\hat{\boldsymbol{\theta}})\right). \tag{11}$$

The commanded torque $\boldsymbol{\tau}_\mathrm{m}$ is assumed to be measurable. Therefore, in the correction step the estimated $\hat{\boldsymbol{\tau}}_\mathrm{m} = \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})(\hat{\boldsymbol{\theta}}+\delta\hat{\boldsymbol{\theta}})$ is compared with the measured one. Now, by reformulating (9), the observer dynamics are

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{Y}(\cdot)\delta\hat{\boldsymbol{\theta}}\right) = \dot{\boldsymbol{Y}}(\cdot)\delta\hat{\boldsymbol{\theta}} + \boldsymbol{K}\left(\boldsymbol{\tau}_\mathrm{m} - \boldsymbol{Y}(\cdot)(\hat{\boldsymbol{\theta}} + \delta\hat{\boldsymbol{\theta}})\right). \tag{12}$$

Figure 2 shows the block diagram of this observer. Since the joint variables $\hat{\boldsymbol{q}}, \hat{\dot{\boldsymbol{q}}}$ and $\hat{\ddot{\boldsymbol{q}}}$ are initially estimated by a nonlinear estimator, the regressor matrix $\hat{\boldsymbol{Y}}(\cdot)$, its pseudo inverse $\hat{\boldsymbol{Y}}^{+}(\cdot)$ and derivative $\dot{\hat{\boldsymbol{Y}}}(\cdot)$ as well as the available erroneous robot model $\hat{\boldsymbol{Y}}(\cdot)\hat{\boldsymbol{\theta}}$ are denoted by $\hat{\cdot}$ symbol. Furthermore, $\boldsymbol{r}$ is the remainder signal, whose characteristics will be discussed later.

Take $\boldsymbol{e}_\mathrm{O} = \boldsymbol{Y}(\cdot)(\delta\boldsymbol{\theta} - \delta\hat{\boldsymbol{\theta}})$ as the observer error. According to (8) and the adaptation law (10), the error dynamics are

$$\dot{\boldsymbol{e}}_\mathrm{O} = \left(\dot{\boldsymbol{Y}}(\cdot)\boldsymbol{Y}^{+}(\cdot) - \boldsymbol{K}\right)\boldsymbol{e}_\mathrm{O}. \tag{13}$$

In the absence of external forces, the regressor matrix and its derivative are bounded for any given trajectory [23]. Moreover, the regressor matrix is assumed to have full rank along the given trajectories and therefore, its pseudo-inverse is non-singular. Thus, the observer converges with large-enough positive definite gain matrix $\boldsymbol{K}$. In turn, it also determines how fast the modeling error converges.

b) Observability: Since we are dealing with a nonlinear system, we need to check the rank condition of the nonlinear observability matrix

$$\mathcal{O} = \begin{pmatrix} \frac{\partial}{\partial\boldsymbol{\tau}_\mathrm{e}}\boldsymbol{g}(\cdot) \\ \frac{\partial}{\partial\boldsymbol{\tau}_\mathrm{e}}(\boldsymbol{L}_\mathrm{f}\boldsymbol{g}(\cdot)) \\ \frac{\partial}{\partial\boldsymbol{\tau}_\mathrm{e}}(\boldsymbol{L}_\mathrm{f}^2\boldsymbol{g}(\cdot)) \\ \vdots \\ \frac{\partial}{\partial\boldsymbol{\tau}_\mathrm{e}}(\boldsymbol{L}_\mathrm{f}^{n-1}\boldsymbol{g}(\cdot)) \end{pmatrix}, \tag{14}$$

where $\boldsymbol{L}_\mathrm{f}\boldsymbol{g}(\cdot)$ is the Lie derivative of $\boldsymbol{g}(\cdot)$ with respect to $\boldsymbol{f}(\cdot)$ [24]. The dynamics of $\dot{\boldsymbol{\tau}}_\mathrm{e} = \boldsymbol{f}(\boldsymbol{\tau}_\mathrm{e}, \boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ are given by (8) and the measurement function $\boldsymbol{g}(\boldsymbol{\tau}_\mathrm{e}, \boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ by (5). Since $\frac{\partial\boldsymbol{g}(\cdot)}{\partial\boldsymbol{\tau}_\mathrm{e}} = \mathcal{I}_\mathrm{n}$ has full rank, the system is *globally* observable with respect to $\boldsymbol{\tau}_\mathrm{e}$. Therefore, as long as $\boldsymbol{\tau}_\mathrm{e}$ (or equivalently $\delta\boldsymbol{\theta}$) is bounded, one will be able to estimate it.

c) Regressor Computation: So far, it is assumed that the correct regressor matrix, its derivative and pseudo-inverse are always available. The regressor is a complex function of the robot kinematics, though. In other words, the regressor can only be computed when the joint variables $\boldsymbol{q}, \dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ are known. In our recent work [1] we introduced a method to estimate joint velocity and acceleration by fusing the link-side position measurements $\boldsymbol{q}$ with the Cartesian acceleration $\ddot{\boldsymbol{X}}$ at the IMU location on the link. The link Cartesian acceleration is measured by an accelerometer installed on each robot manipulator link. IMUs are electrical devices which are typically equipped with a 3-axis accelerometer and a 3-axis gyroscope. Replacing the accelerometers with IMUs on each robot link, we are able to fuse also the gyroscope data in the estimator. This increase the accuracy and bandwidth of the estimated joint velocity and acceleration [2]. Once the highly-accurate estimated joint variables are at hand, one is able to obtain the regressor matrix and its components (e.g. $\dot{\boldsymbol{Y}}(\cdot)$ and $\boldsymbol{Y}^{+}(\cdot)$).

d) External Torque Detection: So far the external torque was absent in the calculations. However, detection of the externally applied torques on the robot links at the observer steady state via the remainder signal $\boldsymbol{r}$ is elaborated here. Given the true inertial parameters $(\hat{\boldsymbol{\theta}}+\delta\boldsymbol{\theta})$, the external torque is

$$\boldsymbol{\tau}_\mathrm{ext} = \boldsymbol{Y}(\cdot)(\hat{\boldsymbol{\theta}} + \delta\boldsymbol{\theta}) - \boldsymbol{\tau}_\mathrm{m}. \tag{15}$$

with the remainder being defined as

$$\boldsymbol{r} = \boldsymbol{Y}(\cdot)(\hat{\boldsymbol{\theta}} + \delta\hat{\boldsymbol{\theta}}) - \boldsymbol{\tau}_\mathrm{m}. \tag{16}$$

Deducting (15) from (16), we get

$$\boldsymbol{r} - \boldsymbol{\tau}_\mathrm{ext} = \boldsymbol{Y}(\cdot)(\delta\hat{\boldsymbol{\theta}} - \delta\boldsymbol{\theta}). \tag{17}$$

Differentiating both sides w.r.t time leads to

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{r} - \boldsymbol{\tau}_\mathrm{ext}) &= \frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{Y}(\cdot)(\delta\hat{\boldsymbol{\theta}} - \delta\boldsymbol{\theta}))\\ &= \dot{\boldsymbol{Y}}(\cdot)(\delta\hat{\boldsymbol{\theta}} - \delta\boldsymbol{\theta}) + \boldsymbol{Y}(\cdot)\dot{\delta\hat{\boldsymbol{\theta}}}\\ &= -\dot{\boldsymbol{Y}}(\cdot)\boldsymbol{Y}^{+}(\cdot)\boldsymbol{Y}(\cdot)(\delta\boldsymbol{\theta} - \delta\hat{\boldsymbol{\theta}}) + \boldsymbol{Y}(\cdot)\dot{\delta\hat{\boldsymbol{\theta}}}.\end{aligned} \tag{18}$$

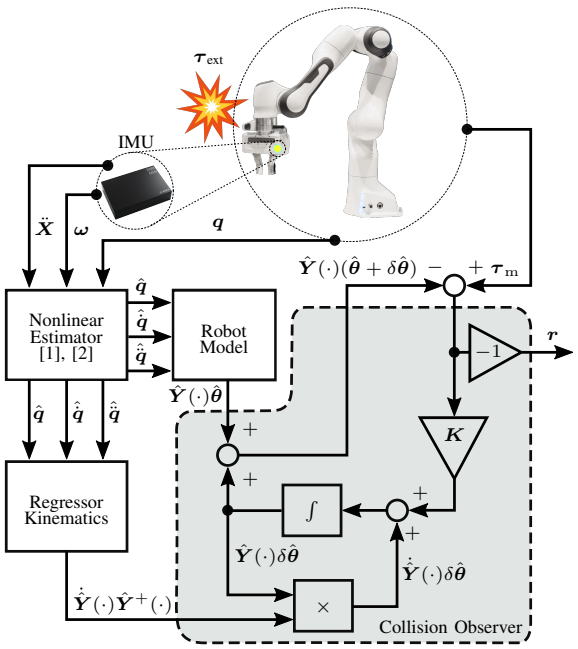


Fig. 2: Model-adaptive collision detection block diagram

As mentioned earlier, $\delta\boldsymbol{\theta}$ is assumed to be constant. A moment before the impact and at its steady-state, the observer error $\boldsymbol{Y}(\cdot)(\delta\boldsymbol{\theta}-\delta\hat{\boldsymbol{\theta}})$ is at minimum. Given that the impact time is practically very small [19], $\boldsymbol{Y}(\cdot)(\delta\boldsymbol{\theta}-\delta\hat{\boldsymbol{\theta}})\approx 0$ can also be assumed during the collision. Please note that even though the observer error is considered to be small during the impact, with large values of $\boldsymbol{K}$, $\boldsymbol{Y}(\cdot)\delta\dot{\hat{\boldsymbol{\theta}}}\neq 0$ is not negligible in (18). Furthermore, $\boldsymbol{Y}(\cdot)\delta\dot{\hat{\boldsymbol{\theta}}}$ is given by the adaptation law (11). With the help of (11) and (16), (18) simplifies into the linear form

$$\boldsymbol{r}(t)=-\boldsymbol{K}\int_{\tau=0}^{t}\boldsymbol{r}(\tau)\mathrm{d}\tau+\boldsymbol{\tau}_{\mathrm{ext}}(t). \tag{19}$$

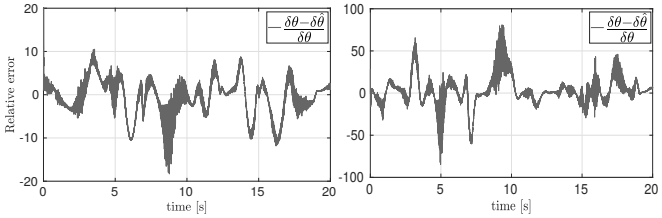
As a result, the transfer function from the actual external torque $\tau_{\mathrm{ext,i}}$ of the i-th link to the signal r_{i} is of the form

$$\frac{r_{\mathrm{i}}}{\tau_{\mathrm{ext,i}}}=\frac{s}{s+K_{\mathrm{i}}}. \tag{20}$$

This shows that the observer acts in first approximately as a high-pass filter on the external torque with the cutoff frequency K_{i}, during the impact incident.

IV. Simulation Experiments

In this section the collision observer is simulated for a 7-DoF robot manipulator. The dynamic model of the manipulator is provided by Gaz et al. [25] and accessible from http://diag.uniroma1.it/~gaz/panda2019.html. One IMU (BMI055 [26]), with all parasitic effects such as noise, bias, quantization and sensitivity and bias change due to temperature, is assumed to be mounted on each robot link. An internal low-pass filter with 1 kHz cutoff frequency truncates the sensor noise. Moreovers, encoders are simulated as 12-bit sensors with 3 LSB affected by noise. For the sake of simplicity, we do not model the entire joint dynamics and assume that the link-side position and joint torque are measurable at 100 Hz -3 dB bandwidth, in this section. We initially investigate the performance of the inertial parameter error estimation ($\delta\hat{\boldsymbol{\theta}}\in\mathbb{R}^{43}$). Given the observer state $\hat{\boldsymbol{\tau}}_{\mathrm{e}}$, the



(a) Model adaptive collision detection (b) Model adaptive control

Fig. 3: The largest relative error among estimated inertial parameters with model-adaptive collision detection (left) and model-adaptive control (right)

estimated inertial parameter error can be computed via

$$\delta\hat{\boldsymbol{\theta}}=\boldsymbol{Y}^{+}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\hat{\boldsymbol{\tau}}_{\mathrm{e}}, \tag{21}$$

where $\boldsymbol{Y}^{+}(\cdot)$ is the exact regressor pseudo-inverse. The inertial parameter error is chosen to be

$$\delta\boldsymbol{\theta}=\mathcal{U}(-0.1\boldsymbol{\theta},0.1\boldsymbol{\theta}), \tag{22}$$

where $\mathcal{U}$ denotes uniform distribution. In other words, the inertial parameters error is uniformly distributed between $\pm 10\%$ of the true values. The minimal inertial parameters vector $\boldsymbol{\theta}$ (and consequently $\delta\boldsymbol{\theta}$) is assumed to be fully excitable with our excitation trajectory. This Fourier-like excitation trajectory respects the joint torque, velocity as well as position limits of the considered system and lasts for 20 seconds. The observer gain is set to $\boldsymbol{K}=100\mathcal{I}_7$. We compare our method performance in estimating the true inertial parameters with a somewhat related model adaptive control system introduced in [27]. Similar to our method, the algorithm is regressor-based. However, the joint acceleration measurement is not required. We picked this sliding mode adaptive controller mainly due to its implementation simplicity and promising stability properties as well as its popularity. The adaptation law for the available erroneous inertial parameters is given by

$$\begin{aligned}\dot{\hat{\boldsymbol{\theta}}}&=-\boldsymbol{\Gamma}^{-1}\boldsymbol{Y}^{T}(\boldsymbol{q},\Delta\boldsymbol{q}/\Delta t,\ddot{\boldsymbol{q}}_{\mathrm{r}})\boldsymbol{s},\\ \ddot{\boldsymbol{q}}_{\mathrm{r}}&=\ddot{\boldsymbol{q}}_{\mathrm{d}}-\boldsymbol{\Lambda}(\Delta\boldsymbol{q}/\Delta t-\dot{\boldsymbol{q}}_{\mathrm{d}}),\\ \boldsymbol{s}&=\Delta\boldsymbol{q}/\Delta t-\dot{\boldsymbol{q}}_{\mathrm{d}}+\boldsymbol{\Lambda}(\boldsymbol{q}-\boldsymbol{q}_{\mathrm{d}}),\end{aligned} \tag{23}$$

where $\boldsymbol{\Gamma}^{-1}\in\mathbb{R}^{m\times m}$ is a symmetric positive definite matrix which determines the adaptation rate. Also, the eigenvalues of the $\boldsymbol{\Lambda}\in\mathbb{R}^{n\times n}$ are strictly on the right-half complex plane. Furthermore, $\boldsymbol{q}_{\mathrm{d}},\dot{\boldsymbol{q}}_{\mathrm{d}},\ddot{\boldsymbol{q}}_{\mathrm{d}}\in\mathbb{R}^{n}$ denote the desired trajectory and $\Delta\boldsymbol{q}/\Delta t$ is the numerical differentiation of the measured link-side position $\boldsymbol{q}$.

Figure 3 depicts the largest estimated inertial parameters vector relative error, $\max\left(\frac{\delta\boldsymbol{\theta}-\delta\hat{\boldsymbol{\theta}}}{\delta\boldsymbol{\theta}}\right)$ estimated by both methods. It shows that even with a rich excitation trajectory, some of the inertial parameters do not converge to their true values. As can be seen, our solution has in general smaller relative error, as it benefits from more accurate joint velocity and acceleration signals. Furthermore, Fig. 4 depicts the estimated external torque error of the second link ($\tau_{\mathrm{ext},2}-\hat{\tau}_{\mathrm{ext},2}$) in the absence of collision. This value (which is the largest one among the seven links) is relatively close to negligible, considering the large maximal torque in the second joint. Thus, it becomes clear that even though not all inertial parameter errors are fully identified, the observer satisfactorily rejects the modeling errors.

At this point, one might argue that instead of using the observer, which requires computation of the regressor matrix, we could simply use a high-pass filter to eliminate the mod-

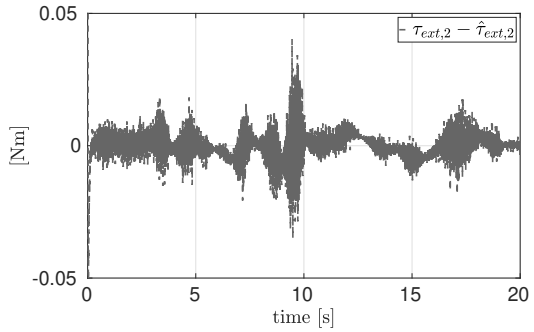


Fig. 4: The largest estimated external torque error (2nd link) with model-adaptive collision detection

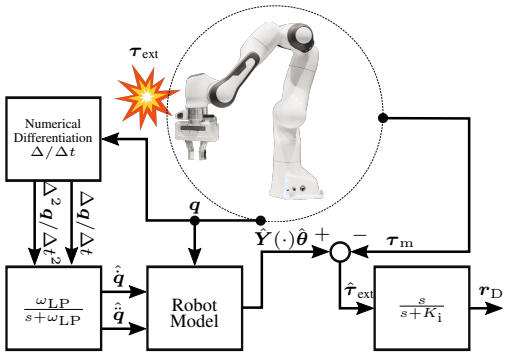


Fig. 5: Collision detection with direct method and high-pass filtering block diagram

eling errors from the remainder signal. As a result, no IMU needs to be installed and the computational costs decrease. However, the main advantage of our method lies in the higher bandwidth, brought by the IMUs and consequently, the estimated joint variables. Therefore, the next simulations aim at investigating the effect of the observer gain on the external torque estimation at different frequencies with and without IMUs. The modeling error (22) as well as the motion trajectory remains the same as in the previous simulation. In order to further simplify the results, the external force is assumed to be applied to the first link only. The external torque is modeled as a step that is filtered via a first-order low-pass filter with cutoff frequency f.

Figure 5 depicts the system block diagram without IMU. ω_{LP} refers to the low-pass filter cutoff frequency and is chosen to be $f_{\text{LP}} = 100 = \omega_{\text{LP}}/2\pi$ Hz. Here, the signs are flipped, when comparing the measured and expected torques to generate the signal $\hat{\tau}_{\text{ext}}$. Therefore, no negation will be later needed. Please also note that without the high-pass filter, the system reduces to the direct method for collision detection with numerical differentiation, fully covered in [2].

Figure 6 (a-d) depict the signal $\boldsymbol{r}_D$. Due to limited measurement bandwidth, the high-frequency components of the external force are already eliminated from $\boldsymbol{r}_D$. As a result, the high-pass filter channels mainly noise. According to Fig. 6 (c) and (d), which are more similar to real impacts, the method works for very limited gains around $K \approx 1$ rad. Larger gains reduce the amplitude of $\boldsymbol{r}_D$, such that the impact is not distinguishable from the severe noise. Smaller gains, on the other hand, do not effectively reject modeling errors. Since only narrow range of K can deliver somewhat useful results, the setup is essentially impractical. In conclusion, high-pass filters may reduce the parametric modeling errors however, fast and proper detection of impacts is neither guaranteed nor robust in this setup.

Figure 6 (e-h) show the estimated external torques with different observer gains K for the model-adaptive collision detection. With high K, slow dynamics external forces are more difficult to detect. As mentioned before, one of the main advantages of the proposed solution is its large estimation bandwidth, which allows high-pass filtering actions on the remainder signal. We further discuss the real-time computation aspects of the algorithm in Sec. V. Please note that, the purpose of the method is not to estimate the exact external torque/force applied to the robot, but to quickly *detect* and report failures in the form of impacts. Thus, detecting only parts of the impact spectrum suffices the desired specification. As a result, K can always be chosen as $K >> 10$. Thus, given the observer error dynamics (13), the observer error ($\boldsymbol{e}_{\text{O}}$) reaches 10% of the initial error ($\boldsymbol{e}_{\text{O},0}$) in approximately less than a second.

V. Experimental Evaluation

The method introduced in Sec. III, is now examined with a 7-DoF robot manipulator. The experiment is carried out in two different setups namely, offline and online modes. In the offline mode, based on the real robot trajectory, one IMU per robot link is simulated. In the online case on the other hand, one IMU is installed at the robot end-effector and the collision is detected under real-time conditions.

a) Traditional Collision Detection Methods: Before proceeding to the experiments, three traditional collision detection methods namely, the generalized momentum observer, observer-extended direct method and the adaptive momentum observer are briefly introduced. Subsequently, the performance of these methods is compared with the new technique.

In the momentum observer, the robot manipulator generalized momentum $\boldsymbol{p} = \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}$, with the dynamics [3]

$$\begin{aligned}\dot{\hat{\boldsymbol{p}}} &= \boldsymbol{\tau}_{\text{m}} - \boldsymbol{\tau}_{\text{f}} - \hat{\boldsymbol{\beta}}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{r}_{\text{M}} \\ \boldsymbol{\beta} &= \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \dot{\boldsymbol{M}}(\boldsymbol{q})\dot{\boldsymbol{q}} \\ \dot{\boldsymbol{r}}_{\text{M}} &= \boldsymbol{K}_{\text{O}}(\dot{\boldsymbol{p}} - \dot{\hat{\boldsymbol{p}}}),\end{aligned} \tag{24}$$

is observed with diagonal observer gain $\boldsymbol{K}_{\text{O}} \in \mathbb{R}^{n\times n}$. Please note that $\boldsymbol{\beta}$ is only an auxiliary variable. The solution of the remainder signal $\boldsymbol{r}_{\text{M}}$ and the output of the observer are given by

$$\boldsymbol{r}_{\text{M}}(t) = \boldsymbol{K}_{\text{O}}\big(\boldsymbol{p}(t) - \int_{\tau=0}^{t} \dot{\hat{\boldsymbol{p}}}(\tau)\text{d}\tau - \boldsymbol{p}(0)\big), \tag{25}$$

which implies

$$\dot{\boldsymbol{r}}_{\text{M}} = \boldsymbol{K}_{\text{O}}(\boldsymbol{\tau}_{\text{ext}} - \boldsymbol{r}_{\text{M}}). \tag{26}$$

In other words, the momentum observer leads to a first-order low-pass filter estimation of the applied external torques.

The observer-extended direct method on the other hand, solves (1) algebraically, given the estimated joint velocity and acceleration provided by the nonlinear estimator. Because of rich sensory fusion, the method shows better bandwidth and accuracy compared to the momentum observer [2]. Both methods, however, neglect modeling errors and assume a perfect robot dynamics model.

In the last traditional collision detection algorithm introduced in [17] however, the robot dynamic model is adapted using gradient correction. The technique is basically a general momenta observer (24) combined with an adaptive finite

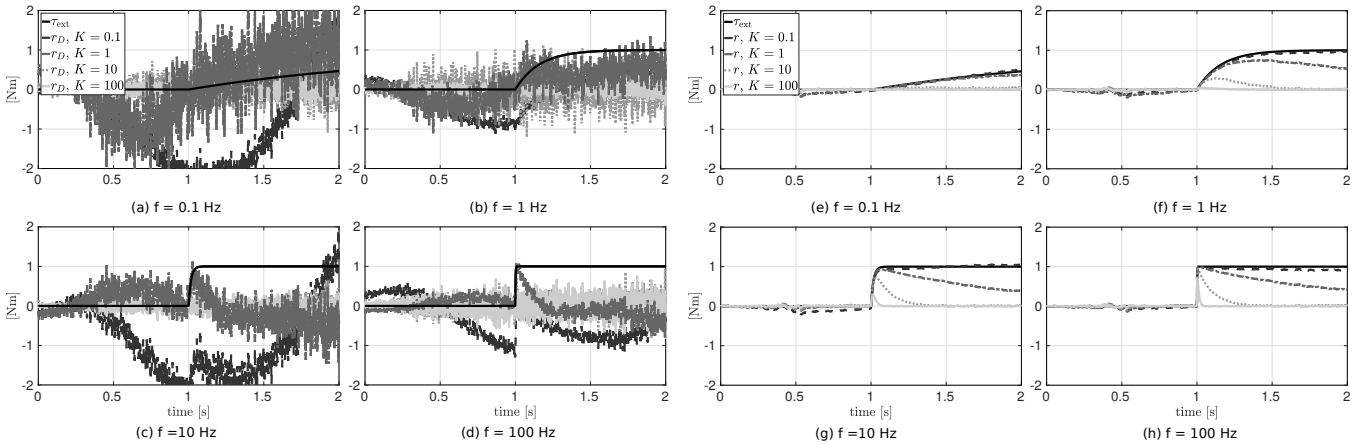


Fig. 6: Collision detection with direct method with high-pass filtering (a-d) compared to model-adaptive collision detection (e-f) (our proposed solution) performance against different external torque reference signals. Both setups contain the same modeling error, measurement bandwidth and trajectory.

impulse response (*FIR*) filter for reducing the modeling errors. Along with the model adaptation, some dynamic threshold for separating the false alarms from collisions are also estimated. The filtered residual $\boldsymbol{r}_{\mathrm{F}} \in \mathbb{R}^n$ is given by

$$\boldsymbol{r}_{\mathrm{F}} = \boldsymbol{H}\boldsymbol{r}_{\mathrm{M}}, \tag{27}$$

where $\boldsymbol{H} \in \mathbb{R}^{n\times n}$ is the adaptive FIR filter and $\boldsymbol{r}_{\mathrm{M}}$ the remainder obtained by momentum observer (24). For detailed explanations on the adaptation laws and threshold estimation, please refer to [17]. For the sake of simplicity, the technique is referred to as *adaptive momentum observer*, in this work.

Offline mode: Complex motion trajectories may introduce highly nonlinear modeling errors that might be more challenging for the observer to estimate. Here, we investigate the modeling error rejection as well as the detection sensitivity of our method when the collision happens while executing a Cartesian motion.

To the best of the authors knowledge, there exist no robot manipulator, which is fully equipped with IMUs as of today. In the offline mode however, the robot motion is recorded and one IMU per link is simulated subsequently. Specifically, the IMUs are simulated based on the BMI055 [26] datasheet, indicating all relevant parasitic effects. Accordingly, noise power spectral density is 150 $\mu\mathrm{g}/\sqrt{\mathrm{Hz}}$ for the accelerometer and $0.014^\circ/\mathrm{s}/\sqrt{\mathrm{Hz}}$ for the gyroscope. The full scale is chosen 1 g for the accelerometer and $1000^\circ/\mathrm{s}$ for the gyroscope. Both sensors report the measurements in 12 bits. The cutoff frequency of the internal second-order low-pass filters is 250 Hz for the accelerometer and 64 Hz for the gyroscope. The advantage of the offline-mode experiment is that all joints can be actuated during the experiment, without needing to install real IMUs on all links. The robot moves linearly in Cartesian space from the start configuration and collides perpendicularly with a force plate in this experiment (Fig. 7). The force plate contains high-bandwidth high-accuracy force sensors, which return the contact wrench at the contact position. The trapezoidal Cartesian joint trajectory has three cruise velocities, namely low (0.2 m/s) and normal (0.5 m/s) and high (2 m/s) velocities. Since the external torques can be observed at all joints and for the sake of clarity, the external force at the contact is compared via

$$\boldsymbol{f}_{\mathrm{c}} = \boldsymbol{J}^{-T}\boldsymbol{r}, \tag{28}$$

with $\boldsymbol{f}_{\mathrm{c}} \in \mathbb{R}^6$ being the exerted wrench at the contact point, $\boldsymbol{J} \in \mathbb{R}^{6\times 7}$ and $\boldsymbol{r} \in \mathbb{R}^{7\times 1}$ the remainder signal computed by any suitable algorithm. Therefore, the contribution of the detected external torque at all joints can be summed up and compared. Furthermore, since the impact direction is nearly perpendicular to the force plate, we only report and compare the third element of $\boldsymbol{f}_{\mathrm{c}}$ ($f_{\mathrm{z}} = [0\,0\,1\,0\,0\,0]\boldsymbol{f}_{\mathrm{c}}$), which is the force in the direction of the motion. Figure 8

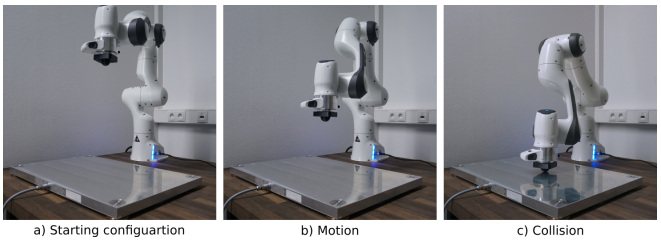


Fig. 7: Offline collision monitoring experimental setup with force plate

depicts the estimated $\hat{f}_{\mathrm{z}}$ by different methods. As can be seen in all figures, the model-adaptive collision detection rejects the modeling errors with satisfactory accuracy. The smaller modeling errors allow smaller detection threshold selection, leading to higher detection sensitivity. The other methods, however, output large modeling errors, which makes it especially difficult to detect short-impact low-amplitude collisions. In fact, the dynamic threshold of the adaptive momentum observer (with adaptive FIR filter of length 32) has such a large error that the collision at 0.2 m/s is not detected, at all. The modeling error in the model-adaptive collision detection method never exceeds 1 N, in Fig. 8 (a). Therefore, we may take 5 N as the highly conservative detection threshold. According to Fig. 8 (c), the detection delay is less than 5 ms for the low velocity (smallest τ_{ext}) motion, which shows excellent performance. In the next section, we attached a real IMU to the robot end-effector and further developed the experimental evaluations.

Online mode: The online experiment setup (Fig. 9) is similar to our previous work in [2]. An IMU is installed at the robot end-effector. The numerical robot dynamics model is also provided by the robot interface at each sampling time (1 ms). The data obtained from the robot (including numerical robot dynamics model and proprioceptive sensing) and the IMU are fed to an Intel Core i7-7700 with 8 GB memory processor. The remainder signal is computed in a Simulink model, running on this processor. The regressor matrix $\boldsymbol{Y}(\cdot) \in \mathbb{R}^{43\times 7}$ and its components are obtained based on the geometrical information provided online by the robot

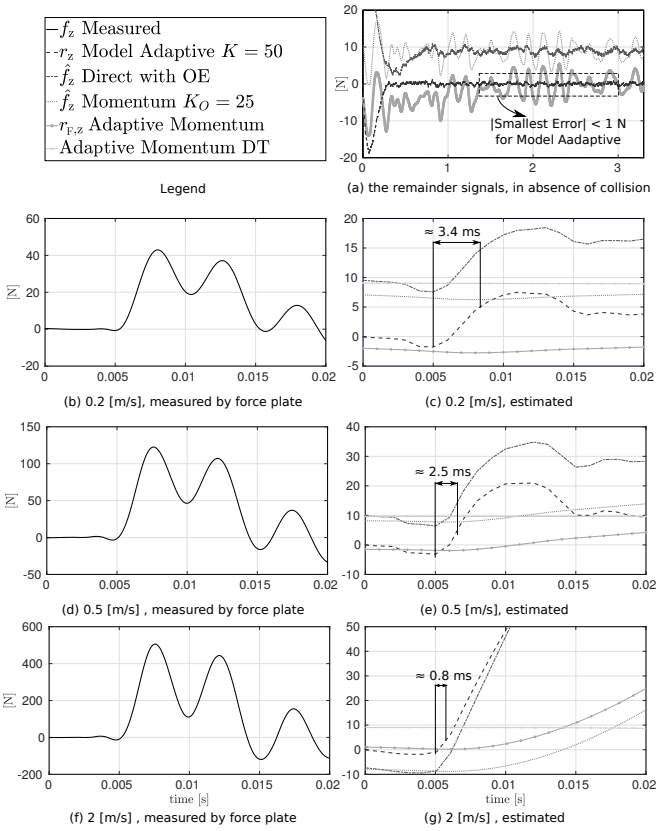


Fig. 8: Comparison of traditional collision detection methods with model-adaptive collision detection. Adaptive Momentum DT refers to the dynamic threshold computed by the adaptive momentum observer. Also momentum and Direct with OE stand for the momentum observer and the observer-extended direct method in this figure, respectively. f_z refers to the external force measured by the force plate. The time window specified in the right column indicates the model-adaptive collision detection delay with a conservative threshold of 5 N, which is the detection threshold for the method.

manufacturer. Compared to our previous experiments, we have decreased the magnitude of the contact force in order to underline the sensitivity advantage of our method.

The joint velocity and acceleration nonlinear estimator

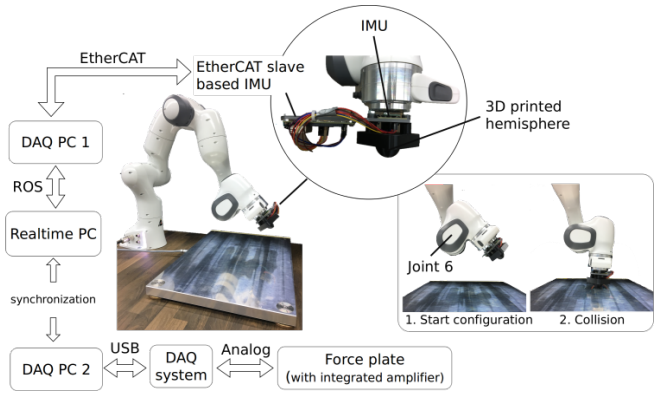


Fig. 9: Online collision monitoring experimental setup against force plate

requires at least one IMU per link. Since we only use one IMU in this experiment, we move one joint (6th joint) only. The robot moves slowly from the start configuration in Fig. 9 to the collision pose and gently touches the force plate. This contact wrench is then rotated and transformed to the robot 6th joint frame as the ground truth. K_O of the momentum observer is 25, which guarantees the method's robustness in practice. For the model adaptive collision detection method, $K = 100$ rad is chosen empirically. Figure 10 depicts the estimated $\hat{\tau}_{\text{ext}}$ before and during collision. As can be seen from Fig. 10 (a), the adaptive momentum observer has relatively large dynamic thresholds. As a result, the monitoring signal never violates the threshold and therefore, the algorithm fails to detect such a small-amplitude external force. Furthermore, according to Fig. 10 (b), the momentum observer and the observer-extended method have slow drift due to the modeling errors. On the other hand, $\hat{\tau}_{\text{ext}}$ estimated by the model-adaptive collision detection method oscillates around 0 with relatively small amplitude, which implies satisfactory modeling error rejection. Moreover, when the robot collides with the force plate (Fig. 10 (c)), the momentum observer and the observer-extended direct method have errors as large as the impact amplitude, which makes the detection unreliable. It is due to this error that the collision threshold is typically set to 1 Nm in the momentum observer. In other words, collision will not be reported unless its amplitude is greater than 1 Nm at the joint in momentum observers. While, the error of the proposed solution does not exceed 0.1 Nm. Meaning that small amplitude external forces ($\geq$ 0.1 Nm) can be effectively detected. With such 0.1 Nm threshold, the collision is detected within 1.2 ms.

Even with the presence of modeling errors, the observer-extended direct method detects the collision faster and more accurately than the momentum observer. The model-adaptive collision detection method, however, follows the measured τ_{ext} with relatively accurate amplitude. In the observer-extended direct method and model-adaptive collision detection method the bandwidth is determined by the IMU and the nonlinear estimator. Therefore, both methods follow a roughly similar curvature for the estimated signal $\hat{\tau}_{\text{ext}}$. Since the model-adaptive collision detection acts as a high-pass filter, the estimated external torque tends to zero, quickly after the peaks.

Furthermore, as we have shown in our previous works, due to the extended Kalman filter that is used as the nonlinear estimator, the estimated joint variables are smooth and numerically stable. Therefore, the numerical differentiation of the regressor signal is not even possible but relatively accurate, as the experiments also confirm. Calculating large matrices such as the pseudo-inverse of the regressor (in our simulation and experiment, $\hat{\boldsymbol{Y}}^{+}(\cdot) \in \mathbb{R}^{43\times7}$) might be computationally expensive, though. However, we would argue that with today's powerful, yet relatively cheap computers calculating such a matrix is a matter of micro seconds. Specifically, the average computation time of the remainder signal (which includes the estimation of the joint variables and computation of the full-size regressor matrix, its pseudo inverse and derivative) with model-adaptive collision detection is approximately 415 micro seconds at each time step (1 ms), in our experimental setup.

VI. Conclusions

Modern collision detection methods are typically dependent on the available robot model. Therefore, the limited quality of the available model reduces the detection reliability and sensitivity, leading to an increase in collision detection delay. In this paper a novel method for estimating the robot manipulator modeling errors for improving collision detection accuracy was introduced. The regressor-based observer monitors and rejects modeling errors from estimated external

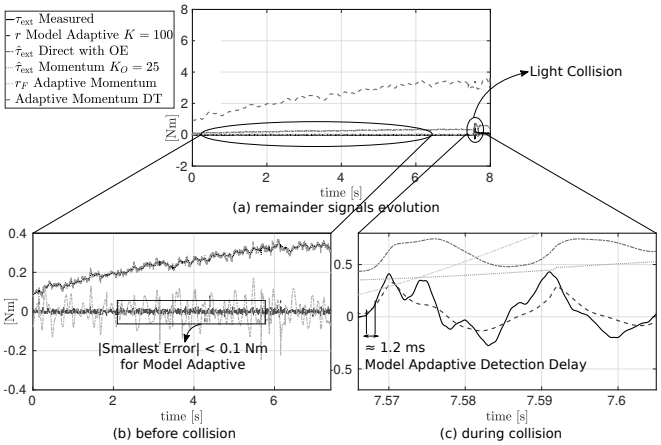


Fig. 10: Remainder signal in the 6th joint estimated online by four methods for a very light collision at $t \approx 7.5$ s. The legends across the figure are the measured/estimated torques by i.) the force plate s (ground truth), ii.) model-adaptive collision detection with experimental observer gain $K = 100$ (the proposed solution), iii.) observer-extended direct method, iv.) momentum observer with observer gain $K_O = 25$, v.) adaptive momentum observer, with with dynamic thresholding. Our model-adaptive collision detection shows smallest modeling error (< 0.1 Nm). Taking 0.1 Nm as detection threshold, the impact is detected after ≈ 1.2 ms in comparison to the other methods that hardly can detect at all.

torques. The main idea behind the method is based on the fact that the former have slow dynamics compared to abrupt collisions. In turn, both effects can be separated according to their dynamics, ultimately leading to very low detection thresholds and high-bandwidth detection. The simulations and experiments conducted on a 7-DoF robot manipulator show excellent results against available state-of-the-art methods. It is worth noticing that the introduced algorithm relies on the robot dynamics regressor with according joint velocity and acceleration information, a necessity that was intentionally avoided in previous methods. However, in our recent work we showed that these quantities can indeed be estimated with satisfactory precision and bandwidth by fusing link-side position information with modern IMU measurements installed on the robot links.

Overall, our presented work is a significant step forward in bringing robot collision detection to its practically feasible performance.

Acknowledgment and Conflicts of Interest

We greatly acknowledge the funding of this work by Franka Emika GmbH and the Alfried Krupp von Bohlen und Halbach Foundation. This work was also supported by the European Unions Horizon 2020 research and innovation programme as part of the project I.AM. under grant no. 871899 and project ILIAD under grant no. 732737. Please also note that S. Haddadin has a potential conflict of interest as shareholder of Franka Emika GmbH.

References

[1] S. A. B. Birjandi, J. Kühn, and S. Haddadin, "Joint velocity and acceleration estimation in serial chain rigid body and flexible joint manipulators," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2019, pp. 7503–7509.

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

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

[4] A. De Luca and R. Mattone, "Actuator failure detection and isolation using generalized momenta," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 1. IEEE, 2003, pp. 634–639.

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

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

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

[8] S. Haddadin, A. Albu-Schäffer, A. De Luca, and G. Hirzinger, "Collision detection and reaction: A contribution to safe physical human-robot interaction," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 3356–3363.

[9] S. Takakura, T. Murakami, and K. Ohnishi, "An approach to collision detection and recovery motion in industrial robot," in *15th Annual Conference of IEEE Industrial Electronics Society*, Nov 1989, pp. 421–426 vol.2.

[10] K. Lim and M. Eslami, "Robust adaptive controller designs for robot manipulator systems," *IEEE Journal on robotics and automation*, vol. 3, no. 1, pp. 54–66, 1987.

[11] R. Horowitz and M. Tomizuka, "An adaptive control scheme for mechanical manipulatorscompensation of nonlinearity and decoupling control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 108, no. 2, pp. 127–135, 06 1986.

[12] Y. J. Heo, D. Kim, W. Lee, H. Kim, J. Park, and W. K. Chung, "Collision detection for industrial collaborative robots: A deep learning approach," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 740–746, 2019.

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

[14] C. N. Cho, J. T. Hong, and H. J. Kim, "Neural network based adaptive actuator fault detection algorithm for robot manipulators," *Journal of Intelligent & Robotic Systems*, vol. 95, no. 1, pp. 137–147, 2019.

[15] S. Dubowsky and D. T. DesForges, "The application of model-referenced adaptive control to robotic manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 101, no. 3, pp. 193–200, 09 1979.

[16] A. Balestrino, G. De Maria, and L. Sciavicco, "An adaptive model following control for robotic manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 105, no. 3, pp. 143–151, 09 1983.

[17] M. Guo, H. Zhang, C. Feng, M. Liu, and J. Huo, "Manipulator residual estimation and its application in collision detection," *Industrial Robot: An International Journal*, 2018.

[18] P. Cao, Y. Gan, and X. Dai, "Model-based sensorless robot collision detection under model uncertainties with a fast dynamics identification," *International Journal of Advanced Robotic Systems*, vol. 16, no. 3, p. 1729881419853713, 2019.

[19] D.-G. Zhang and J. Angeles, "Impact dynamics of flexible-joint robots," *Computers & structures*, vol. 83, no. 1, pp. 25–33, 2005.

[20] C. G. Atkeson, C. H. An, and J. M. Hollerbach, "Estimation of inertial parameters of manipulator loads and links," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 101–119, 1986.

[21] M. Gautier and W. Khalil, "A direct determination of minimum inertial parameters of robots," in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*. IEEE, 1988, pp. 1682–1687.

[22] M. Gautier and W. Khalil, "Direct calculation of minimum set of inertial parameters of serial robots," *IEEE Transactions on robotics and Automation*, vol. 6, no. 3, pp. 368–373, 1990.

[23] H.-S. Liu and Y. Huang, "Bounded adaptive output feedback tracking control for flexible-joint robot manipulators," *Journal of Zhejiang University-SCIENCE A*, vol. 19, no. 7, pp. 557–578, 2018.

[24] K. L. Yano, *The theory of Lie derivatives and its applications*. North-Holland, 1957, vol. 3.

[25] C. R. Gaz, M. Cognetti, A. A. Oliva, P. R. Giordano, and A. De Luca, "Dynamic identification of the franka emika panda robot with retrieval of feasible parameters using penalty-based optimization," *IEEE Robotics and Automation Letters*, 2019.

[26] *BMI055 small, versatile 6DoF sensor module*, Bosch, 7 2014, rev. 1.2.

[27] J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *The international journal of robotics research*, vol. 6, no. 3, pp. 49–59, 1987.