frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 23 November 2020
doi: 10.3389/frobt.2020.571574

Practical Aspects of Model-Based Collision Detection

Shamil Mamedov[1]* *and Stanislav Mikhel*[2]

[1] *Laboratory of Mechatronics, Control and Prototyping, Center for Technologies in Robotics & Mechatronics Components, Innopolis University, Innopolis, Russia,* [2] *Industrial Robotics Lab, Center for Technologies in Robotics & Mechatronics Components, Innopolis University, Innopolis, Russia*

Recently, with the increased number of robots entering numerous manufacturing fields, a considerable wealth of literature has appeared on the theme of physical human-robot interaction using data from proprioceptive sensors (motor or/and load side encoders). Most of the studies have then the accurate dynamic model of a robot for granted. In practice, however, model identification and observer design proceeds collision detection. To the best of our knowledge, no previous study has systematically investigated each aspect underlying physical human-robot interaction and the relationship between those aspects. In this paper, we bridge this gap by first reviewing the literature on model identification, disturbance estimation and collision detection, and discussing the relationship between the three, then by examining the practical sides of model-based collision detection on a case study conducted on UR10e. We show that the model identification step is critical for accurate collision detection, while the choice of the observer should be mostly based on computation time and the simplicity and flexibility of tuning. It is hoped that this study can serve as a roadmap to equip industrial robots with basic physical human-robot interaction capabilities.

Keywords: dynamic identification, collision detection, disturbance observer, human-robot interaction, observer design, physical human-robot interaction

OPEN ACCESS

Edited by:
Alessandra Sciutti,
Italian Institute of Technology (IIT), Italy

Reviewed by:
Gabriele Nava,
Italian Institute of Technology (IIT), Italy
Matej Hoffmann,
Czech Technical University in Prague, Czechia

***Correspondence:**
Shamil Mamedov
sh.mamedov@innopolis.ru

Specialty section:
This article was submitted to Human-Robot Interaction, a section of the journal Frontiers in Robotics and AI

Received: *11 June 2020*
Accepted: *07 October 2020*
Published: *23 November 2020*

Citation:
Mamedov S and Mikhel S (2020) Practical Aspects of Model-Based Collision Detection. Front. Robot. AI 7:571574. doi: 10.3389/frobt.2020.571574

1. INTRODUCTION

With the advancement and proliferation of robotics, particularly in the manufacturing industry, human-robot interaction (HRI) is becoming more complex. The safe collaboration of humans and robots is being studied within the physical human-robot interaction (pHRI) field, which considers such collaboration to be based upon the ability of robots to sense their environment. In its simplest form pHRI boils down to monitoring robots' collisions with the environment or humans and stopping it if a collision has been detected. A more sophisticated realization of pHRI in terms of software should include collision localization (Haddadin et al., 2017; Mikhel et al., 2019), collision reaction strategies (Haddadin et al., 2008), relevant control techniques such as force/impedance control, and real-time motion planning (De Santis et al., 2008). In this paper, we focus on a simple form of pHRI which can be used independently—in situations when a robot is performing a task and a human happens to be in its way, or in situations when a human, by deliberately coming into contact with a robot, can prevent it from hurting itself, others, or damage the environment—or as a part of more sophisticated pHRI used for supportive, collaborative, or cooperative interactions (Haddadin and Croft, 2016).

There are two main approaches to collision detection: model-free and model-based. Model-free methods usually compare torques/currents needed to execute the trajectory with real torques applied to the robot, and if a collision occurs, a certain threshold is exceeded (Takakura et al., 1989). These methods require access to the trajectory planner and controller of the robot; however, such access is not available to the users of the industrial manipulators. On the other hand, model-based methods use the dynamic model of a robot to estimate the disturbance torques using observers, and, based on the value of such torques, conclude the presence of a collision.

Most studies on model-based pHRI focus primarily on collision detection schemes or external torque estimation where they assume that the real model of the robot is known (Haddadin et al., 2017; Garofalo et al., 2019). However, systematic treatment of all three aspects of basic pHRI–model identification, observer design, and collision detection–and the relationship between the three has not been addressed yet. This study aims to bridge this gap and put forward a clear roadmap for enabling industrial robots with basic pHRI capabilities. The remaining part of the paper proceeds as follows: in the Model Identification subsection we outline the main steps needed for accurate dynamic parameters estimation; for each step we review relevant literature and provide practical advice. Compared to Wu et al. (2010) our literature review includes important recent contributions and is primarily intended for practitioners. In the Disturbance Estimation subsection, we review and compare several algorithms in terms of computation time as well as simplicity and flexibility of tuning. Our paper reviews alternative observers to the well-established pHRI community momentum observer and provides a new perspective of the momentum observer. In the Collision Detection subsection, we review both classical and recently proposed collision detection algorithms and discuss their advantages and limitations. Finally, we demonstrate the roadmap on a case study that involved manipulator UR10e.

2. THEORETICAL PRELIMINARIES

Industrial manipulators might be subject to external disturbances due to collisions or contact with the environment. Usually, industrial manipulators are supplied with a controller and a trajectory planner. The majority of such manipulators allow users to input high-level tasks and measure torques/currents, positions, and velocities; however, such robots give users access to neither control nor the output of the trajectory planner. Given everything mentioned above, a question arises as to the most efficient way of detecting collisions of robot with their environment. In this section, we answer this question by consecutively discussing three main cornerstones that constitute model-based collision detection: model identification, observation, and detection.

2.1. Model Identification

An accurate dynamic model is the paramount element of any model-based collision detection algorithm. The manipulator dynamics equation generally used is

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

where $\boldsymbol{q},\ \dot{\boldsymbol{q}},\ \ddot{\boldsymbol{q}} \in \mathbb{R}^n$ are the vectors of generalized coordinates, velocities, and acceleration, respectively; $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is an inertia matrix; $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ is a matrix of Coriolis and centrifugal forces; $\boldsymbol{F}(\dot{\boldsymbol{q}}) \in \mathbb{R}^n$ and $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$ are the vectors of friction and gravitational torques; $\boldsymbol{\tau}$ is the vector of the actuation torques; $\boldsymbol{\tau}_{ext}$ is the vector of the torques induced at the joints by the contact forces, in the absence of contact/collision with the environment $\boldsymbol{\tau}_{ext} = \boldsymbol{0}$ (Siciliano et al., 2010). The model requires the exact knowledge of the kinematic and dynamic parameters. In real life, kinematic parameters are known (because of manufacturing standards and calibration of robots before shipping) and provided by manufacturers. However, the latter usually do not provide dynamic parameters, with rare exceptions like, for example, Universal Robots that provide CAD models. Even with the CAD model the torque prediction lacks accuracy because it contains neither friction nor motor inertia parameters. A more accurate model can be obtained by performing identification. In this subsection, we provide an outline of the identification procedure.

The process of the identification of the dynamic parameters of mechanical systems can be divided into seven main steps (Swevers et al., 2007):

1. derivation of the dynamic model in regressor form
2. computation of base inertial parameters
3. trajectory planning
4. experiment conducting and collecting data
5. data processing
6. parameter estimation
7. validation

Identification starts by deriving the dynamic model of a manipulator moving in free space ($\boldsymbol{\tau}_{ext} = \boldsymbol{0}$) in a linear regressor form

$$\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{\pi} = \boldsymbol{\tau}, \tag{2}$$

where $\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n_\pi}$ is the regressor matrix, that can be obtained symbolically or numerically using recursive Lagrange-Euler formulation (Siciliano et al., 2010), modified recursive Newton-Euler formulation (Atkeson et al., 1986), or screw theory formulation (Garofalo et al., 2013), and $\boldsymbol{\pi} \in \mathbb{R}^{n_\pi}$ is the vector of dynamic parameters (standard parameters). It consists of inertial parameters of each link modeled as a rigid body, an actuator, and friction parameters of each joint. The inertial parameters of a link include mass m_i, the first moment of inertia $\boldsymbol{h}_i = m_i\boldsymbol{r}_i$ (the product of a mass and the position of the center of mass), and inertia tensor with respect to the origin of the link $\boldsymbol{I}_i$. Friction parameters are the coefficient of the Coloumb friction f_c, the coefficient of the viscous friction f_v, and an offset due to motor current offset f_o. Actuator parameter is reflected rotor inertia J_i [If servomotor inertia $I_{m,i}$ and the gear ratio of the transmission k_{ri} are known, then reflected rotor inertia can be computed as $J_i = I_{m,i}/k_{ri}^2$ and the components of the regressor corresponding to reflected inertia should be moved to the right-hand side of Equation (2)]. In total, the vector of dynamic parameters for each link is

$$\boldsymbol{\pi}_i = \left[I^i_{xx}\ I^i_{xy}\ I^i_{xz}\ I^i_{yy}\ I^i_{yz}\ I^i_{zz}\ \boldsymbol{h}^T_i\ m_i\ J_i\ f^i_v\ f^i_c\ f^i_o \right]^T \in \mathbb{R}^{14}. \quad (3)$$

Remark. More advanced friction models, for example, models with the Stribeck effect are not linear in parameters, yield a non-linear regressor matrix, that significantly complicates the identification process. However, if the linear friction model does not accurately describe friction torques, then it can be replaced by a non-linear model after identifying all other parameters. The non-linear model is usually identified for each link separately using non-linear least squares (Gaz et al., 2018).

Not all the dynamic parameters enter the dynamic equation of the robot independently; some of them do not enter at all, some of them enter in linear combinations with other parameters. Due to that, the regressor matrix $\boldsymbol{Y}(\cdot)$ has zero or linearly dependent columns leading to singularity which causes problems during trajectory planning and parameter estimation. Several methods have been proposed to overcome these limitations by deriving a set of identifiable parameters called base parameters $\boldsymbol{\pi}_b$ and corresponding base regressor matrix $\boldsymbol{Y}_b(\cdot)$. Gautier and Khalil (1990) proposed an analytical method based on the recursive properties of robot energy. However, a year later Gautier (1991) proposed numerical methods based on QR- and SVD- decompositions. The analytical method was derived using Denavit–Hartenberg convention and it employs heuristics for determining linear dependence or independence of actuator parameters. Numerical methods (especially QR) are easy to implement and not limited to a specific convention, thus they should be the first choice when approaching parameter reduction.

The choice of trajectory significantly affects the accuracy of identification. For example, random point-to-point motion will most probably not lead to accurate parameter estimates because of the lack of property called persistence of excitation (Anderson, 1982). To find a persistently exciting trajectory it is necessary to carry out trajectory planning that is usually posed as an optimization problem (non-linear program) that aims to find $\boldsymbol{q}_*(t)$, $\dot{\boldsymbol{q}}_*(t)$, and $\ddot{\boldsymbol{q}}_*(t)$ such that chosen objective function associated with persistence of excitation is minimized. The most commonly used objective functions are condition number of the observation matrix [cond($\boldsymbol{W}_b$), for the definition of $\boldsymbol{W}_b$ see Equation (5)] or log-determinant of the moment matrix (log(det($\boldsymbol{W}^T_b\boldsymbol{W}_b$))) (Hollerbach et al., 2016). For industrial manipulators, several families of trajectories were proposed in the literature: fifth-order polynomials (Atkeson et al., 1986), truncated Fourier series (Swevers et al., 1997), and their combination (Wu et al., 2012). Fifth order polynomials have fewer parameters to optimize and can guarantee zero velocity and acceleration in the beginning and at the end of the trajectory. Periodic trajectories have an advantage in terms of data processing, as the same periodic trajectory can be executed several times and the collected data can be averaged. Moreover, they allow for exact frequency domain differentiation. However, there is a disadvantage in terms of abrupt change in initial velocities and accelerations ($\dot{q}_i(0) \neq 0$, $\dot{q}_i(T) \neq 0$, $\ddot{q}_i(0) \neq 0$, $\ddot{q}_i(T) \neq 0$), which may cause robot vibration as well as hinder accurate trajectory tracking. Even in cases when zero initial and final velocities and accelerations are imposed as constraints of the optimization problem, the solver struggles to satisfy the constraints. The addition of a fifth-order polynomial solves this problem leading to the trajectory in the form

$$q_i(t) = \sum_{k=1}^{N} \left[\frac{a_{i,k}}{\omega_f k} \sin(\omega_f k t) - \frac{b_{i,k}}{\omega_f k} \cos(\omega_f k t) \right] + \sum_{i=0}^{5} c_{i,k} \left(t - \lfloor \frac{t}{T} \rfloor T \right)^k, \quad (4)$$

where w_f is the fundamental frequency, N is the number of harmonics, and $\lfloor\ \rfloor$ is floor function.

Once trajectory planning has been performed, it is necessary to execute the trajectory on a real robot in closed-loop and record currents/torques, generalized positions $\boldsymbol{q}(t)$, and, if available, generalized velocities $\dot{\boldsymbol{q}}(t)$. After the trajectory execution, the recorded data should be processed to remove measurement noise and to reconstruct missing variables from the available ones. If $\dot{\boldsymbol{q}}(t)$ are measurable, then only accelerations have to be estimated from velocities; otherwise, velocities have to be found first based on position measurements. As data processing is performed offline, the central difference scheme should be used for a more accurate estimation of derivatives (Hoffman and Frankel, 2018). However, derivative estimates are noisy even with a central difference scheme, so they should be filtered. To avoid introducing delays, non-casual filters (a zero-phase filter) should be used during filtering (Mitra and Kuo, 2006).

For parameter estimation, it is necessary to calculate observation matrix $\boldsymbol{W}_b(\cdot)$ and observation vector $\boldsymbol{T}$ from the collected and processed data as

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{\tau}(t_1) \\ \dots \\ \boldsymbol{\tau}(t_k) \\ \dots \\ \boldsymbol{\tau}(t_n) \end{bmatrix},\ \boldsymbol{W}_b = \begin{bmatrix} \boldsymbol{Y}_b\big(\boldsymbol{q}(t_1), \dot{\boldsymbol{q}}(t_1), \ddot{\boldsymbol{q}}(t_1)\big) \\ \dots \\ \boldsymbol{Y}_b\big(\boldsymbol{q}(t_k), \dot{\boldsymbol{q}}(t_k), \ddot{\boldsymbol{q}}(t_k)\big) \\ \dots \\ \boldsymbol{Y}_b\big(\boldsymbol{q}(t_n), \dot{\boldsymbol{q}}(t_n), \ddot{\boldsymbol{q}}(t_n)\big) \end{bmatrix}; \quad (5)$$

then, compute base parameters estimates employing ordinary least squares

$$\hat{\boldsymbol{\pi}}_b = (\boldsymbol{W}^T_b\boldsymbol{W}_b)^{-1}\boldsymbol{W}^T_b\boldsymbol{T}, \quad (6)$$

which minimizes the squared torque prediction error, i.e., the cost function $J = \frac{1}{2}||\boldsymbol{T} - \boldsymbol{W}_b\boldsymbol{\pi}_b||_2$. However, Equation (5) does not guarantee the physical consistency of parameters, thus the properties of the dynamic model (Siciliano et al., 2010). Since model-based techniques are extensively used in robotics, and the properties of dynamic equations are used to prove theorems and to guarantee convergence of controllers and observers, several important developments have been made in dynamic parameter identification in terms of the physical consistency of parameters. First, Sousa and Cortesão (2014) proposed to impose linear matrix inequality constraint on kinetic energy, more specifically—on the generalized inertia matrix of each link (semi-physical consistency)

$$\begin{bmatrix} \boldsymbol{I}_i & \boldsymbol{S}(m_i\boldsymbol{r}^T_i) \\ \boldsymbol{S}(m_i\boldsymbol{r}^T_i) & m_i\boldsymbol{I} \end{bmatrix} \succ 0, \quad (7)$$

where $\boldsymbol{S}(\cdot)$ is the operator that maps a 3×1 vector into a 3×3 skew-symmetric matrix. Then, Wensing et al. (2017) and Sousa and Cortesao (2019) proposed to impose triangle inequality as the linear matrix inequality constraint (physical consistency)

$$\begin{bmatrix} \frac{1}{2}tr(\boldsymbol{I}_i)\boldsymbol{I} - \boldsymbol{I}_i & \boldsymbol{r}_i \\ \boldsymbol{r}_i^T & m_i \end{bmatrix} \succ 0. \tag{8}$$

Besides the constraints on inertial parameters of links, it is possible to impose constraints on other dynamic parameters: reflected inertia ($J_i > 0$) and friction ($f_v^i > 0$, $f_c^i > 0$). To impose physical or semi–physical consistency constraints while searching for base parameters, Sousa and Cortesão (2014) proposed a bijective mapping from the base and dependent parameters to standard parameters. It does not require additional computation, rather uses the result of base parameter calculation.

For some robots, instead of torque measurements, current measurements are available. In such cases, in addition to $\boldsymbol{\pi}$ ($\boldsymbol{\pi}_b$), it is necessary to estimate drive gains. For that, Gautier and Briot (2014) proposed to carry out an additional experiment with a load attached to the end-effector of the robot and use the extended observation vector $\boldsymbol{T}$ and matrix $\boldsymbol{W}_{be}$

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{I}_u \\ \boldsymbol{I}_l \end{bmatrix} \boldsymbol{K},\ \boldsymbol{W}_{be} = \begin{bmatrix} \boldsymbol{W}_b & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{W}_b & \boldsymbol{W}_{lu} & \boldsymbol{W}_{lk} \end{bmatrix}, \tag{9}$$

where $\boldsymbol{I}_u$ and $\boldsymbol{I}_l$ are current measurements for unloaded and loaded cases respectively; $\boldsymbol{K}$ is a vector of drive gains; $\boldsymbol{W}_l = \begin{bmatrix} \boldsymbol{W}_{lu} & \boldsymbol{W}_{lk} \end{bmatrix} \in \mathbb{R}^{10}$ is the observation matrix of the load, divided into parts corresponding to unknown and known inertial parameters. Using Equation (9), it is possible to derive the dynamics equation in a regression form linear with respect to unknown parameters

$$\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{W}_{lk}\boldsymbol{\pi}_{lk} \end{bmatrix} = \begin{bmatrix} \boldsymbol{W}_b & \boldsymbol{0} & \boldsymbol{I}_u \\ \boldsymbol{W}_b & \boldsymbol{W}_{lu} & \boldsymbol{I}_l \end{bmatrix} \begin{bmatrix} \boldsymbol{\pi}_b \\ \boldsymbol{\pi}_{lu} \\ \boldsymbol{K} \end{bmatrix}, \tag{10}$$

that allows estimating base dynamic parameters, unknown load inertial parameters, and drive gains, satisfying the (semi)-physical consistency constraints provided that some of the parameters of the load $\boldsymbol{\pi}_{lk}$ are known (at least one), usually it is mass because it is easier to measure compared with other inertial parameters. Gautier and Briot (2014) proposed another method for computing drive gains using total least squares to avoid bias errors caused by the same $\boldsymbol{q}(t)$, $\ddot{\boldsymbol{q}}(t)$, and $\ddot{\boldsymbol{q}}(t)$ used in both parts of Equation (10). However, it is not clear how to impose constraints in that case; without constraints, such a computation method can result in negative drive gains.

Remark. For drive gain identification, it is important to choose a load such that torques for loaded and unloaded trajectories differ. It can be achieved by choosing a heavy load and attaching it in a way that is not aligned with the axis of the end-effector.

After the identification of parameters, model validation should be performed to assess the accuracy of torque prediction for any arbitrary trajectory executed by the robot. One example of a metric that can be used to measure accuracy is root mean square for the prediction error. If torque predictions are poor, then steps 3 to 6 should be repeated until the accurate model of the robot is obtained.

2.2. Disturbance Estimation

Ideally, to implement collision detection, it is necessary to estimate $\boldsymbol{\tau}_{ext}$, as it is the only indicator of the collision. However, even with the most recent model identification methods, it is impossible to reconstruct the exact model of a robot (Equation 1). The most accurate model available is

$$\hat{\boldsymbol{M}}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \hat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{F}}(\dot{\boldsymbol{q}}) + \hat{\boldsymbol{g}}(\boldsymbol{q}) = \hat{\boldsymbol{\tau}} + \boldsymbol{\tau}_{ext}, \tag{11}$$

with a mismatch between the real model and its estimate (unmodeled dynamics) being equal to

$$\boldsymbol{\tau}_{um} = \Delta\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \Delta\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \Delta\boldsymbol{F}(\dot{\boldsymbol{q}}) + \Delta\boldsymbol{g}(\boldsymbol{q}). \tag{12}$$

Using Equation (12) it is possible to rewrite Equation (11) as

$$\hat{\boldsymbol{M}}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \hat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{F}}(\dot{\boldsymbol{q}}) + \hat{\boldsymbol{g}}(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_d. \tag{13}$$

Thus, the parameter that can be estimated is lumped disturbance $\boldsymbol{\tau}_d = -\boldsymbol{\tau}_{um} + \boldsymbol{\tau}_{ext}$. If variables $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$, $\boldsymbol{\tau}$ can be measured, then the vector of disturbance torques can be found from Equation (13) through straightforward subtraction. In practice, this approach is hardly applicable, since acceleration measurements are not available, and estimating them from noisy velocity measurements through numerical differentiation further amplifies noise. Therefore, more sophisticated techniques should be considered that would allow for estimating $\boldsymbol{\tau}_d$ in absence of acceleration measurements. In the following two subsections we review several disturbance observers proposed in the context of pHRI.

2.2.1. Non-linear Disturbance Observer (NDOB)

Inspired by the disturbance observer for linear systems, Chen et al. (2000)—first assuming that acceleration measurements are available—proposed an observer for lumped disturbance estimation

$$\dot{\hat{\boldsymbol{\tau}}}_d = \boldsymbol{L}\boldsymbol{e},\ \boldsymbol{e} = (\boldsymbol{\tau}_d - \hat{\boldsymbol{\tau}}_d), \tag{14}$$

where $\boldsymbol{L} \in \mathbb{R}^{n\times n}$ is the observer gain matrix. In general, the error dynamics of the observer is

$$\dot{\boldsymbol{e}} = \dot{\boldsymbol{\tau}}_d - \boldsymbol{L}\boldsymbol{e}, \tag{15}$$

if there is no prior information on $\boldsymbol{\tau}_d$, then $\dot{\boldsymbol{\tau}}_d$ is set to zero yielding

$$\dot{\boldsymbol{e}} + \boldsymbol{L}\boldsymbol{e} = 0. \tag{16}$$

When $\boldsymbol{L}$ is a constant stable matrix, the estimation error tends to be zero. However, due to lack of acceleration measurements,

the authors modified the original observer by introducing an auxiliary variable $\boldsymbol{z} = \hat{\boldsymbol{\tau}}_d - \boldsymbol{\psi}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ such that

$$\frac{d}{dt}\boldsymbol{\psi}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{L}\hat{\boldsymbol{M}}(\boldsymbol{q})\ddot{\boldsymbol{q}}, \tag{17}$$

while the dynamics of $\boldsymbol{z}$ is:

$$\dot{\boldsymbol{z}} = -\boldsymbol{L}\boldsymbol{z} + \boldsymbol{L}\Big[\hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{F}}(\dot{\boldsymbol{q}}) + \hat{\boldsymbol{g}}(\boldsymbol{q}) - \boldsymbol{\tau} - \boldsymbol{\psi}(\boldsymbol{q}, \dot{\boldsymbol{q}})\Big]. \tag{18}$$

(It is obtained by taking the time derivative of $\boldsymbol{z}$ followed by substituting Equation (14) for $\dot{\hat{\boldsymbol{\tau}}}_d$ and Equation (13) for $\boldsymbol{\tau}_d$.) The special choice of $\boldsymbol{\psi}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ used in modified observer results is the same error dynamics as the original observer (Equation 14).

The main difficulty in designing NDOB is choosing gain matrix $\boldsymbol{L}$ and vector $\boldsymbol{\psi}(\boldsymbol{q}, \dot{\boldsymbol{q}})$. Chen et al. (2000) proposed a design procedure for planar 2 degrees of freedom (DOF) manipulator, Nikoobin and Haghighi (2009) extended it to n-DOF planar manipulators. Later, Mohammadi et al. (2013) extended the observer to the spatial case, more specifically authors proposed to choose $\boldsymbol{L}$ and $\boldsymbol{\psi}$ as

$$\boldsymbol{\psi}(\dot{\boldsymbol{q}}) = \boldsymbol{X}^{-1}\dot{\boldsymbol{q}},\ \boldsymbol{L} = \boldsymbol{X}^{-1}\hat{\boldsymbol{M}}(\boldsymbol{q})^{-1}. \tag{19}$$

For $\dot{\boldsymbol{\tau}}_d = 0$ using candidate Lyapunov function $V = \boldsymbol{e}^T\boldsymbol{X}^T\hat{\boldsymbol{M}}(\boldsymbol{q})\boldsymbol{X}\boldsymbol{e}$ they proved that if $\boldsymbol{X}$ is invertible and there exists positive definite symmetric matrix $\boldsymbol{\Gamma}$ such that $\boldsymbol{X} + \boldsymbol{X}^T - \boldsymbol{X}^T\dot{\hat{\boldsymbol{M}}}(\boldsymbol{q})\boldsymbol{X} \geq \boldsymbol{\Gamma}$ then disturbance estimation error converges exponentially to zero (Mohammadi et al., 2013, Theorem 1). For $\dot{\boldsymbol{\tau}}_d \neq 0$ if the rate of change of the lumped disturbance is bounded i.e., $||\dot{\boldsymbol{\tau}}_d|| \leq \kappa,\ \kappa > 0$, it was proven that the estimation error exponentially converges to a ball of a certain radius (Mohammadi et al., 2013, Theorem 2). In both cases, to design $\boldsymbol{X}$ the user can either solve linear matrix inequality

$$\begin{bmatrix} \boldsymbol{Y} + \boldsymbol{Y}^T - \xi\boldsymbol{I} & \boldsymbol{Y}^T \\ \boldsymbol{Y} & \boldsymbol{\Gamma}^{-1} \end{bmatrix} \geq 0, \tag{20}$$

where $\boldsymbol{Y} = \boldsymbol{X}^{-1}$, ξ is upper bound of $||\dot{\boldsymbol{M}}(\boldsymbol{q})||$ and $\boldsymbol{I}$ is identity matrix of proper dimension, or use analytical solution for the case when matrices $\boldsymbol{Y}$ and $\boldsymbol{\Gamma}$ are scaled identity matrices

$$\boldsymbol{Y} = 0.5(\xi + 2\beta\sigma_2)\boldsymbol{I}, \tag{21}$$

where β is minimum convergence rate, σ is upper bound of inertia matrix ($\boldsymbol{M}(\boldsymbol{q}) \leq \sigma_2\boldsymbol{I}$).

2.2.2. Momentum Observer

Another widely used approach for disturbance estimation is based on generalized momentum and was proposed in the context of actuator fault detection and isolation (De Luca and Mattone, 2003). Below we show the derivation of the observer following the same steps as for NDOB. Given the definition of the momenta

$$\boldsymbol{p} = \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{22}$$

and its time derivative

$$\begin{aligned} \dot{\boldsymbol{p}} &= \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{M}}(\boldsymbol{q})\dot{\boldsymbol{q}} \\ &= \boldsymbol{\tau} + \boldsymbol{C}^T(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{F}(\dot{\boldsymbol{q}}) + \boldsymbol{\tau}_d, \end{aligned} \tag{23}$$

we can rewrite the disturbance dynamics defined in Equation (14) in terms of the rate of change of momentum

$$\dot{\hat{\boldsymbol{\tau}}}_d = \boldsymbol{L}(\dot{\boldsymbol{p}} - \dot{\hat{\boldsymbol{p}}}). \tag{24}$$

If gain matrix $\boldsymbol{L}$ is constant, then by integrating both parts we can obtain a so-called momentum observer (Haddadin et al., 2017)

$$\hat{\boldsymbol{\tau}}_d = \boldsymbol{L}\Big\{\boldsymbol{p} - \int_0^t \big(\boldsymbol{\tau} + \hat{\boldsymbol{C}}^T(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \hat{\boldsymbol{g}}(\boldsymbol{q}) - \hat{\boldsymbol{F}}(\dot{\boldsymbol{q}}) + \hat{\boldsymbol{\tau}}_d\big)d\xi - \boldsymbol{p}(0)\Big\}, \tag{25}$$

Compared with NDOB, momentum observer does not require inertia matrix inversion and each diagonal entry of the gain matrix $\boldsymbol{L}$ can be assigned independently.

Equation (23) can be interpreted as linear system

$$\dot{\boldsymbol{p}} = \boldsymbol{0}\,\boldsymbol{p} + \boldsymbol{I}\,\boldsymbol{u} + \boldsymbol{\tau}_d. \tag{26}$$

where $\boldsymbol{u} = \boldsymbol{\tau} + \boldsymbol{C}^T(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{F}(\dot{\boldsymbol{q}})$ and $\boldsymbol{0}$ is zero matrix of proper dimension. In Oh and Chung (1999), the authors used Equation (26) together with apriori information on disturbance–the disturbance is the output of some linear system (Johnson, 1971)

$$\dot{\boldsymbol{\omega}} = \boldsymbol{G}\boldsymbol{\omega},\ \boldsymbol{\tau}_d = \boldsymbol{F}\boldsymbol{\omega}, \tag{27}$$

– to obtain extended linear system

$$\begin{aligned} \dot{\boldsymbol{\zeta}} &= \begin{bmatrix} \boldsymbol{0} & \boldsymbol{F} \\ \boldsymbol{0} & \boldsymbol{G} \end{bmatrix}\boldsymbol{\zeta} + \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{0} \end{bmatrix}\boldsymbol{u} \\ \boldsymbol{y} &= \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \end{bmatrix}\boldsymbol{\zeta}, \end{aligned} \tag{28}$$

where $\boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{p}^T & \boldsymbol{\omega}^T \end{bmatrix}^T$ is the extended state vector. Then Oh and Chung (1999) used a reduced state observer in order to estimate $\boldsymbol{\tau}_d$. Hu and Xiong (2017) assumed that both states and output of Equation (28) are affected by white Gaussian noise and used the Kalman filter to estimate $\boldsymbol{\tau}_d$.

2.2.3. Sliding Mode Momentum Observer

Garofalo et al. (2019) proposed a second-order sliding mode (Super Twisting Algorithm, Fridman and Levant, 2002) extension of the classical momentum observer:

$$\begin{aligned} \dot{\hat{\boldsymbol{p}}} &= \boldsymbol{u} - \boldsymbol{T}|\hat{\boldsymbol{p}} - \boldsymbol{p}|^{\frac{1}{2}}sgn(\hat{\boldsymbol{p}} - \boldsymbol{p}) + \boldsymbol{\sigma} \\ \dot{\boldsymbol{\sigma}} &= -\boldsymbol{S}\,sgn(\hat{\boldsymbol{p}} - \boldsymbol{p}), \end{aligned} \tag{29}$$

where $\boldsymbol{S}, \boldsymbol{T} \in \mathbb{R}^{n\times n}$ are positive definite diagonal matrices with diagonal elements satisfying conditions given in Moreno and Osorio (2008) that guarantees global finite-time stability of the equilibrium point $\begin{bmatrix} \hat{\boldsymbol{p}} - \boldsymbol{p} & \boldsymbol{\sigma} - \boldsymbol{\tau}_d \end{bmatrix}^T = \boldsymbol{0}$. Convergence property

of the second order sliding mode observer can be improved—from finite time to exponential—by introducing linear correction terms (Moreno and Osorio, 2008; Garofalo et al., 2019):

$$\begin{aligned}\dot{\hat{\boldsymbol{p}}} &= \boldsymbol{u} - \boldsymbol{T}_1|\hat{\boldsymbol{p}} - \boldsymbol{p}|^{\frac{1}{2}} sgn(\hat{\boldsymbol{p}} - \boldsymbol{p}) - \boldsymbol{T}_2(\hat{\boldsymbol{p}} - \boldsymbol{p}) + \boldsymbol{\sigma} \\ \dot{\boldsymbol{\sigma}} &= -\boldsymbol{S}_1\, sgn(\hat{\boldsymbol{p}} - \boldsymbol{p}) - \boldsymbol{S}_2(\hat{\boldsymbol{p}} - \boldsymbol{p}),\end{aligned} \tag{30}$$

A sliding mode momentum observer is superior to a classical momentum observer in terms of noise attenuation and convergence time; however, it requires tuning many more parameters.

2.2.4. Filtered Dynamics Observer

A different approach to disturbance estimation was put forward in Van Damme et al. (2011) and Ho and Song (2013) where they used the idea from online dynamic parameter estimation: if both parts of Equation (13) are filtered with a strictly stable filter then there is no need for acceleration measurements. To outline the derivation, assume we choose a filter with a transfer function

$$F(s) = \frac{1}{s/\omega + 1}, \tag{31}$$

and impulse response $f(t) = \mathcal{L}^{-1}(F(s)) = \omega \exp(-\omega t)$. Multiplying both parts of Equation (13) by Equation (31) is equivalent to

$$f(t) * \{\hat{\boldsymbol{M}}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{F}}(\dot{\boldsymbol{q}}) + \hat{\boldsymbol{g}}(\boldsymbol{q})\} = f(t) * \{\boldsymbol{\tau} + \boldsymbol{\tau}_d\}, \tag{32}$$

where $*$ is used to denote convolution operation. Using properties of the convolution operation, we can rewrite Equation (32) as

$$\begin{aligned}f(t) * \{\boldsymbol{\tau} + \boldsymbol{\tau}_d\} = \dot{f}(t) * \{\hat{\boldsymbol{M}}(\boldsymbol{q})\dot{\boldsymbol{q}}\} + f(0)\hat{\boldsymbol{M}}(\boldsymbol{q})\dot{\boldsymbol{q}} - f(t)\hat{\boldsymbol{M}}(\boldsymbol{q}(0))\dot{\boldsymbol{q}}(0) \\ + f(t) * \{\hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{F}}(\dot{\boldsymbol{q}}) + \hat{\boldsymbol{g}}(\boldsymbol{q}) - \dot{\hat{\boldsymbol{M}}}(\boldsymbol{q})\dot{\boldsymbol{q}}\},\end{aligned} \tag{33}$$

where $\dot{f}(t)$ is the impulse response of a stable filter

$$F_2(s) = -\frac{\omega^2}{s + \omega}, \tag{34}$$

Rearranging Equation (33) and using properties of the rigid body dynamics, it is possible to obtain an expression for filtered disturbance

$$f(t) * \boldsymbol{\tau}_d = \dot{f}(t) * \{\boldsymbol{p}\} + \omega\boldsymbol{p} + f(t) * \{\hat{\boldsymbol{F}}(\dot{\boldsymbol{q}}) + \hat{\boldsymbol{g}}(\boldsymbol{q}) - \hat{\boldsymbol{C}}^T\dot{\boldsymbol{q}} - \boldsymbol{\tau}\}, \tag{35}$$

(For detailed derivation see Lewis et al., 2003 Chapter 6.6 or Van Damme et al., 2011). Van Damme et al. (2011) showed that momentum observer and filtered dynamics observer are equivalent for special choice of $\boldsymbol{L}$ in Equation (25): all diagonal entries are the same.

2.3. Collision Detection

From Equation (13) we know that disturbance observers estimate the sum of torques due to unmodeled dynamics and interaction. An easy and effective way to detect collisions in presence of unmodeled dynamics is to set a static threshold (upper and lower bounds on unmodeled dynamics). The value of the threshold can be chosen as a percentage from maximum torque, for example, $\pm 0.1\tau_{max}$ (Haddadin et al., 2008) or determined from data used for validation of dynamic parameters. However, inaccurate parameter identification or use of dynamic parameters from CAD may result in a high threshold yielding long collision detection time. To deal with it several solutions were proposed in the literature.

Hu and Xiong (2017) proposed to improve the dynamic model by training a neural network (multi-layer perceptron) to predict the residual between measured torques and torques found from a rigid body model. As the input of the network, they used generalized positions and velocities that, according to authors, provide the same degree of accuracy as the network with generalized positions, velocities, and accelerations. Rigid body model enhanced with neural network allowed authors to choose tight thresholds thus decrease collision detection time.

Ho and Song (2013) proposed using a 2nd order band-pass filter in Equation (33) instead of a low-pass filter (Equation 31). By carefully choosing cut-off frequencies they were able to filter out low-frequency torques due to the motion of the robot and high-frequency torques due to noise, leaving torques caused by the collision. Band-pass filtering allowed authors to decrease threshold 5-fold compared to classical momentum observer yielding to the detection of collision in 5–6 ms. However, their approach is limited to cases when the dynamics of collision is significantly faster than the dynamics of the task, moreover, band-pass filtering modifies estimated disturbance torques making them impossible to use, for example, in collision localization. Haddadin et al. (2008) and Li et al. (2019) proposed to use two observers, one to detect slow or soft collisions using low-pass filtered collision torques and one to detect fast collisions using band-pass filtered collision torques.

Sotoudehnejad et al. (2012) proposed using time-variant thresholds that take into account uncertainties in inertial parameters of the robot as well as friction parameters. To estimate uncertainties authors proposed to conduct time-consuming experiments with a robot applying various collision torque at different states. Although it sounds promising, the difference in the detection time between the time-variant threshold and constant threshold is of the order of 0.1–0.01 s. Briquet-Kerestedjian et al. (2019) proposed to also consider uncertainties due to numerical differentiation used to find generalized velocities and acceleration from noisy encoder measurements. Their approach is different from Sotoudehnejad et al. (2012) in a way they treat the model of the manipulator. Briquet-Kerestedjian et al. (2019) consider a decentralized linear model and treats non-linear coupling between joints as a disturbance. It allows authors to use linear estimation techniques such as Kalman filter. Sotoudehnejad et al. (2012) on the other hand consider a centralized non-linear model of the manipulator and use the momentum observer. In general, it is not clear if

time-consuming experiments worth the amount of time gained in collision detection time and if the time-variant threshold is robust and provide consistent result in the whole workspace of the robot.

3. CASE STUDY: UR10E

In this section, we demonstrate all the steps required to implement model-based collision detection on real robot UR10e which is a collaborative industrial robot arm from the Universal Robots company. It weighs 33.5 kg, has a 10 kg payload, and the radius of workspace up to 1,300 mm. The robot consists of six rotating joints which allow having the full degree of freedom in Cartesian space.

3.1. Software Implementation

Within the scope of this work, we developed a C++ library for processing the measurements and estimating disturbance torques (https://github.com/mikhel1984/ext_observer). The library consists of two basic abstract classes: robot and observer. The former provides an interface for defining the dynamic model of a robot. The latter allows working with an observer regardless of its internal implementation: it defines an obligatory method that takes current joint angles, velocities, torques, and time step, and returns the disturbance torque estimate. The library can be included in the source code of a robot control program, used with MATLAB (loaded as a C shared library) or robot operating system (ROS) as a standalone component.

To start working with the library, a user has to create a new robot instance inheriting the basic abstract class and define its dynamic model. The dynamic model can be supplied in two forms: functions for $\hat{\boldsymbol{M}}(\boldsymbol{q})$, $\hat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})$, $\hat{\boldsymbol{F}}(\dot{\boldsymbol{q}})$, and $\hat{\boldsymbol{g}}(\boldsymbol{q})$ generated based on symbolically derived model; numerical procedure such as recursive Newton-Euler algorithm (RNEA) implemented as $\boldsymbol{rnea}_g(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})$ where subscript g indicates the value of the gravity constant. If the dynamics of the robot is computed numerically, the problem arises related to the computation of $\hat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})^T$ used in the majority of the observers. One way to overcome this problem is to use the modified Newton-Euler algorithm (De Luca and Ferrajoli, 2009), another is to use the numerical approximation of the time derivative of the inertia matrix ($\dot{\hat{\boldsymbol{M}}} \approx \Delta\hat{\boldsymbol{M}}/\Delta t$). Even though the second method is easier to implement, its accuracy depends on sampling time Δt. We propose a simple way to improve the accuracy, by using the definition of the rate of change of the inertia matrix

$$\dot{\hat{\boldsymbol{M}}} = \left[\frac{\partial\hat{\boldsymbol{M}}}{\partial q_1}\cdots\frac{\partial\hat{\boldsymbol{M}}}{\partial q_n}\right]\dot{\boldsymbol{q}}, \tag{36}$$

where $\partial\hat{\boldsymbol{M}}_i/\partial q_i \approx \Delta\hat{\boldsymbol{M}}/\Delta q_i$ has to be calculated numerically, for example, with finite differences. Further, noting that all the observers we have discussed utilize the product $\hat{\boldsymbol{C}}^T$ and $\dot{\boldsymbol{q}}$ rather than $\hat{\boldsymbol{C}}^T$, it is possible to reduce computational costs by evaluating the product directly as $\dot{\hat{\boldsymbol{M}}}\dot{\boldsymbol{q}} - \hat{\boldsymbol{C}}\dot{\boldsymbol{q}}$ or

$$\hat{\boldsymbol{C}}^T\dot{\boldsymbol{q}} \approx \sum_{i=1}^{n}\frac{1}{\Delta q_i}(\boldsymbol{rnea}_0(\boldsymbol{q}+\Delta q_i,\boldsymbol{0},\dot{\boldsymbol{q}}) - \boldsymbol{p}_0) - \boldsymbol{rnea}_0(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{0}), \tag{37}$$

where $\boldsymbol{p}_0 = \boldsymbol{rnea}_0(\boldsymbol{q},\boldsymbol{0},\dot{\boldsymbol{q}})$ and n is the number of joints. From a computational point of view Equation (37) is cheaper than computing $\dot{\hat{\boldsymbol{M}}}$ and $\hat{\boldsymbol{C}}$ separately because it allows avoiding multiple calls of $\boldsymbol{rnea}_g(\cdot)$ for column-wise evaluation of the matrix $\dot{\hat{\boldsymbol{M}}}$.

Each observer has to be initialized when called for the first time; we suggest initializing based on the assumption that there is no collision, and disturbance torques are equal to zero. Moreover, each observer depends on its previous states that can be implicit as in the case of filters, or explicit and implemented via the integration of some parameters. Therefore, observer does not work as a "pure function" and cannot be used several times during a time instant.

3.1.1. NDOB

NDOB is among the easiest in terms of implementation, partially because it does not require $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})^T\dot{\boldsymbol{q}}$. The dynamics of the auxiliary variable (Equation 18) can be stiff for high convergence rates, therefore, to integrate it, we used the implicit Euler method. As for initialization, since the output is $\hat{\boldsymbol{\tau}}_d = \boldsymbol{z} + \boldsymbol{\psi}(\boldsymbol{q},\dot{\boldsymbol{q}})$, the initial value can be set to $\boldsymbol{z}(0) = -\boldsymbol{\psi}(\boldsymbol{q},\dot{\boldsymbol{q}})$. In terms of performance, NDOB was among the slowest (**Table 3**) because of the matrix inversion, to decrease computation time associated with it the properties of the inertia matrix have to be exploited.

3.1.2. Momentum Observer

The observer retains states of the integrator $\int_0^t\left(\boldsymbol{\tau} + \hat{\boldsymbol{C}}^T(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \hat{\boldsymbol{g}}(\boldsymbol{q}) + \hat{\boldsymbol{\tau}}_d\right)d\xi$, that can be initialized with zeros, but it is better to use $-p(0)$ to avoid subtraction during the next calls. The efficiency of the observer depends on the accuracy of integration; a comparison of different integration schemes showed that a simple trapezoidal rule is sufficient for high sampling rates.

3.1.3. Sliding Mode Momentum Observer

Both observers Equation (29) and Equation (30) can be implemented in single program code, as they only differ in coefficients $\boldsymbol{T}_2$, $\boldsymbol{S}_2$ that can be set to zero for second-order sliding mode observer. In our implementation, the dynamics of the observers is integrated using the explicit Euler method, and $sgn(\cdot)$ is replaced with the hyperbolic tangent to remove chattering. To initialize, $\hat{\boldsymbol{p}}(0)$ can be set to $\boldsymbol{p}(0)$.

3.1.4. Kalman Filter Observer

The method evaluates the variable $\boldsymbol{u}$ and calls the implementation of the discrete-time Kalman filter. The filter requires the user to define the discrete-time model of the system (Equation 28). If the sampling rate is constant, then the discrete-time model can be calculated from the original continuous-time model offline. However, if sampling time varies then online matrix exponent computation has to be used. For a high sampling rate, we approximated matrix exponential by the first terms of its Taylor series expansion.

3.1.5. Filtered Dynamics Observer

The implementation of the observer is based on infinite impulse response filters. In particular, $F(s)$ and $F_2(s)$ are implemented as $y_i = k_1 y_{i-1} + k_2(x_i + x_{i-1})$ where x and y are the input and output respectively, k_1 and k_2 are coefficients of the filters. Both coefficients depend on the cut-off frequency of the filter and can be found, for example, via bilinear transforms. For disturbance estimation, $F(s)$ can be initialized as $\hat{\boldsymbol{g}}(\boldsymbol{q}) - \hat{\boldsymbol{C}}^T\dot{\boldsymbol{q}} - \boldsymbol{\tau}$, while $F_2(s)$ as $\boldsymbol{p}$.

3.2. Identifying Dynamic Model

Universal Robots provides a description of the UR10e in the universal robot description file (URDF). The file contains kinematic parameters of the robot as well as inertial parameters of links but does not contain inertial parameters of the motors, drive gains, and friction parameters needed for accurate modeling of dynamics. Thus, we performed model identification, following the main steps from section 2.1. First, we symbolically derived regressor matrix $\boldsymbol{Y}(\cdot) \in \mathbb{R}^{6\times 84}$ using kinematic parameters from URDF and Euler-Lagrange formulation of dynamics. Then reduced parameter space using QR—decomposition obtaining $\boldsymbol{\pi}_b \in \mathbb{R}^{40}$ and $\boldsymbol{Y}_b(\cdot) \in \mathbb{R}^{6\times 40}$.

In experiment design, as a trajectory, we chose the combination of truncated Fourier series and fifth-order polynomials (Equation 4); as an objective function—condition number of the observation matrix. The optimization problem was posed in MATLAB and solved using the *patternsearch* algorithm. We performed trajectory optimization for different periods T (20, 30, 40, 50 s) and number of harmonics N (5, 8, 10, 12, 15); the best estimation in terms of root mean square error of torque prediction was obtained for $N = 12$ and $T = 30$ (**Figure 1**).

The optimized trajectory was executed on UR10e in the velocity control mode. There are several methods to program the robot to execute the desired trajectory: MATLAB, ROS, UR Script. We used UR Script because it allows sending commands to the robot with a higher sampling rate than other methods. Despite the trajectory tracking capabilities of the UR10e, the nominal and real trajectories do not always coincide because of safety limitations on velocities and accelerations. To overcome that either safety constraints can be removed, or maximum velocity and acceleration constraints can be decreased during the trajectory planning phase.

In the data processing stage currents, positions, and velocities were filtered with zero-phase fifth-order Butterworth filter (the *filtfilt* function in MATLAB). For acceleration estimation, we used numerical differentiation—central difference scheme—followed by zero-phase filtering to remove the noise.

Parameter estimation was divided into two parts: first, we identified drive gains on one trajectory ($T = 50$, $N = 14$), then on a different trajectory ($T = 30$, $N = 12$) the vector of base and standard parameters. In both cases, physical consistency was imposed by linear matrix inequality constraints using the YALMIP toolbox (Lofberg, 2004). Finally, the parameters were validated on a different harmonic trajectory (**Figure 2**) for which the root mean square error of torque prediction is $rms = \begin{bmatrix} 2.81 & 4.16 & 1.90 & 0.70 & 0.65 & 0.46 \end{bmatrix}$ Nm. The estimated parameters are shown in **Tables 1**, **2**.

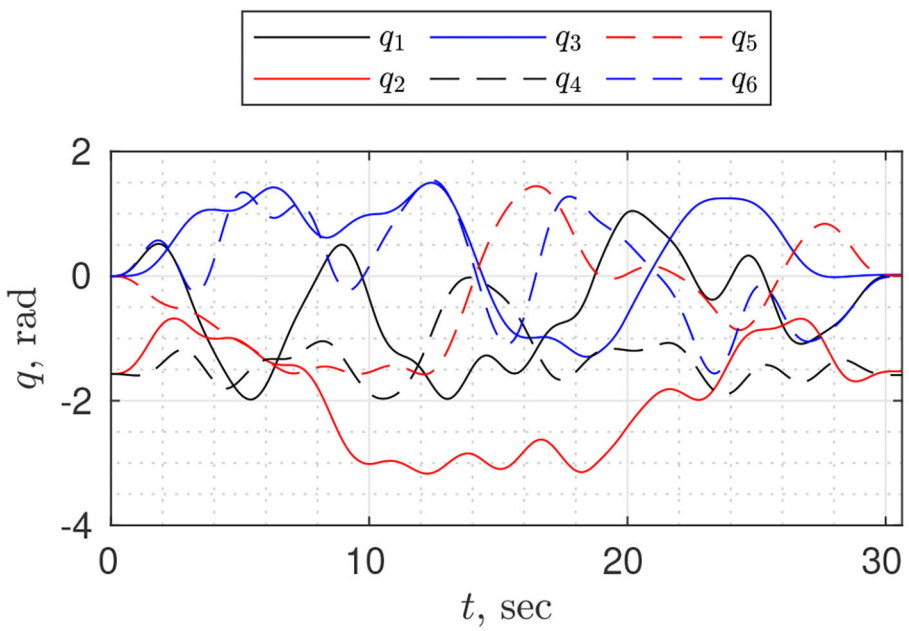

FIGURE 1 | Optimized trajectory for dynamic parameter identification, executed on the robot UR10e. The condition number of the base observation matrix is $cond(\boldsymbol{W}_b(\cdot)) \approx 93.27$.

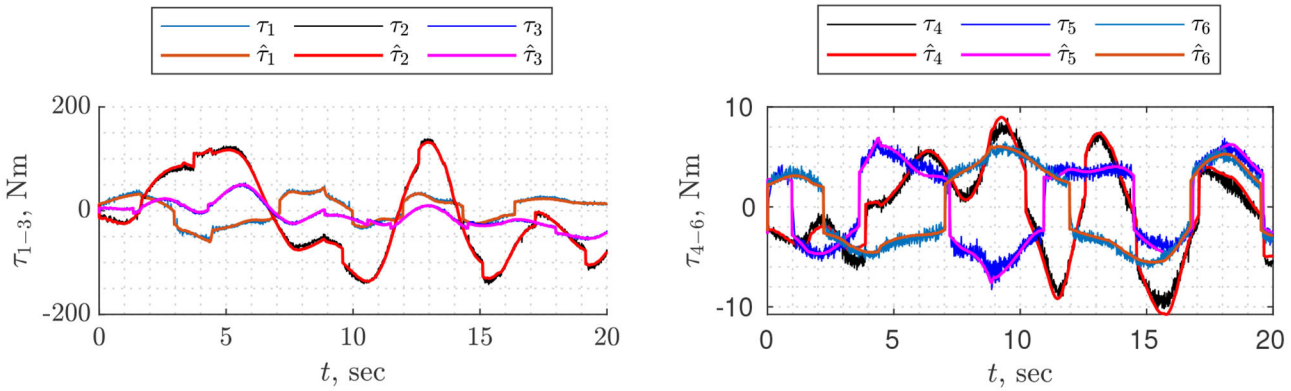

FIGURE 2 | Validation of identified dynamic parameters. τ_i is measured current multiplied by identified drive gains k_i, $\hat{\tau}_i$ estimated torque based on identified dynamic paramters.

TABLE 1 | Identified link parameters.

No	I_{xx} $kg \cdot m^2$	I_{xy} $kg \cdot m^2$	I_{xz} $kg \cdot m^2$	I_{yy} $kg \cdot m^2$	I_{yz} $kg \cdot m^2$	I_{zz} $kg \cdot m^2$	h_x $kg \cdot m$	h_y $kg \cdot m$	h_z $kg \cdot m$	m m
1	16.02	0	0	16.02	0	10^{-6}	0	0	0	4.83
2	3.93	0.85	0.15	3.72	−0.59	1.90	0.05	−0.28	3.26	7.93
3	2.22	−0.53	−0.18	1.39	−0.22	1.39	0.02	0.24	0.89	2.48
4	1.03	0.23	−0.01	0.08	0.07	1.07	-10^{-3}	−0.25	0.19	2.16
5	0.03	−0.03	−0.03	0.11	−0.01	0.11	0.01	0.09	0.11	2.16
6	0.03	-10^{-3}	−0.01	0.04	-10^{-3}	0.01	10^{-3}	-10^{-3}	−0.02	0.22

TABLE 2 | Identified motor and friction parameters.

No	K $N \cdot m \cdot A^{-1}$	J $kg \cdot m$	f_v $N \cdot s$	f_c $N \cdot m$	f_0 $N \cdot m$
1	10.0	0	21.25	12.54	0.20
2	10.70	3.74	20.22	13.27	−0.75
3	8.46	0	10.38	4.99	0.20
4	9.0	0.07	3.58	2.0	0.05
5	9.48	0.23	2.49	2.69	−0.01
6	10.12	0.44	3.03	2.30	0.04

3.3. Choosing and Tuning Observers

In this subsection we provide the general guideline for selecting and tuning the disturbances observer discussed in the section 2.2. All the observers in absence of collisions show non-zero disturbance torques because of the inaccurate estimates of some dynamic parameters or unmodeled dynamics. The closer identified parameters are to real ones, the smaller are going to be estimated disturbance torques and vice versa. It emphasizes the significance of the accurate dynamic model in estimating torques associated with collisions. For high level comparison of the observers see **Table 3**.

TABLE 3 | Comparison of observers in terms of computation time (average time needed to predict disturbance torques at a given time instant, laptop parameters: CPU Core i3 2.30GHz, RAM 4 GB) and tuning.

	Average time	Tuning	
	ms	Simplicity	Flexibility
Momentum observer	0.028	✓	✓
Nonlinear disturbance observer	0.146	✓	✓
Sliding mode observer	0.032	✗	✓
Kalman disturbance observer	0.294	✗	✓
Filtered dynamics	0.028	✓	✗

3.3.1. NDOB

To use NDOB, a user needs to specify gain matrix $\boldsymbol{L}$ and vector $\boldsymbol{\psi}$, both of which can be obtained from $\boldsymbol{Y} = \boldsymbol{X}^{-1}$ (Equation 19). There are two approaches to finding $\boldsymbol{Y}$: solving linear matrix inequality for a given positive definite symmetric matrix $\boldsymbol{\Gamma}$ (Equation 20); using analytical solution found for the case when $\boldsymbol{\Gamma}$ and $\boldsymbol{Y}$ are constrained to be scaled identity matrices (Equation 21). The latter is easier from a user perspective because it requires tuning single parameter β that defines minimum convergence rate whereas ξ and σ_2 are constants that depend on the dynamics of the robot. While tuning the NDOB, whether matrix $\boldsymbol{\Gamma}$ or scalar β, the user should seek a compromise between convergence rate and measurement noise amplification. **Figure 3** shows disturbance torque estimation for different values of β. For the second joint the lower is β the less noisy is $\hat{\tau}_{d,2}$, but for the fourth joint the value of β does not affect the amount of noise. It is the drawback of the analytical solution that all diagonal elements of $\boldsymbol{Y}$ are identical, thus the convergence rate of each joint cannot be tuned separately. Solving linear matrix inequality instead of using an analytical solution can serve as a remedy to this problem.

3.3.2. Momentum Observer

If $\boldsymbol{L}$ in Equation (25) is chosen as diagonal, then the classical momentum observer can be interpreted as n low-pass filters

driven by disturbance torque of each joint (**Figure 4**). Therefore, the user can conveniently design each filter independently. On the other hand, the momentum observer can be treated as a linear time-invariant system (Equation 28) where disturbance torques are non-measurable states. In that case, the classical state observer can be designed using pole placement or linear quadratic techniques (Åström and Murray, 2010). The former is easier because the user needs to specify poles i.e., 12 parameters but it is not optimal, while the latter is more difficult to design as the user needs to tune matrices $\boldsymbol{Q} \in \mathbb{R}^{12\times12}$ and $\boldsymbol{R} \in \mathbb{R}^{6\times6}$ but it is optimal.

If unmodeled dynamics is mainly due to friction modeling and can be described as zero-mean Gaussian white noise, then theoretically we can estimate $\boldsymbol{\tau}_{ext}$ by filtering out unmodeled dynamics using the Kalman filter. The price for accuracy is a time-consuming tuning procedure. To design the Kalman filter user needs to identify the diagonal covariance matrix of the unmodeled dynamics $\boldsymbol{Q}_{\tau_{um}}$ by individually moving each joint and calculating the variance of the $\boldsymbol{\tau}^i_{um}$, then the diagonal covariance matrix of output measurement $\boldsymbol{R}_p$ which is assumed to be mainly due to noise in velocity measurement. In turn, the covariance of the velocity measurement $\boldsymbol{R}_{\dot{q}}$ can be found by moving each joint at a constant velocity and finding the variance of the residual $\Delta\dot{q}_i = \dot{q}_i - \bar{\dot{q}}_i$. Then time-variant covariance matrix of the momentum can be computed by applying linear transformation properties of the normal $\boldsymbol{R}_p = \hat{\boldsymbol{M}}\boldsymbol{R}_{\dot{q}}\hat{\boldsymbol{M}}^T$. The only parameter that has to be tuned is the covariance matrix of the external torque dynamics. In the absence of any apriori information, it can be chosen the diagonal. The general guideline is that the larger diagonal elements are the less filter trusts the dynamics and the more amplifies noise. The interested reader is referred to Wahrburg et al. (2017) for more details. **Figure 5** shows $\hat{\tau}_d$ for three different values of tuning parameter, the estimates are much less noisy and slower even for high values of the tuning parameter compared with other observers. Moreover, it does not filter model noise, probably because the assumptions on the

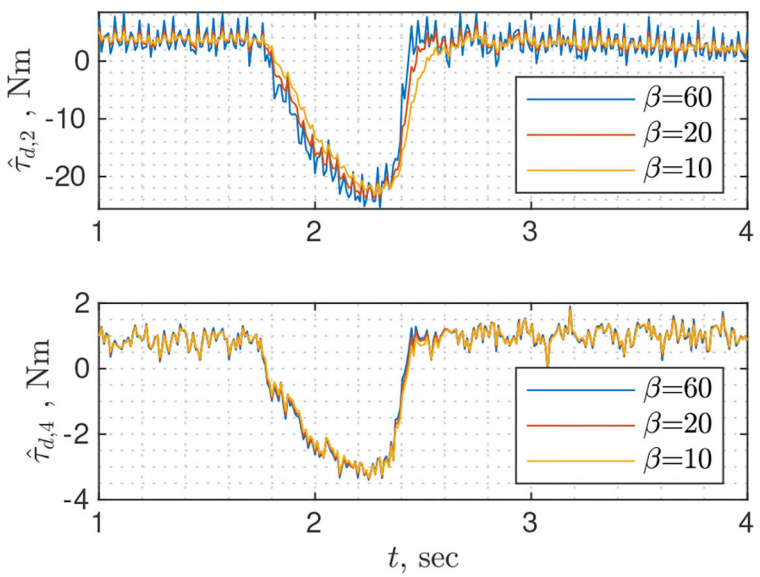

FIGURE 3 | Disturbance torque estimation for joints 2 and 4 using nonlinear disturbance observer with different rates of convergence β. The change in τ_d is due to collision with the wrist.

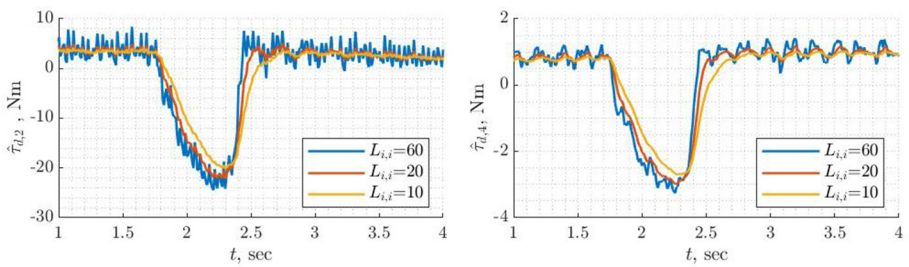

FIGURE 4 | Disturbance torque estimation for joints 2 and 4 using classical momentum observer with different $L(4, 4)$. The change in τd is due to collision with the wrist.

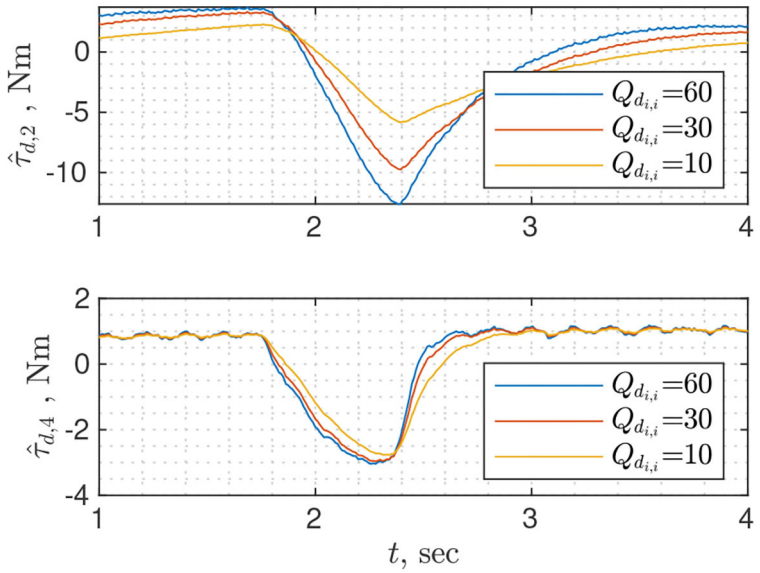

FIGURE 5 | Disturbance torque estimation for joints 2 and 4 using Kalman disturbance observer with different disturbance covariance matrices.

diagonal structure of the covariance matrix as well as the source of unmodeled dynamics are too strong.

3.3.3. Sliding Momentum Observer

The sliding mode observers are the most difficult to tune as they require tuning two matrices for second-order sliding mode observer and four matrices for second-order slide mode observer with linear terms matrices. Even if the matrices are restricted to be diagonal, the user needs to tune 12 and 24, parameters respectively for 6-DOF industrial manipulator. In order to guarantee the stability of the observers, the parameters should satisfy certain constraints (Moreno and Osorio, 2008). To obtain the desired behavior of the observer Garofalo et al. (2019) suggested considering the observer as a non-linear proportional-integral controller with all underlying tuning procedure. More specifically, they suggest tuning $\boldsymbol{S}$, then starting from $\boldsymbol{T} = 1.5\sqrt{\boldsymbol{S}}$ tune $\boldsymbol{T}$. Linear terms of the second orders sliding mode observer with linear terms can be interpreted as second-order linear filter that allows obtaining the desired damping. For a more detailed discussion of tuning sliding mode momentum observers, readers are referred to Garofalo et al. (2019).

3.3.4. Filtered Dynamics Observer

Probably the simplest observer in terms of tuning is the filtered dynamics observer (Equation 35) as it requires tuning the single parameter: cut-off frequency of the low-pass filter. It is identical to classical momentum observer when gain matrix $\boldsymbol{L}$ is a diagonal with identical entries along the diagonal. The drawback is the lack of flexibility e.g., a cut-off frequency can be well suited for the first joints but can be too high for the rest. **Figure 6** shows disturbance torque estimation for three different values of the cut-off frequency. Different behavior of $\hat{\tau}_{d,2}$ for higher ($\omega = 60$) and lower frequencies ($\omega = 10/20$) can be explained by high frequency dynamics, which was filtered during the model identification phase.

3.4. Detecting Collisions

For collision detection we did two experiments: in one experiment the robot experienced a soft collision with a human, while in another test—a hard collision with a brick. In both cases, manufacturer safety presets were set to the least restricted to achieve fast motions. Model-based estimated disturbance torques were compared with those estimated based on the real and target currents of the UR10e that were computed as

$$\hat{\tau}_{d,UR} = diag(K)(I_{target} - I_{real}), \tag{38}$$

where K is the identified vector of drive gain coefficients, I_{target} and I_{real} are the vectors of measured and target currents, respectively.

Due to the modeling error and torque/current measurements noise, the estimated disturbance torques do not equal zero during nominal operation conditions (without collision). To detect a collision it is necessary to set upper and lower thresholds on estimated disturbance torque in the absence of collisions $\hat{\boldsymbol{\tau}}_{d,ub}$ and $\hat{\boldsymbol{\tau}}_{d,lb}$, respectively. When a collision occurs, some of $\hat{\boldsymbol{\tau}}_{d,i}$ significantly increases depending on the location of a collision crossing corresponding $\hat{\tau}_{d,ub,i}$ or $\hat{\tau}_{d,lb,i}$. The static threshold is the easiest approach to detecting collisions; it can be chosen for a specific task or the whole workspace (any task). On the one side, task-specific bounds for a single or several tasks are tighter and result in faster collision detection time. On the other side, if a robot is reprogrammed to execute a new task, then the threshold can be violated in the absence of collision. Thus, for robots that are constantly reprogrammed to execute different tasks, global

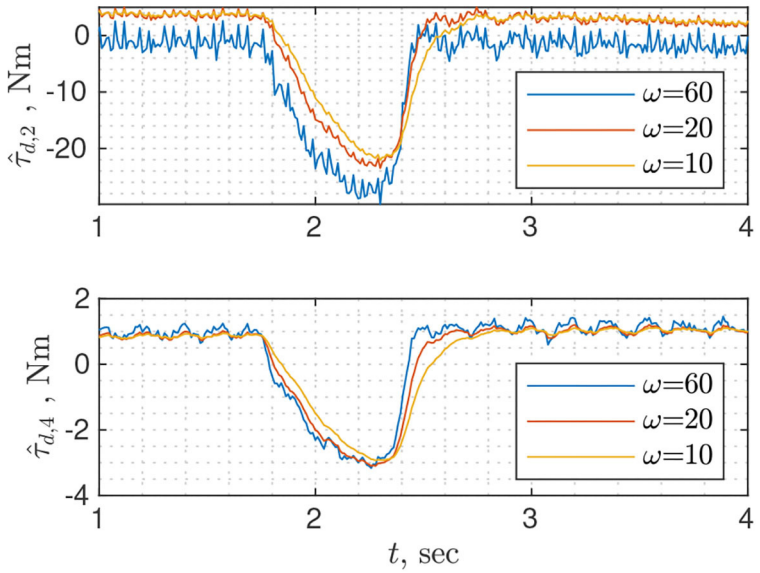

FIGURE 6 | Disturbance torque estimation for joints 2 and 4 using filtered dynamics observer with different cutoff frequencies.

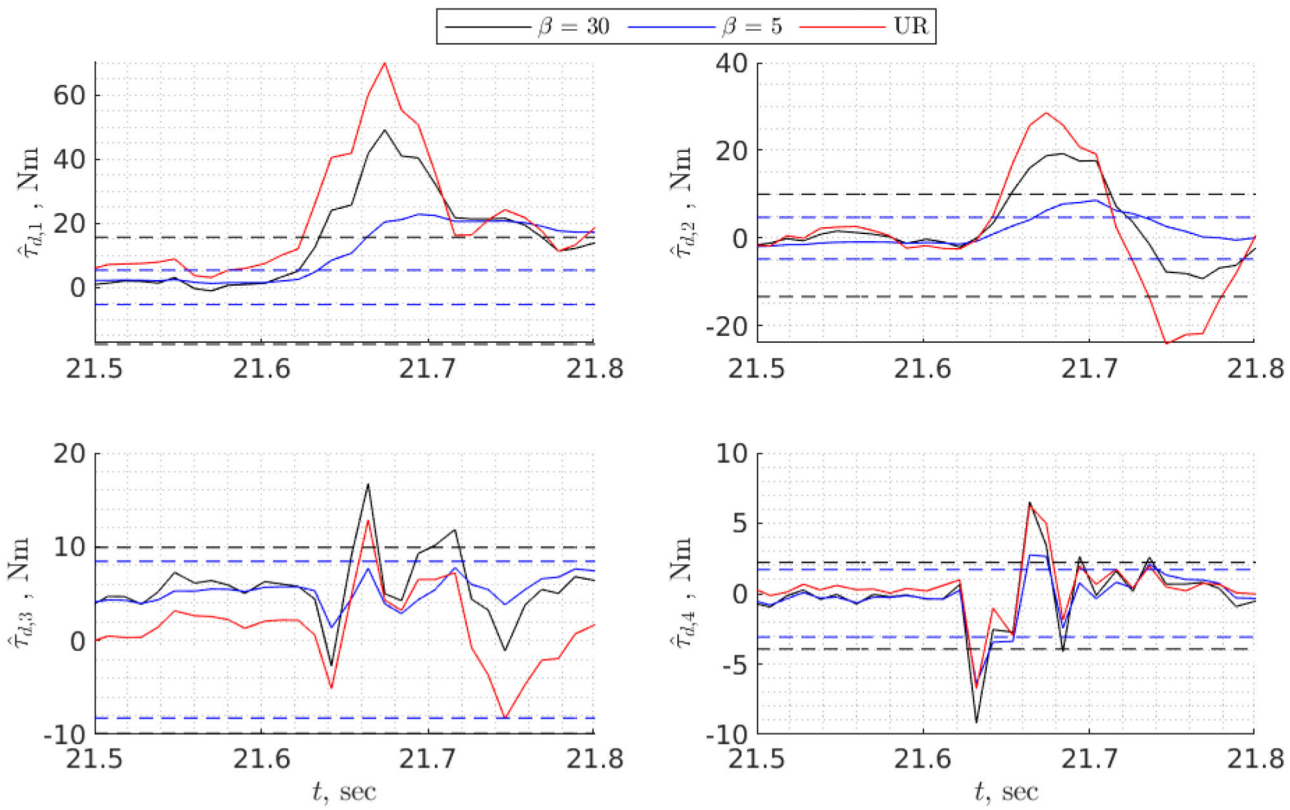

FIGURE 7 | Detecting a hard collision of the UR10e with a brick using static threshold. Thresholds and estimated disturbance torques are shown for two nonlinear disturbance observers with different rate of convergence. The third disturbance torque (in red) is estimated from current measurements of the UR10e. Dashed lines are thresholds associated with specific value of β.

bounds are more advantageous even though they can be looser and can result in higher detection time.

To find task-specific thresholds it is necessary to execute a task making sure there is no collision, then estimate disturbance torques and choose bounds that exceed maximum and minimum values of $\hat{\boldsymbol{\tau}}_{d,i}$ by 5–10% (**Figure 8**). For global bounds, it is possible to use data from validation or execute several trajectories that cover the whole workspace, then perform the procedure similar to the one for task-specific thresholds. In any case, the bounds will depend on the choice of disturbance observer, particularly on the convergence rate: the higher the convergence rate, the higher the bounds and vice versa (**Figure 8**). However, although for faster convergence rates the bounds are higher, the detection time is shorter, especially for fast collisions (**Figure 7**).

Both **Figures 7**, **8** show that collision detection capabilities of the UR10e and model-based collision detection algorithms with $\beta = 30$ were identical for hard collisions (**Table 4**), while for soft collisions UR10e was slightly faster (**Table 5**). One possible explanation is that the internal control loop of the UR10e is much faster than the rate at which we read generalized coordinates and velocities, and currents which are around 100 Hz. Another explanation is that the convergence rate of the disturbance observer inside the UR10e control system is higher than the convergence rate of the non-linear disturbance observer we use. The behavior of slower disturbance observer with $\beta = 5$ is different from the others; for example, in the case of hard collision of the third joint, estimated disturbance torque did not cross thresholds (**Figure 7**), but overall collision detection time was the same as for $\beta = 30$. In the case of soft collision, the observer with $\beta = 5$ was the fastest to detect collision (**Table 5**). In view of this result, the best strategy could be to use several disturbance observers with different convergence rates and static thresholds.

Theoretically, thresholds can be significantly reduced for fast collisions—collisions that have a distinguishingly higher frequency than the motion of robot—by a band-pass filtering, the dynamics of the robot (Ho and Song, 2013; Li et al., 2019). However, in our experiments by applying band-pass filter we were not able to reduce thresholds and to obtain faster collision detection time without getting false positives. A possible explanation is that the trajectory had abrupt changes in the sign of generalized velocities that caused fast changes in the friction torques, therefore the dynamics of the robot had high frequencies.

Another way to reduce the threshold is to use machine learning to predict unmodeled dynamics (Hu and Xiong, 2017). If enough data is provided for training, then neural networks can approximate unmodeled dynamics with a high degree of accuracy. The limitation of this method is that it requires knowledge of the field that engineers may not have, making it suitable for few who are familiar with neural networks.

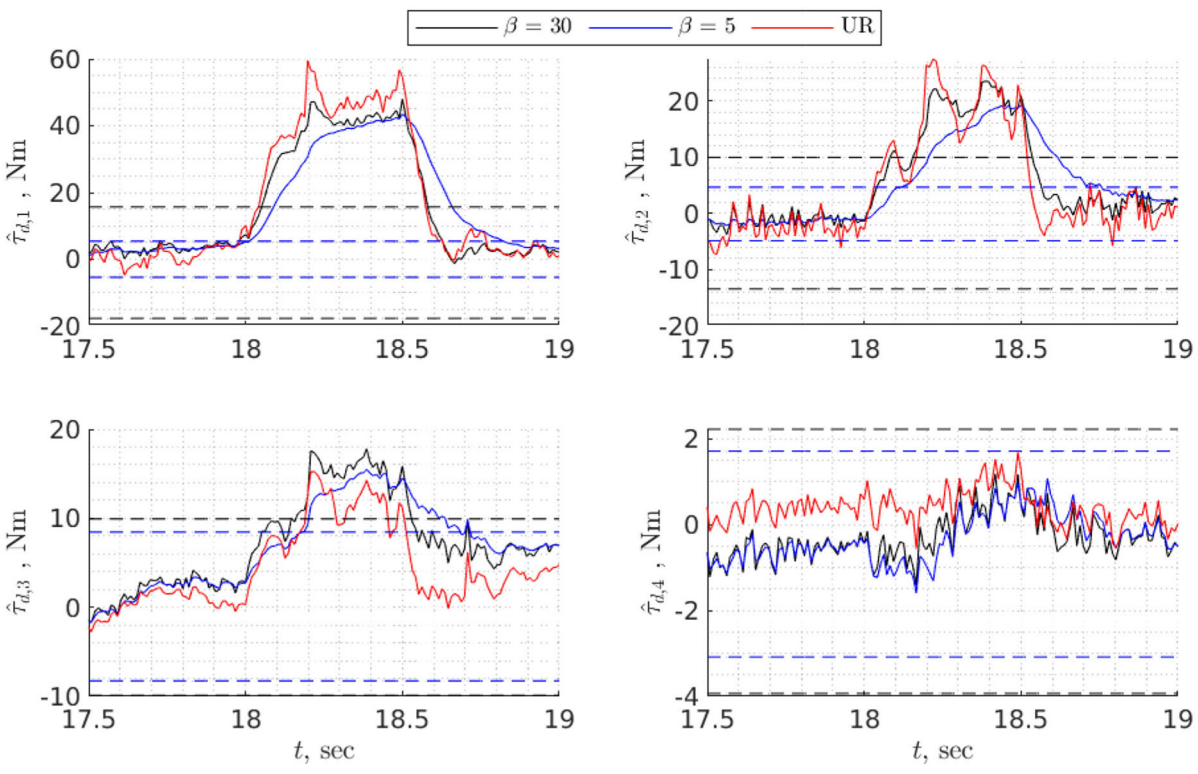

FIGURE 8 | Detecting a soft collision of the UR10e with a human using static threshold. Thresholds and estimated disturbance torques are shown for two nonlinear disturbance observers with different rates of convergence. The third disturbance torque (in red) is estimated from current measurements of the UR10e. Dashed lines are thresholds associated with specific value of β.

TABLE 4 | Identified link parameters.

Joint no	Collision detection time, *s*		
	$\beta = 30$	$\beta = 5$	UR
1	21.642	21.642	21.642
2	21.654	21.674	21.654
3	21.664	–	21.664
4	21.632	21.632	21.632
Overall	21.632	21.632	21.632

TABLE 5 | Identified motor and friction parameters.

Joint no	Collision detection time, *s*		
	$\beta = 30$	$\beta = 5$	UR
1	18.052	18.02	18.042
2	18.082	18.136	18.072
3	18.178	18.198	18.208
4	–	–	–
Overall	18.052	18.02	18.042

4. CONCLUSION

This paper has outlined a roadmap for engineers and specialists new to the field, to equip industrial manipulators with collision detection capabilities needed to guarantee workplace safety. The roadmap consists of three main steps: identifying the dynamic model, choosing and tuning the disturbance observer, and detecting collisions based on the output of the disturbance observer. The most important step is model identification because the accurate model allows choosing tighter thresholds, and consequently reducing collision detection time. Accurate dynamic parameter estimation might be time-consuming, especially its trajectory planning phase (depending on the trajectory and the number of parameters, trajectory optimization can take up to 10 h on a usual desktop computer), which is necessary to guarantee persistent excitation. At the cost of model accuracy, the trajectory planning phase can be notably simplified by omitting optimization, and instead use several points connected with fifth-order polynomials. The rest of the steps, unfortunately, cannot be further simplified.

For disturbance estimation, there is a great variety of disturbance observers. If simplicity is the priority, then non-linear disturbance observer or filtered dynamics observer can be chosen as they are tuned with single parameter β and ω, respectively. If more flexibility is required, then a classical momentum observer can be used because it allows prescribing the convergence rate of each joint separately. If better measurement and model noise filtering are required, then the Kalman momentum observer should be used. If finite time convergence is required, then second-order sliding mode observers should be chosen.

As the final step, it is necessary to choose thresholds for estimated disturbance torques in the absence of collisions, those can be chosen regardless of task type or on task-specific basis. Task-specific thresholds are tighter and result in faster detection time, while global bounds are not limited to any task but may result in slower detection time. In both cases, thresholds depend on the convergence rate of the disturbance observer, the faster the observer, the higher the threshold. In our experiments, faster disturbance observers were better at detecting hard collisions, while slower observers—at detecting soft collisions. It can suggest that the use of both slow and fast observer can lead to faster collision detection regardless of the type of collision. To further decrease collision detection time, several authors proposed ways to reduce threshold by using band-pass filtering, neural networks, while others proposed to use time-varying thresholds. Band-pass filtering works well with very fast collisions, depends on the dynamic model (friction model) and the trajectory of robot—in our experiment with a brick band-pass filtering did not decrease collision detection time—the use of neural networks requires a certain background that prevents engineers from using it. Finally, identifying coefficients of the time-varying thresholds involves time consuming joint-wise experiments and do not result in a significant reduction in collision detection time (Sotoudehnejad et al., 2012).

DATA AVAILABILITY STATEMENT

All datasets presented in this study are included in the article/**Supplementary Material**.

AUTHOR CONTRIBUTIONS

SMa did the literature review, dynamic identification and wrote the paper. SMi implemented disturbance observers library, did experiments with the robot and took part in writing the paper.

FUNDING

This work was supported by Russian Foundation for Basic Research (RFBR) grant 18-38-20186 and center for technologies in robotics & mechatronics components.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frobt.2020.571574/full#supplementary-material

REFERENCES

Åström, K. J., and Murray, R. M. (2010). *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton, NJ: Princeton University Press.

Anderson, B. D. (1982). "Exponential convergence and persistent excitation," in *1982 21st IEEE Conference on Decision and Control* (Orlando, FL: IEEE), 12–17.

Atkeson, C. G., An, C. H., and Hollerbach, J. M. (1986). Estimation of inertial parameters of manipulator loads and links. *Int. J. Robot. Res.* 5, 101–119. doi: 10.1177/027836498600500306

Briquet-Kerestedjian, N., Makarov, M., Grossard, M., and Rodriguez-Ayerbe, P. (2019). Performance quantification of observer-based robot impact detection

under modeling uncertainties. *Int. J. Intell. Robot. Appl.* 3, 207–220. doi: 10.1007/s41315-018-0068-4

Chen, W.-H., Ballance, D. J., Gawthrop, P. J., and O'Reilly, J. (2000). A nonlinear disturbance observer for robotic manipulators. *IEEE Trans. Indus. Electron.* 47, 932–938. doi: 10.1109/41.857974

De Luca, A., and Ferrajoli, L. (2009). "A modified newton-euler method for dynamic computations in robot fault detection and control," in *2009 IEEE International Conference on Robotics and Automation* (Kobe: IEEE), 3359–3364.

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

De Santis, A., Siciliano, B., De Luca, A., and Bicchi, A. (2008). An atlas of physical human–robot interaction. *Mech. Mach. Theor.* 43, 253–270. doi: 10.1016/j.mechmachtheory.2007.03.003

Fridman, L., and Levant, A. (2002). Higher order sliding modes. *Sliding Mode Control Eng.* 11, 53–102. doi: 10.1201/9780203910856.ch3

Garofalo, G., Mansfeld, N., Jankowski, J., and Ott, C. (2019). "Sliding mode momentum observers for estimation of external torques and joint acceleration," in *2019 International Conference on Robotics and Automation (ICRA)* (Montreal, QC: IEEE), 6117–6123.

Garofalo, G., Ott, C., and Albu-Schäffer, A. (2013). "On the closed form computation of the dynamic matrices and their differentiations," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Tokyo: IEEE), 2364–2359.

Gautier, M. (1991). Numerical calculation of the base inertial parameters of robots. *J. Robot. Syst.* 8, 485–506.

Gautier, M., and Briot, S. (2014). Global identification of joint drive gains and dynamic parameters of robots. *J. Dyn. Syst. Meas. Control* 136:051025. doi: 10.1115/1.4027506

Gautier, M., and Khalil, W. (1990). Direct calculation of minimum set of inertial parameters of serial robots. *IEEE Trans. Robot. Autom.* 6, 368–373.

Gaz, C., Magrini, E., and De Luca, A. (2018). A model-based residual approach for human-robot collaboration during manual polishing operations. *Mechatronics* 55, 234–247. doi: 10.1016/j.mechatronics.2018.02.014

Haddadin, S., Albu-Schaffer, A., De Luca, A., and Hirzinger, G. (2008). "Collision detection and reaction: a contribution to safe physical human-robot interaction," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Nice: IEEE), 3356–3363.

Haddadin, S. and Croft, E. (2016). "Physical human–robot interaction," in *Springer Handbook of Robotics*, eds B. Siciliano and O. Khatib (Cham: Springer), 1835–1874.

Haddadin, S., De Luca, A., and Albu-Schäffer, A. (2017). Robot collisions: a survey on detection, isolation, and identification. *IEEE Trans. Robot.* 33, 1292–1312. doi: 10.1109/TRO.2017.2723903

Ho, C.-N., and Song, J.-B. (2013). Collision detection algorithm robust to model uncertainty. *Int. J. Control Autom. Syst.* 11, 776–781. doi: 10.1007/s12555-012-0235-6

Hoffman, J. D., and Frankel, S. (2018). *Numerical Methods for Engineers and Scientists.* Boca Raton: CRC Press.

Hollerbach, J., Khalil, W., and Gautier, M. (2016). "Model identification," in *Springer Handbook of Robotics*, eds B. Siciliano and O. Khatib (Berlin; Heidelberg: Springer), 113–138.

Hu, J., and Xiong, R. (2017). Contact force estimation for robot manipulator using semiparametric model and disturbance kalman filter. *IEEE Trans. Indus. Electron.* 65, 3365–3375. doi: 10.1109/TIE.2017.2748056

Johnson, C. (1971). Accomodation of external disturbances in linear regulator and servomechanism problems. *IEEE Trans. Autom. Control* 16, 635–644.

Lewis, F. L., Dawson, D. M., and Abdallah, C. T. (2003). *Robot Manipulator Control: Theory and Practice*. Boca Raton: CRC Press.

Li, Z., Ye, J., and Wu, H. (2019). A virtual sensor for collision detection and distinction with conventional industrial robots. *Sensors* 19:2368. doi: 10.3390/s19102368

Lofberg, J. (2004). "Yalmip: a toolbox for modeling and optimization in matlab," in *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No. 04CH37508)* (New Orleans, LA: IEEE), 284–289.

Mikhel, S., Popov, D., Mamedov, S., and Klimchik, A. (2019). Development of typical collision reactions in combination with algorithms for external impacts identification. *IFAC Pap. Online* 52, 253–258. doi: 10.1016/j.ifacol.2019.11.177

Mitra, S. K., and Kuo, Y. (2006). *Digital Signal Processing: A Computer-Based Approach, Vol. 2*. New York, NY: McGraw-Hill.

Mohammadi, A., Tavakoli, M., Marquez, H. J., and Hashemzadeh, F. (2013). Nonlinear disturbance observer design for robotic manipulators. *Control Eng. Pract.* 21, 253–267. doi: 10.1016/j.conengprac.2012.10.008

Moreno, J. A., and Osorio, M. (2008). "A lyapunov approach to second-order sliding mode controllers and observers," in *2008 47th IEEE Conference on Decision and Control* (Cancun: IEEE), 2856–2861.

Nikoobin, A., and Haghighi, R. (2009). Lyapunov-based nonlinear disturbance observer for serial n-link robot manipulators. *J. Intell. Robot. Syst.* 55, 135–153. doi: 10.1007/s10846-008-9298-2

Oh, Y., and Chung, W. K. (1999). Disturbance-observer-based motion control of redundant manipulators using inertially decoupled dynamics. *IEEE ASME Trans. Mechatron.* 4, 133–146.

Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2010). *Robotics: Modelling, Planning and Control*. London: Springer Science & Business Media.

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

Sousa, C. D., and Cortesão, R. (2014). Physical feasibility of robot base inertial parameter identification: a linear matrix inequality approach. *Int. J. Robot. Res.* 33, 931–944. doi: 10.1177/0278364913514870

Sousa, C. D., and Cortesao, R. (2019). Inertia tensor properties in robot dynamics identification: a linear matrix inequality approach. *IEEE ASME Trans. Mechatron.* 24, 406–411. doi: 10.1109/TMECH.2019.2891177

Swevers, J., Ganseman, C., Bilgin Tükel, D., De Schutter, J., and Van Brüssel, H. (1997). Optimal robot excitation and identification. *IEEE Trans. Robot. Autom.* 13, 730–740.

Swevers, J., Verdonck, W., and De Schutter, J. (2007). Dynamic model identification for industrial robots. *IEEE Control Syst. Magaz.* 27, 58–71. doi: 10.1109/MCS.2007.904659

Takakura, S., Murakami, T., and Ohnishi, K. (1989). "An approach to collision detection and recovery motion in industrial robot," in *15th Annual Conference of IEEE Industrial Electronics Society* (Philadelphia, PA: IEEE), 421–426.

Van Damme, M., Beyl, P., Vanderborght, B., Grosu, V., Van Ham, R., Vanderniepen, I., et al. (2011). "Estimating robot end-effector force from noisy actuator torque measurements," in *2011 IEEE International Conference on Robotics and Automation* (Shanghai: IEEE), 1108–1113.

Wahrburg, A., Bös, J., Listmann, K. D., Dai, F., Matthias, B., and Ding, H. (2017). Motor-current-based estimation of cartesian contact forces and torques for robotic manipulators and its application to force control. *IEEE Trans. Autom. Sci. Eng.* 15, 879–886. doi: 10.1109/TASE.2017.2691136

Wensing, P. M., Kim, S., and Slotine, J.-J. E. (2017). Linear matrix inequalities for physically consistent inertial parameter identification: a statistical perspective on the mass distribution. *IEEE Robot. Autom. Lett.* 3, 60–67. doi: 10.1109/LRA.2017.2729659

Wu, J., Wang, J., and You, Z. (2010). An overview of dynamic parameter identification of robots. *Robot. Comput. Integr. Manuf.* 26, 414–419. doi: 10.1016/j.rcim.2010.03.013

Wu, W., Zhu, S., Wang, X., and Liu, H. (2012). Closed-loop dynamic parameter identification of robot manipulators using modified fourier series. *Int. J. Adv. Robot. Syst.* 9:29. doi: 10.5772/45818

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Mamedov and Mikhel. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.