Safety-oriented Robot Payload Identification using Collision-free Path Planning and Decoupling Motions

Saverio Farsoni[a,*], Federica Ferraguti[b], Marcello Bonfè[a]

[a]*Department of Engineering, University of Ferrara, Italy*
[b]*Department of Science and Methods for Engineering, University of Modena and Reggio Emilia, Italy*

Abstract

The paper presents a procedure for the identification of the inertial parameters of the payload mounted on the end-effector of a robotic manipulator operating in a collaborative setup. The procedure is therefore designed considering as a primary objective the safe execution of the payload estimation process, in terms of avoidance of physical objects and protected zones within the robot workspace, in which the presence of human operators is allowed and expected. The methods applied in the definition of the proposed operational sequence include a collision-free path planner and a joint trajectory generator that allows the safe execution of specific test motions. In particular, each one of such test motions decouples the effects of all but one inertial parameter of the payload, in order to improve the accuracy of identification. Experimental results on a redundant industrial manipulator demonstrate the feasibility of the proposed identification method.

Keywords: Payload estimation, Collaborative robotics, Collision-free path planning, Interaction control, Real-time Algorithms

1. Introduction

Recent perspectives on the future of the manufacturing industry give a strong emphasis on the collaboration between humans and robots, in terms

*Corresponding author
Email addresses: saverio.farsoni@unife.it (Saverio Farsoni), federica.ferraguti@unimore.it (Federica Ferraguti), marcello.bonfe@unife.it (Marcello Bonfè)

of both sharing a common space during working activities and direct physical interactions (i.e. physical Human-Robot Interaction, pHRI [1]), from manual guidance and teaching of a robotic task (e.g. skill-based instruction [2], lead-through programming [3]) to real co-working (e.g. assemblying or trasporting a workpiece together). Collaborative robots are already available on the market of industrial manipulators (e.g. KUKA LBR iiwa, Universal Robots UR3/5/10, ABB Yumi, FANUC CR-35iA etc.), even though the road to their widespread installation in daily operated workcells, including the presence of humans and allowing physical contacts and interactions, is paved with doubts. The most important issue related to industrially relevant uses of collaborative robots is certainly the safety of humans.

Needless to say, such an issue is covered by international standards and related government regulations, that have been refined in recent years to address specific aspects of human-robot collaboration. In particular, the parts 1 and 2 of the ISO 10218 [4, 5] standard specify requirements and guidelines on the safe operation of industrial robots, while providing definitions and examples of collaborative tasks. More recently, the technical specification ISO/TS 15066 [6] introduced specific safety requirements on collaborative industrial robotic systems, supplementing those given by the ISO 10218. The ISO 10218 standard and the ISO/TS 15066 specification define four collaborative operating modes: Safety-rated Monitored Stop (SMS), Hand Guiding (HG), Speed and Separation Monitoring (SSM), Power and Force Limiting (PFL). A brief and pictorial description of such operating modes can be found in [7] and in [8] with a focus on the implementation of SSM. From a control systems and engineering point of view, operating modes classified as HG and PFL imply significant design challenges, mainly related to the estimation of forces due to contacts or collisions between the robot and a human (i.e. to limit power/force exerted by the collision), which is even more challenging when intentional contacts (i.e. guiding the robot by hand) must be distinguished from unintentional ones.

Solutions for these challenges may be designed using joint torque sensors or joint motor currents and an accurate knowledge of robot dynamics. For example, model-based observers and the so-called residuals analysis for robot collision detection are presented in [9, 10]. When accurate knowledge of the dynamic parameters of the robot (i.e. masses, centers of mass and inertia tensors of each link, joint friction) is not available, alternative solutions may adopt frequency-domain analysis of motor currents, as described in [11], or explicitly include modeling uncertainties in the collision detection method,

by means of adaptive residual thresholds and filters [12]. Detection of human intents and their isolation from accidental collisions is reported, among others, in [10] and [13]. The latter proposes a control effort-based approach that should not require a dynamic model of the robot, but only the possibility for a human to manually change its end-effector position. This approach is reasonable for light and back-drivable haptic devices, like the one used in the experiments of [13], but questionable for large industrial robots, even if designed to collaborate with humans. Focusing on the HG mode, the installation of a force/torque (F/T) sensor on the end-effector of the robot and the implementation of force or impedence/admittance control schemes could be a viable option [14], provided that instabilities due to contact/non-contact transitions or environment/human stiffness are addressed [15, 16].

Using either model-based control schemes or schemes requiring an F/T sensor on the end-effector of the robot, the inertial properties (i.e. mass, center of mass, inertia tensor) of the robotic tool or payload introduce non-negligible effects. Indeed, impedance control resolved at joint torque level or model-based collision detection must consider the dynamics of the full robotic structure, while the implementation of an HG mode, based on measuring human inputs with wrist-mounted F/T sensors, requires proper elimination of non-contact effects caused by the payload during motion [17, 18, 19]. Other methods to accomplish the identification of payload inertial parameters that do not require a F/T sensor on the robot end-effector, exploit the observation of the joint torques (directly measured by sensors or estimated by means of motor currents) compared to the torques expected according to the dynamic model of the system [20]. However, such methods lead to less accurate results compared to those exploiting the F/T sensor. Moreover, while the inertial parameters of an industrial robot may be precisely calibrated by the manufacturer during robot design and assembly, this is not commonly true for tools realized by system integrators or heterogeneous parts grasped in an industrial plant. Therefore, a procedure for the identification of the robot payload properties, that could be executed online in an industrial setup involving human-robot collaborations (i.e. minimizing risks related to collisions and limiting velocity and accelerations of robot motions), would be a relevant solution for addressing safety-related issues in collaborative robotics.

The aim of the paper is to present a practical method for the estimation of the inertial parameters of a rigid body mounted on the end-effector of a robot, that takes into account the presence of obstacles, humans and other protected spaces in the robot operating environment. First, we as-

sume that the robot is equipped with a wrist-mounted F/T sensor or that its embedded controller provides an estimation of external forces/torques at the wrist. Then, we design the proposed method on the basis of a motion planner generating a sequence of trajectories designed to selectively isolate the effect on the wrist F/T measurements of one inertial parameter at a time, while simultaneously addressing worskpace constraints and safety-related velocity/acceleration limits in both cartesian and joint space. The contribution of the paper is therefore two-fold:

- the results of the proposed identification procedure allow the accurate setup of robot control algorithms providing safe execution of HG, SSM and PFL operating modes as defined by the ISO/TS 15066 specification;

- the safety of human operators is for the first time, to the best of our knowledge, taken into account during each phase of the identification procedure, whose execution does not require to put the robotic workcell in an offline mode and prevent the access of humans.

Moreover, in the context of an industrial application, the proposed identification procedure can be launched by the same person interacting with the robot during production as an automatic procedure, reducing the need for trained operators that usually carry out the cell configuration steps in manual operational mode.

The paper proceeds as follows: Sec. 2 describes the literature on inertial parameter identification for rigid bodies; Sec. 3 defines how to design test conditions and motions for a selective identification of a given inertial parameter at a time; Sec. 4 presents the set of functionalities to be designed and implemented for the execution of the proposed identification procedure; finally, Sec. 5 reports the results of practical experiments on a KUKA LWR 4+ redundant manipulator, equipped with an ATI Mini45 F/T sensor.

2. Related literature

The knowledge of the full set of inertial parameters (i.e. mass, center of mass (COM) and six elements of the inertia tensor 3×3 symmetric matrix) of a rigid body is of fundamental importance whenever precise motion control or estimation for that rigid body is required, not only in robotics. Indeed, the estimation of rigid bodies inertial parameters has been actively investigated

in most fields of the mechanical engineering literature and the computation of such parameters is a common feature of CAD software tools. However, it is known that computer-based estimation is not as reliable as experimental investigation, which is therefore considered the only effective way to address uncertainties in geometry, mass distribution, density of materials and so on. Moreover, it is also important to remark that while the identification of mass and center of mass can be done relatively easily with methods based on tests in static conditions, the identification of moments of inertia and inertia tensor is more difficult, since it requires tests enforcing complex and, possibly, highly dynamic motions of the rigid body under investigation (e.g. reverse symmetrical semi-program spherical motions and precessions [21]).

Surveys on experimental identification methods for inertial parameters can be found in [22, 23, 24]. In most cases, it is reported the use of specifically designed mechanical equipment, from torsional or trifilar suspension sytems [25, 26] to more complex platforms like the BTI [27] designed at the Federico II University of Naples, the Tensorobot [28] of the University of Kassel or the InTenso+/InTensino+ test rigs [29] of the Polytechnic University of Milan. Actual identification methods, focusing on those based on dynamic tests, can be classified into two main categories [23]: time-domain [25] and frequency-domain methods [30]. While the former class may be conceptually simpler (e.g. using pendulum-like oscillation period analysis), it may also require cumbersome mechanical setup and skilled operators. On the other hand, methods based on frequency response function (FRF) or modal analysis imply the use of advanced data acquisition systems (e.g. piezoelectric accelerometers with signal sampling at several KHz), that may not be available in daily industrial practice because of their cost, and complex computations on huge datasets, necessarily executed in post-processing. Another interesting aspect that has been investigated recently is the physical consistency of identification results [31, 32].

The applications, not only in robotics or control engineering, in which the experimental identification of inertial parameters is beneficial, if not mandatory, are many and heterogeneous. For example, parameters of spacecrafts and aircrafts are considered in [33, 34]. The importance of inertial properties of a road vehicle even for road accident reconstruction is highlighted in [35], while studies on motions of humans wearing prosthetic limbs are reported in [30]. Considering instead robotic applications, with a focus on industrial manipulators, an important issue that arises (and makes a key difference with previous examples) is the fact that rigid bodies of interest

cannot be detached, most of the time, from their mechanical assembly and mounted on a dedicated test equipment for identification purposes. Indeed, the majority of proposed methods for inertial parameter identification in robotics (see [36] for a review) consider the simultaneous identification of the properties of all mechanical links from joint torque sensors or joint motor currents, acquired during dynamic motion tests. Despite the large number of available references, from seminal works based on data-driven regression [37, 38] or neural networks [39] to more recent approaches in both directions [40, 41], the topic is still attractive from both theoretical (e.g. optimization of exciting trajectories [42, 43]) and practical (e.g. low-sampled data acquisition [44]) perspectives. Restricting further the interest on robot payload, its parameters identification can be viewed as an extension of the full robot dynamics identification [45] or as a specific task based on measurements from wrist-mounted F/T sensors [46, 47]. It is worth remarking that all of these methods assume that the range and the accelerations/velocities of either optimally designed or randomly generated test trajectories is only bounded by the physical capabilities of the robot. As a consequence, robot motions for payload estimation may be unfeasible when the robot is already installed in an industrial workcell, not to mention a collaborative one.

As discussed in the previous section, the payload estimation issue is also related to the safety of humans in collaborative applications, because of its impact on robot control and collision detection accuracy [48]. However, to the best of our knowledge the execution of payload identification procedures themselves, while collaborative modes of operation are active, has never been addressed previously. In this context, a first issue to be considered is the generation of robot motions taking into account the presence of humans as obstacles to be primarily avoided [49]. Collision-free motion planning for complex (e.g. redundant or multi-arm [50]) robotic systems is largely covered by the literature. Many works consider the use of so-called sampling-based algorithms, based on probabilistic approaches [51] or semi-random searches for collision-free and kinematically reachable paths [52]. From a practical point of view, a large collection of efficient motion planning algorithms are implemented by state-of-the-art software libraries, like *trajopt* [53] and OMPL [54]), embedding (or including as external parts, e.g. the FCL library [55]) collision checking functionalities for complex geometric objects. Another safety-related aspect to be considered in the generation of test motions for payload estimation is the limitation of velocities and accelerations, within bounds that are acceptable during human-robot coexistence. Such bounds

are specified in the operational (i.e. cartesian) space by the ISO 10218 and ISO/TS 15066 documents, but are then enforced at joint level. Interesting solutions for real-time generation of joint motion taking into account time-varying velocity/acceleration constraints have been proposed by [56, 57], being the latter specifically designed for redundant manipulators.

3. Decoupling motions for parameter identification

Forces $\mathbf{f}^s$ and torques $\boldsymbol{\tau}^s$ measured by a F/T sensor mounted on the robot end-effector can be expressed as a function of payload velocities and accelerations [46] as follows:

$$\begin{bmatrix} \mathbf{f}^s \\ \boldsymbol{\tau}^s \end{bmatrix} = \begin{bmatrix} m(\mathbf{a}^s - \mathbf{g}^s) + \dot{\boldsymbol{\omega}}^s \times m\mathbf{c}^s + \boldsymbol{\omega}^s \times (\boldsymbol{\omega}^s \times m\mathbf{c}^s) \\ \mathbf{I}^s\dot{\boldsymbol{\omega}}^s + \boldsymbol{\omega}^s \times \mathbf{I}^s\boldsymbol{\omega}^s + m\mathbf{c}^s \times \mathbf{a}^s - m\mathbf{c}^s \times \mathbf{g}^s \end{bmatrix} \tag{1}$$

Where the superscript s indicates that the variable refers to the F/T sensor frame, $\mathbf{g}$ is the gravity vector, m is the payload mass, $\mathbf{c}^s$ is the center of mass and $\mathbf{I}$ is the inertia tensor. Eq (1) can be reworked to highlight the linear dependency of $\mathbf{F}^s$ from the load inertial parameter vector $\boldsymbol{\Phi}^s = [m, mc_x^s, mc_y^s, mc_z^s, I_{xx}^s, I_{xy}^s, I_{xz}^s, I_{yy}^s, I_{yz}^s, I_{zz}^s]$:

$$\mathbf{F}^s = \mathbf{V}^s(\mathbf{a}_s, \boldsymbol{\omega}_s, \dot{\boldsymbol{\omega}}_s, \mathbf{g}_s)\boldsymbol{\Phi}^s \tag{2}$$

where $\mathbf{V}^s$ is a 6×10 matrix containing kinematic variables (linear accelerations $\mathbf{a}_s$, angular velocities $\boldsymbol{\omega}_s$ and angular accelerations $\dot{\boldsymbol{\omega}}_s$) expressed in the sensor frame. In the following the s superscript is omitted where obvious.

When the robot is motionless, Eq. (1) reduces to:

$$\begin{array}{rcrcl} F_x &=& -g_x m & & \\ F_y &=& -g_y m & & \\ F_z &=& -g_z m & & \\ \tau_x &=& -g_z mc_y^s &+& g_y mc_z^s \\ \tau_y &=& g_z mc_x^s &-& g_x mc_z^s \\ \tau_z &=& -g_y mc_x^s &+& g_x mc_y^s \end{array} \tag{3}$$

Such equations can be exploited to extract m and $m\mathbf{c}^s$ by performing a linear regression on the measured forces and torques in static condition. Note that if the gravity vector $\mathbf{g}$ is aligned with one of the sensor axis x^s,y^s,z^s the corresponding COM component is not excited and cannot be properly identified. Therefore, m can be formerly identified during the first motionless

phase, while mc^s could require more static phases to increase the identification accuracy.

We define the identification procedure for m and $m\mathbf{c}^s$ as a cyclic computation, described by Algorithms 1, 2:

Algorithm 1 Identification of m

1: $j = 0$
2: $\mathbf{Resp}_0 = [\quad]$
3: $\mathbf{Regr}_0 = [\quad]$
4: **for** $i = 1, \ldots, N_{\text{sample}}$ **do**
5: **if** $||\dot{\mathbf{x}}_i|| < \epsilon_s$ **then**
6: $\mathbf{Resp}_j = [\mathbf{Resp}_j \quad [F_x, F_y, F_z]^T]$
7: $\mathbf{Regr}_j = [\mathbf{Regr}_j \quad [g_x, g_y, g_z]^T]$
8: $m = \text{regress}(\mathbf{Regr}_j, \mathbf{Resp}_j)$
9: $j = j + 1$
10: **return** m ▷ The identified value

Algorithm 2 Identification of $m\mathbf{c}^s$

1: $j = 0$
2: $\mathbf{Resp}_0 = [\quad]$
3: $\mathbf{Regr}_0 = [\quad]$
4: **for** $i = 1, \ldots, N_{\text{sample}}$ **do**
5: **if** $||\dot{\mathbf{x}}_i|| < \epsilon_s$ **then**
6: $\mathbf{Resp}_j = [\mathbf{Resp}_j \quad [\tau_x, \tau_y, \tau_z]^T]$
7: $\mathbf{Regr}_j = [\mathbf{Regr}_j \quad \begin{bmatrix} 0 & -g_z & g_y \\ g_z & 0 & -g_x \\ -g_y & g_x & 0 \end{bmatrix}]$
8: $m\mathbf{c}^s = \text{regress}(\mathbf{Regr}_j, \mathbf{Resp}_j)$
9: $j = j + 1$
10: **return** $m\mathbf{c}^s$ ▷ The identified value

The function $\mathbf{\Psi} = \text{regress}(\mathbf{Regr}, \mathbf{Resp})$ performs the linear regression using a least-squares method [58] and returns the vector $\mathbf{\Psi}$ of regression coefficients in the linear model $\mathbf{Resp} = \mathbf{Regr} \cdot \mathbf{\Psi}$. In Algorithm 1,2 line 6, $\mathbf{Resp}_j$ is the incremental response vector to the regressor $\mathbf{Regr}_j$; at line 4 N_{sample} is the number of acquired samples; at line 5 $\dot{\mathbf{x}}_i$ is the vector of

Cartesian velocities and ϵ_s is a user defined static threshold, that has to be tuned taking into account the noise on velocities estimations, so that the condition $||\dot{\mathbf{x}}|| < \epsilon_s$ ensures that the system is motionless.

Furthermore, when x^s,y^s or z^s axis of the sensor frame is aligned with the gravity vector $\mathbf{g}$, a pure rotational movement with constant angular acceleration around that axis (without translations) can be generated, so that the expression of the torque acting on the rotating axis reduces to:

$$\begin{array}{ll} \tau_x = \dot{\omega}_x I_{xx} & \text{if } x^s \text{ is rotating} \\ \tau_y = \dot{\omega}_y I_{yy} & \text{if } y^s \text{ is rotating} \\ \tau_z = \dot{\omega}_z I_{zz} & \text{if } x^s \text{ is rotating} \end{array} \tag{4}$$

During such motions, if angular accelerations and torques are known with a proper degree of accuracy, a linear regression can be performed to identify the inertial parameters I_{xx},I_{yy} and I_{zz} as described by Algorithms 3, 4, 5:

Algorithm 3 Identification of I_{xx}

$j = 0$
$\mathbf{Resp}_0 = [\quad]$
$\mathbf{Regr}_0 = [\quad]$
for $i = 1, \ldots, N_{\text{sample}}$ **do**
    if $|\dot{\omega}_x| > \epsilon_1$ AND $|\dot{\omega}_y| < \epsilon_2$ AND $|\dot{\omega}_z| < \epsilon_2$ **then**
        $\mathbf{Resp}_j = [\mathbf{Resp}_j \quad \tau_x]$
        $\mathbf{Regr}_j = [\mathbf{Regr}_j \quad \dot{\omega}_x]$
        $I_{xx} = \text{regress}(\mathbf{Regr}_j, \mathbf{Resp}_j)$
        $j = j + 1$
return I_{xx} ▷ The identified value

Algorithm 4 Identification of I_{yy}

1: $j = 0$
2: $\mathbf{Resp}_0 = [\quad]$
3: $\mathbf{Regr}_0 = [\quad]$
4: **for** $i = 1, \ldots, N_{\text{sample}}$ **do**
5: **if** $|\dot{\omega}_y| > \epsilon_1$ AND $|\dot{\omega}_x| < \epsilon_2$ AND $|\dot{\omega}_z| < \epsilon_2$ **then**
6: $\mathbf{Resp}_j = [\mathbf{Resp}_j \quad \tau_y]$
7: $\mathbf{Regr}_j = [\mathbf{Regr}_j \quad \dot{\omega}_y]$
8: $I_{yy} = \text{regress}(\mathbf{Regr}_j, \mathbf{Resp}_j)$
9: $j = j + 1$
10: **return** I_{yy} ▷ The identified value

Algorithm 5 Identification of I_{zz}

1: $j = 0$
2: $\mathbf{Resp}_0 = [\quad]$
3: $\mathbf{Regr}_0 = [\quad]$
4: **for** $i = 1, \ldots, N_{\text{sample}}$ **do**
5: **if** $|\dot{\omega}_z| > \epsilon_1$ AND $|\dot{\omega}_y| < \epsilon_2$ AND $|\dot{\omega}_x| < \epsilon_2$ **then**
6: $\mathbf{Resp}_j = [\mathbf{Resp}_j \quad \tau_z]$
7: $\mathbf{Regr}_j = [\mathbf{Regr}_j \quad \dot{\omega}_z]$
8: $I_{zz} = \text{regress}(\mathbf{Regr}_j, \mathbf{Resp}_j)$
9: $j = j + 1$
10: **return** I_{zz} ▷ The identified value

In Algorithms 3,4,5 at line 5, ϵ_1,ϵ_2 with $\epsilon_1 > \epsilon_2$ are user-defined thresholds that has to be tuned taking into account the noise that affects the estimation on kinematic derivatives. The condition $|\dot{\omega}_x| > \epsilon_1$ AND $|\dot{\omega}_y| < \epsilon_2$ AND $|\dot{\omega}_z| < \epsilon_2$ has to ensure that during the identification of I_{xx} the rotational movement is only performed on the x^s axis. Similar considerations apply to I_{yy} and I_{zz}.

The remaining parameters I_{xy},I_{xz},I_{yz} are not considered in this work, because in a typical industrial application the center of mass of robot tool and payload has a commonly dominant component along the approach-axis (z^s). This fact makes negligible the elements of the inertia tensor outside the diagonal (I_{xy},I_{xz},I_{yz}) with respect to the diagonal elements I_{xx},I_{yy},I_{zz}. Indeed, many industrial robot controllers, such as Kuka Sunrise do not allow the

configuration of the payload inertia tensor completely, but only its diagonal elements. However, the proposed approach could be extended to identify off-diagonal terms of the inertia tensor, even though such extension would require to specify more complex and composite movements.

Compensation of F/T sensor offsets

The raw measurement $\mathbf{F}_{\text{raw}}$,$\boldsymbol{\tau}_{\text{raw}}$ of the F/T sensor includes an undesired temperature-dependent component $\mathbf{F}_o = [F_{o_x}, F_{o_y}, F_{o_z}]^T$, $\boldsymbol{\tau}_o = [\tau_{o_x}, \tau_{o_y}, \tau_{o_z}]$, typical of strain gauge devices:

$$\begin{aligned} \mathbf{F}_{\text{raw}} &= \mathbf{F} + \mathbf{F}_o \\ \boldsymbol{\tau}_{\text{raw}} &= \boldsymbol{\tau} + \boldsymbol{\tau}_o \end{aligned} \tag{5}$$

Assuming that offsets do not significantly vary during the experiments, they can be estimated by means of an initial calibration procedure that consists of:

- Move the robot along a collision free path from the current pose to a target pose in which the x^s axis of the sensor frame is aligned to the gravity vector.
- Stop the motion for an adequate time and log the sensor raw measurements for a given time period.
- Keep the mean value $\bar{F}^+_{\text{raw}_x}$,$\bar{\tau}^+_{\text{raw}_x}$ of the logged measurements.
- Move the robot along a collision free path from the current pose to a target pose in which the x^s axis of the sensor frame is opposed to the gravity vector.
- Stop the motion for an adequate time and log the sensor raw measurements for a given time period.
- Keep the mean value $\bar{F}^-_{\text{raw}_x}$,$\bar{\tau}^-_{\text{raw}_x}$ of the logged measurements.
- Compute the offset component as $F_{o_x} = \frac{\bar{F}^+_{\text{raw}_x} + \bar{F}^-_{\text{raw}_x}}{2}$,$\tau_{o_x} = \frac{\bar{\tau}^+_{\text{raw}_x} + \bar{\tau}^-_{\text{raw}_x}}{2}$
- Repeat the procedure for y^s and z^s axis.

After this calibration procedure, the sensor raw acquisitions can be corrected by subtracting the estimated offsets.

4. Methods

The three identification movements involve a task that is defined in terms of Cartesian angular velocities expressed in the sensor frame. Because the ISO standards impose the maximum value on linear velocities, the ramp on the angular task velocities has to be properly designed to not exceed these limits. If the robot payload can be contained into a box of known dimensions the maximum admissible value of angular velocity can be computed as the ratio between the maximum linear velocity and the maximum distance of the sensor frame from the extreme point of the box.

Therefore, the starting pose can be chosen taking into account safety and practical considerations. In this work the starting pose is computed by a grid-based optimization algorithm that maximizes the distance of the robot link and payload from the obstacles and the protected area, while ensuring an high manipulability to avoid singular configurations during the motion. After reaching the starting pose the robot has to be properly controlled to perform the identification movement without exceeding the limits on joint velocities and accelerations. As proposed in [57], a redundant robot can efficiently accomplish a task expressed in Cartesian velocities domain exploiting its null space and taking into account hard constraints as limited joint range or bounded velocities. In the remainder of the section the grid-based search of optimal starting configuration and the Saturation in the Null Space algorithm (SNS) are presented.

Grid-based search of optimal starting configuration

The robot starts the identification movements from a configuration that can be optimized with reference to a customized objective index. In this work the adopted indices are:

- **Minimum value of the normalized link-obstacle distance vector**, where the normalization is computed by means of the robot workspace diameter;
- **Manipulability index**, computed as the ratio between the minimum and the maximum singular value of the jacobian matrix $\mathbf{J}$.

Therefore the objective function can be expressed as:

$$f_o(\mathbf{q}) = w_d \min(\mathbf{D}) + w_m \frac{\min(\text{svd}(\mathbf{J}(\mathbf{q})))}{\max(\text{svd}(\mathbf{J}(\mathbf{q})))} \tag{6}$$

Where svd($\mathbf{J}$) returns a vector containing the singular values of the matrix $\mathbf{J}$, min and max operators extract respectively the minimum and the maximum value of an input vector. $\mathbf{D}$ is the normalized vector of minimum link-obstacle distances.

The whole robot workspace is sampled into a three-dimensional grid of N_v vertices. For each vertex, the objective function is computed for a number of N_c joint configurations and the optimum configuration is maintained. Then, the result can be refined generating a second grid, smaller than the previous one and centered in the selected vertex. The same iterative procedure is repeated for the second grid and the optimum configuration $\mathbf{q}_{\text{opt}}$ is finally selected as starting configuration for the identification movement. The pseudo-code of Algorithm 6 describes the iterative double-grid procedure to generate the optimal starting configuration for the identification movement.

Algorithm 6 The double grid-based search of the optimal starting configuration

```
1: procedure getStartConfiguration(axis)                ▷ axis = x, y, z
2:     ojective = −1
3:     q_opt = null
4:     v* = null
5:     V_1 = Cartesian workspace grid
6:     for v in V_1 do
7:         for i = 1, ..., N_c do
8:             q = getConfigForPose(v)
9:             obj = f(q)
10:            if obj > ojective then
11:                objective = obj
12:                q_opt = q
13:                v* = v
14:    V_2 = smaller grid centered inv*
15:    for v in V_2 do
16:        for i = 1, ..., N_c do
17:            q = getConfigForPose(v)
18:            obj = f(q)
19:            if obj > ojective then
20:                objective = obj
21:                q_opt = q
22:    return q_opt                          ▷ The optimal starting configuration
```

Where $\mathbf{v} = v_x, v_y, v_z$ is a vertex of the selected grid, the function getConfigForPose($\mathbf{v}$) returns a random collision-free configuration that solves the inverse kinematic problem with reference to the sensor frame. Fig. 1 highlights the first grid containing the robot workspace and the smaller grid where the optimal configuration search has been refined.

Note that the sensor frame in the desired vertex position has the to be oriented so that the selected axis is parallel to the gravity vector. Then, the remaining axes can be oriented taking into account the operator safety conditions. Indeed, a protected zone, in which an operator can freely move, is often defined in a robot workcell. However, even if the robot is outside the protected area, accidental break of power supply can cause the detachment of the robot payload. Therefore, assuming that the identification movement

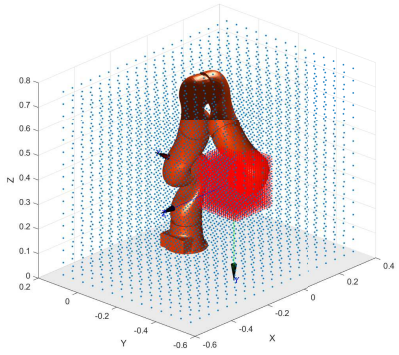

Figure 1: The double grid and the optimal robot starting configuration for the identification movement.

consists of a partial rotation along one axis, other axes can be at first oriented so that in case of detachment the load is thrown outside the protected area. This can be realized choosing, if possible, the z-axis of payload to start the identification motion orthogonal to the plane that represents the boundary of the protected area.

Saturation in the null space

The saturation in the null space (SNS) algorithm maps a desired Cartesian task $\dot{\mathbf{x}}_{\text{task}}$ at the current robot configuration $\mathbf{q}$ into joint target velocities $\dot{\mathbf{q}}_{\text{ref}}$ ensuring to not exceed the specified limits. SNS starts computing the minimum velocity norm solution by means of the pseudoinverse of the task Jacobian $\mathbf{J}(\mathbf{q})$, if it does not represent a feasible solution, one or more joints can be saturated to limit exceeding velocities while maintaining the desired target task. If joint saturation is still insufficient, the task is finally scaled. The basic version of SNS and the procedure to get the task scaling factor and the most critical joint (i.e. the joint whose velocity is the closest to the limits) are described in form of pseudo-code in Algorithm 7 and Algorithm 8, respectively.

Algorithm 7 Saturation in the null space algorithm, the main loop

procedure SNS
  $\mathbf{W} = -1$, $\dot{\mathbf{q}}_N = \mathbf{0}$,$s = 1$, $s^* = 0$,limit = FALSE
  while not limit **do**
    $\dot{\mathbf{q}}_{\text{ref}} = \dot{\mathbf{q}}_N + \text{pinv}(\mathbf{JW})(\dot{\mathbf{x}}_{\text{task}} - \mathbf{J}\dot{\mathbf{q}}_N)$
    if $\exists q_i \in \dot{\mathbf{q}}_{\text{ref}} : q_i > Q_{i,\max}$ OR $q_i < Q_{i,\min}$ **then**
      limit = TRUE
      $\mathbf{a} = \text{pinv}(\mathbf{JW})\dot{\mathbf{x}}$
      $\mathbf{b} = \dot{\mathbf{q}}_{\text{ref}} - \mathbf{a}$
      $[s_{\text{tmp}}, j_c] = \text{getScaleAndCriticalJoint}(\mathbf{a}, \mathbf{b})$
      if $s_{\text{tmp}} > s^*$ **then**
        $s^* = s_{\text{tmp}}$
        $\mathbf{W}^* = \mathbf{W}$
        $\dot{\mathbf{q}}_N^* = \dot{\mathbf{q}}_N$
      $W_{j_c j_c} = 0$
      if $\dot{q}_{j_c} > \dot{Q}_{\max}$ **then**
        $\dot{q}_{N,j_c} = \dot{Q}_{\max}$
      else
        if $\dot{q}_{j_c} < \dot{Q}_{\min}$ **then**
          $\dot{q}_{N,j_c} = \dot{Q}_{\min}$
      if $\text{rank}(\mathbf{JW}) < 6$ **then**
        $s = s^*$, $\mathbf{W} = \mathbf{W}^*$, $\dot{\mathbf{q}}_N = \dot{\mathbf{q}}_N^*$
        $\dot{\mathbf{q}}_{\text{ref}} = \dot{\mathbf{q}}_N + \text{pinv}(\mathbf{JW})(s\dot{\mathbf{x}}_{\text{task}} - \mathbf{J}\dot{\mathbf{q}}_N)$
        limit = FALSE
  return $\dot{\mathbf{q}}_{\text{ref}}$ ▷ The target joint velocities

Algorithm 8 Saturation in the null space algorithm, get the task scaling factor and the most critical joint

procedure GETSCALEANDCRITICALJOINT($\mathbf{a}, \mathbf{b}$)
  for $i = 1, n$ **do**
    $S_{\min,i} = (\dot{Q}_{\min,i} - b_i)/a_i$
    $S_{\max,i} = (\dot{Q}_{\max,i} - b_i)/a_i$
    if $S_{\min,i} > S_{\max,i}$ **then**
      $S_{\min,i} = S_{\max,i}$
      $S_{\max,i} = S_{\min,i}$
  $s_{\max} = \min_i(S_{\max,i})$
  $s_{\min} = \max_i(S_{\min,i})$
  $j_c = \text{argmin}_i(S_{\max,i})$
  if $s_{\min} > s_{\max}$ OR $s_{\max} < 0$ OR $s_{\min} > 1$ **then**
    $s = 0$
  else
    $s = s_{\max}$
  return s,j_c ▷ The task scaling factor and the most critical joint

In Algorithm 7 pinv$(\cdot)$ is the pseudoinverse operator, $\mathbf{W}$ is a diagonal matrix with 0/1 elements, if W_{ii} is null then the velocity of the joint j_i is saturated. While in Algorithm 8 $s \leq 1$ is the task scaling factor, j_c is the most critical joint and $\dot{\mathbf{q}}_N$ is the null space component of the joint target velocities that contains the contribution of the saturated joints.

Note that the specified limits on joint positions $\mathbf{Q}_{\min} < \mathbf{q} < \mathbf{Q}_{\max}$, velocities $\mathbf{V}_{\min} < \dot{\mathbf{q}} < \mathbf{V}_{\max}$ and accelerations $\mathbf{A}_{\min} < \ddot{\mathbf{q}} < \mathbf{A}_{\max}$ are merged together and mapped into velocity constraints taking into account the sampling time T_s as follows:

$$\begin{aligned} \dot{Q}_{\min,i} &= \max(\tfrac{Q_{\min,i} - q_i}{T_s}, -V_{\max,i}, -\sqrt{2A_{\max,i}(q_i - Q_{\min,i})}) \\ \dot{Q}_{\max,i} &= \min(\tfrac{Q_{\max,i} - q_i}{T_s}, V_{\max,i}, -\sqrt{2A_{\max,i}(Q_{\max,i} - q_i)}) \end{aligned} \tag{7}$$

Identification of load inertial parameters

The overall identification procedure consists of two robot movements for each of the three axes of the F/T sensor. First, the robot has to be moved along a collision-free path from its current configuration towards the optimal configuration computed by means of the grid-based procedure of Algorithm 6. The collision-avoidance motion can be planned by means of the Open

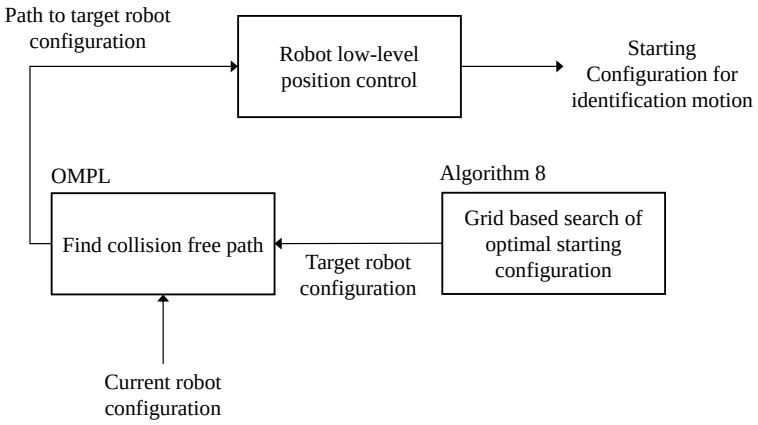

Figure 2: The block diagram of the motion to the optimal starting configuration for the identification motion.

Motion Planning Library (OMPL) framework [54] involving the usage of RRT-Connect or RRT-Star algorithm in the joint space. The block diagram of Fig. 2 shows how the motion to the optimal starting configuration is accomplished.

Once the starting configuration is reached, the robot has to be controlled to perform the identification movement. As discussed in Section 3, the desired task $\mathbf{x}_{\text{task}}$ is expressed in the Cartesian angular velocities domain and consists in one or more trapezoidal signals generating a constant acceleration motion. The task is mapped into joint target velocities by means of the SNS algorithm and then integrated to provide the reference signal to the robot low-level position controller. During the motion the current joint positions are used to compute the next joint target positions (SNS) and check if the minimum distance between robot, payload and obstacles are exceeding a certain safe threshold and, in case, stop the robot and restart the procedure.

In the meanwhile, the F/T sensor measurements are acquired and linear accelerations, angular velocities and acceleration expressed in the F/T sensor frame are on-line estimated by means of the Kalman filter of [19] so that the

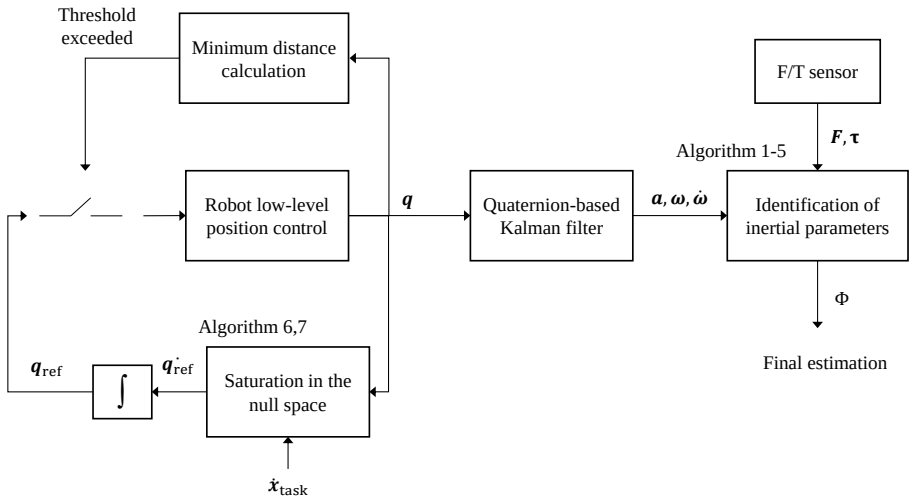

Figure 3: The block diagram of the procedure for the identification of inertial parameters.

identification regression can be performed and the inertial parameters can be updated. The block diagram of Fig. 3 represents the proposed identification procedure.

The overall identification procedure can be summarized as follows:

- Motion to start configuration, with x^s aligned with the gravity vector:
  1. Grid-based search of optimal starting configuration (Algorithm 6);
  2. Generation of collision avoidance path to starting configuration;
  3. Execute motion;
- Identification motion for I_{xx}
  1. Command the robot using the target computed by means of SNS (Algorithm 7);
  2. Monitor robot configuration to avoid collision;
  3. Compute Cartesian velocities and acceleration of the payload;
  4. Acquire F/T measurements;
  5. Identify the inertial parameters (Algorithm 1,2,3).
- Repeat the procedure for y^s and z^s axis.

5. Experimental results

The proposed identification scheme has been implemented using the Orocos real-time framework and tested on a 7-DOF KUKA LWR 4+ (workspace radius 800 mm, rated payload 7 kg), equipped with a 6-DOF F/T ATI Mini45 IP65 (measuring range ± 145 N on X-Y, ± 290 N on Z, ± 5 Nm on all axes, resolution 1/16 N - 1/752 Nm). The planning of robot motions has been implemented using V-REP [59] by Coppelia Robotics, a robotic simulation framework whose main features are:

- Robot models: 3D and kinematic models of several manipulators can be included in a simulation.
- 3D import: STL and other common 3D mesh file formats can be imported to create a realistic scene of the workcell.
- Scripting: Lua and C++ regular APIs can be used to create customizable scripts associated to the scene objects.
- Path planning: OMPL are wrapped into a dedicated plugin.
- Minimum distance calculation: distances between any scene object can be efficiently computed.
- Remote Control: thanks to the Remote API framework the simulation can establish a client-server TCP/IP socket to interact with an external application as a Matlab script or a Java program or directly control a remote hardware (i.e. the robot).

The experimental workcell has been reconstructed in V-REP including virtual obstacle planes representing the obstacles and the limit of the protected area. Fig. 4 shows the simulated scene with the robot at the cell home-position (A) and the optimal configurations for the identification movements (B,C,D) computed by means of the grid-based search.

The proposed method has been tested by using two different payloads whose inertial description has been CAD computing and used as the reference for the identification of inertial parameters. The payloads adopted in the experiments are an asymmetric steel block and an aluminum cylinder.

The Cartesian velocity task for each of the three robot movements consist of two opposed trapezoidal signals with accelerations equal to 4.8 rad/s^2

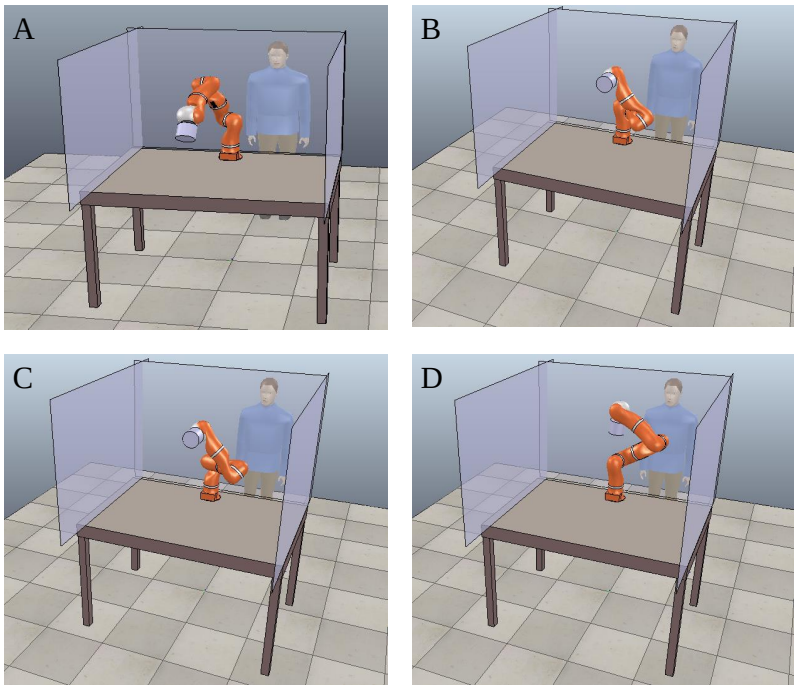

Figure 4: A: home-position; B,C,D robot optimal configurations for the identification movements.

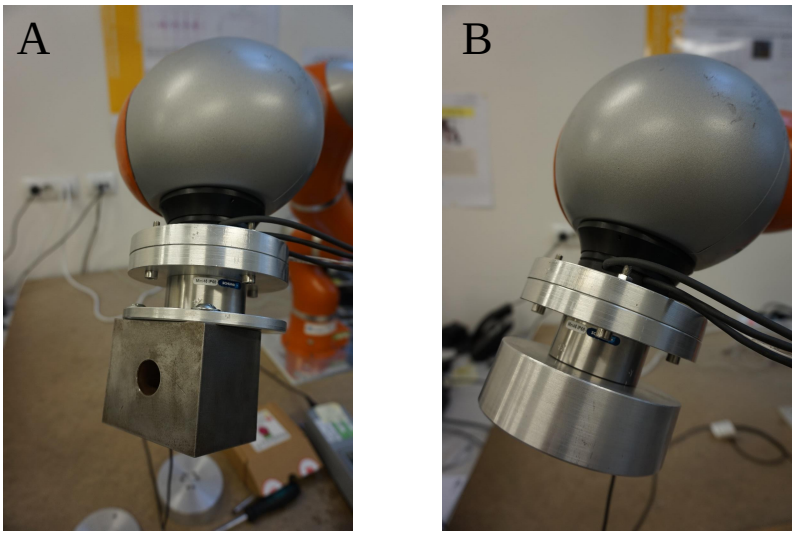

Figure 5: Payloads used in the experiments. A: steel block; B: cylinder.

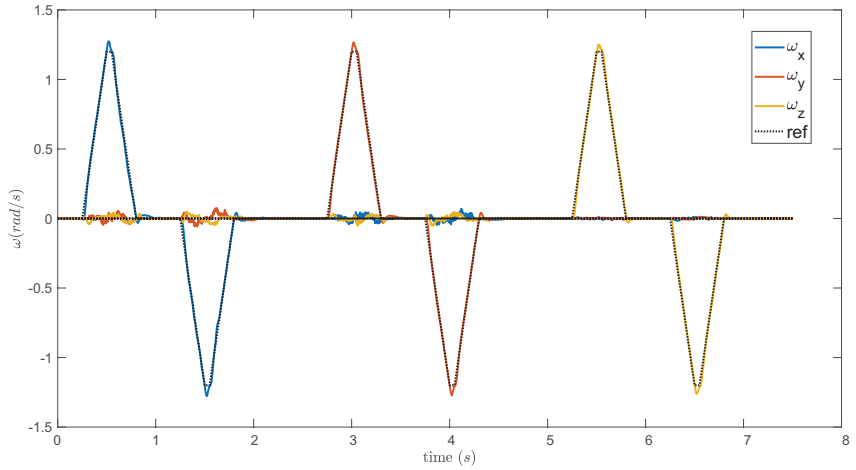

Figure 6: The angular velocities during the identification movement.

followed by a motionless phase of about 1 s for the identification of mass and COM. Fig. 6 and Fig. 7 show the tracking of the task expressed in form of angular velocities and accelerations, respectively.

Moreover, Fig. 8 and Fig. 9 show the F/T sensor measurements compared to the forces and torques reconstructed offline using the estimated velocities and accelerations and the identified inertial parameters.

The identified parameters and the percentage errors are reported in Table 1. Note that the adopted reference values come from the direct measurement for m, while $\mathbf{c}$ and $\mathbf{I}$ are CAD-computed using the measured mass as input for computation.

Table 1: Identification results

	Steel block			Cylinder		
	reference	identified	error (%)	reference	identified	error (%)
m (kg)	2.7830	2.7806	0.08	1.0600	1.0598	0.01
$mc_x(\mathrm{kg}\cdot\mathrm{m})$	0	$< 10^{-7}$	0	0	$< 10^{-7}$	0
$mc_y(\mathrm{kg}\cdot\mathrm{m})$	$-4.682\cdot10^{-2}$	$-4.662\cdot10^{-2}$	0.42	0	$< 10^{-7}$	0
$mc_z(\mathrm{kg}\cdot\mathrm{m})$	$1.015\cdot10^{-1}$	$1.014\cdot10^{-1}$	0.05	$1.908\cdot10^{-2}$	$1.904\cdot10^{-2}$	0.16
$I_{xx}(\mathrm{kg}\cdot\mathrm{m}^2)$	$6.928\cdot10^{-3}$	$7.033\cdot10^{-3}$	1.51	$1.259\cdot10^{-3}$	$1.278\cdot10^{-3}$	1.54
$I_{yy}(\mathrm{kg}\cdot\mathrm{m}^2)$	$6.634\cdot10^{-3}$	$7.407\cdot10^{-3}$	11.64	$1.259\cdot10^{-3}$	$1.417\cdot10^{-3}$	12.57
$I_{zz}(\mathrm{kg}\cdot\mathrm{m}^2)$	$3.326\cdot10^{-3}$	$3.081\cdot10^{-3}$	7.36	$1.603\cdot10^{-3}$	$1.755\cdot10^{-3}$	9.48

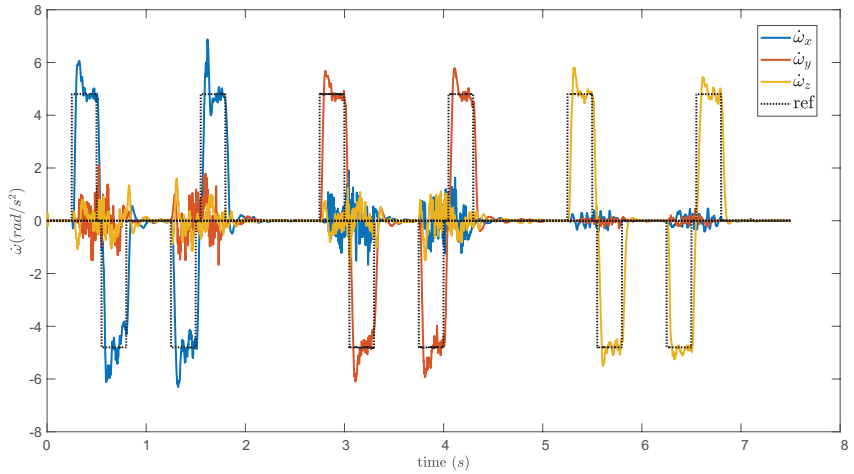

Figure 7: The angular acceleration during the identification movement.

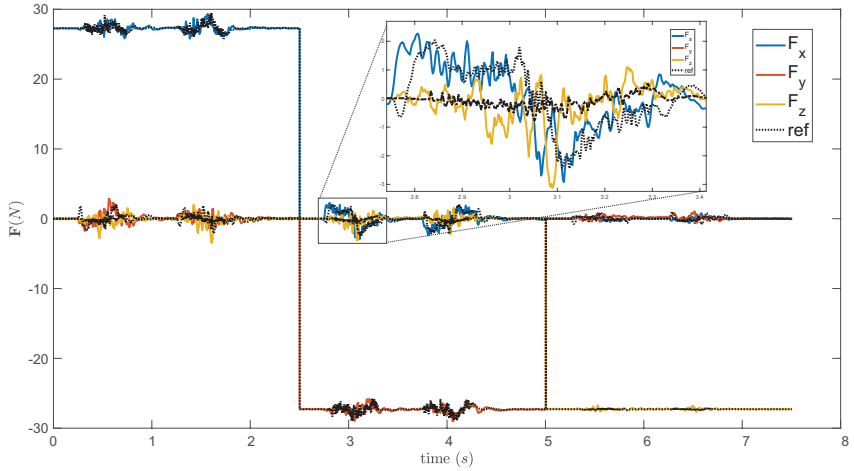

Figure 8: Forces measured by the F/T sensor during the identification movement.

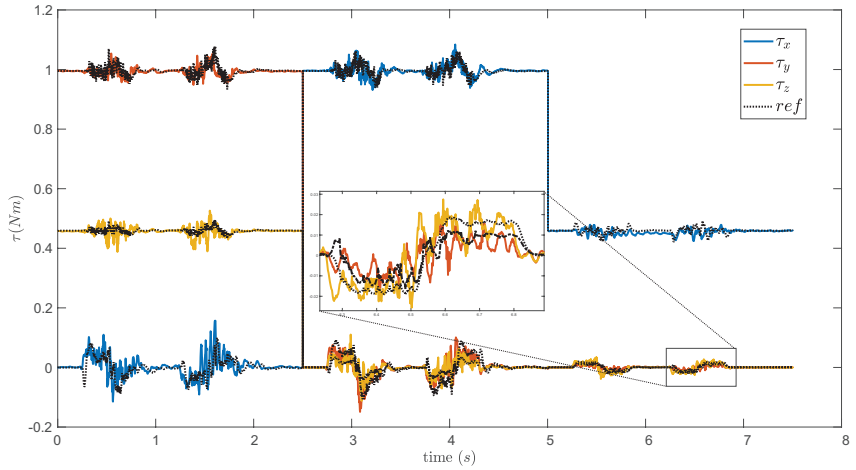

Figure 9: Forces measured by the F/T sensor during the identification movement.

Error analysis

The main sources of error in the identification of inertial parameters are the F/T sensor measurement noise and the uncertainties on the estimations of orientation and angular acceleration of the F/T sensor frame provided by the Kalman filter during the identification motions. Table 2 shows the variances of F/T sensor measurements, computed during the motionless phases and the variance of the Kalman filter estimations of orientation (represented by the quaternion $\mathbf{q}$) and angular acceleration $\dot{\boldsymbol{\omega}}$. In the following the experiments are referred to the steel-block payload.

Table 2: The main uncertainty sources

(a) F/T sensor noise variances

Parameter	Variance
$F_x(\mathrm{N})^2$	$3.0 \cdot 10^{-3}$
$F_y(\mathrm{N})^2$	$2.3 \cdot 10^{-3}$
$F_z(\mathrm{N})^2$	$5.3 \cdot 10^{-3}$
$\tau_x(\mathrm{Nm})^2$	$7.9 \cdot 10^{-6}$
$\tau_y(\mathrm{Nm})^2$	$9.3 \cdot 10^{-6}$
$\tau_z(\mathrm{Nm})^2$	$9.2 \cdot 10^{-6}$

(b) Kalman filter estimation variances

Parameter	Variance
$q_{0..3}$	$\approx 10^{-14}$
$\dot{\omega}_x(\mathrm{rad/s^2})^2$	$1.1 \cdot 10^{-3}$
$\dot{\omega}_y(\mathrm{rad/s^2})^2$	$1.1 \cdot 10^{-3}$
$\dot{\omega}_z(\mathrm{rad/s^2})^2$	$1.1 \cdot 10^{-3}$

It is worth observing that the variances associated to orientation estimation are negligible if compared to those of F/T measurements, so that it can

be assumed that in Eq. (3) the gravity vector $\mathbf{g}$, computed on the basis of the estimated quaternion, is not affected by uncertainty propagation, while the only error source derives from F/T sensor measurements. Therefore, the covariance matrix $\mathbf{\Sigma}$ of the identified parameters m and $m\mathbf{c}$ can be recursively expressed as in [46]:

$$\mathbf{\Sigma}_k = [\mathbf{1} - \mathbf{\Sigma}_{k-1}\mathbf{regr}_k^T(\mathbf{regr}_k\mathbf{\Sigma}_{k-1}\mathbf{regr}_k^T + \mathbf{\Lambda})] \tag{8}$$

where $\mathbf{\Lambda}$ contains the variances of the F/T sensor measurements and $\mathbf{1}$ is the identity matrix. Note that in case of mass identification $\mathbf{\Sigma} = \Sigma_m$ reduces to one-element matrix (Alg. 1), while $\mathbf{\Sigma} = \mathbf{\Sigma_c}$ is a 3x3 matrix in case of $m\mathbf{c}$ identification (Alg. 2).

Uncertainties can be finally defined by means of the 1-sigma rule as the square root of the diagonal terms (neglecting correlations) of the error covariance matrix $\mathbf{\Sigma}$, to obtain a 68% confidence level:

$$\begin{aligned} \delta m &= \sqrt{\Sigma_m} \\ \delta(mc_x) &= \sqrt{\mathbf{\Sigma_c}_{1,1}} \\ \delta(mc_y) &= \sqrt{\mathbf{\Sigma_c}_{2,2}} \\ \delta(mc_z) &= \sqrt{\mathbf{\Sigma_c}_{3,3}} \end{aligned} \tag{9}$$

where the symbol δ indicates the uncertainty associated to a variable.

On the other hand, the error model of Eq. (8) is not appropriate for calculating the uncertainties associated to the identified values of I_{xx},I_{yy},I_{zz}. In Eq. (4) both uncertainties on torque measurements and angular acceleration estimations have to be taken into account as they have a similar order of magnitude. In that case, uncertainties can be propagated by means of the following calculations [60]:

$$\begin{aligned} \frac{\delta I_{xx}}{I_{xx,\text{ref}}} &= \sqrt{(\frac{\delta\tau_x}{\tau_{x,\text{ref}}})^2 + (\frac{\delta\dot{\omega}_x}{\dot{\omega}_{x,\text{ref}}})^2} \\ \frac{\delta I_{yy}}{I_{yy,\text{ref}}} &= \sqrt{(\frac{\delta\tau_y}{\tau_{y,\text{ref}}})^2 + (\frac{\delta\dot{\omega}_y}{\dot{\omega}_{y,\text{ref}}})^2} \\ \frac{\delta I_{zz}}{I_{zz,\text{ref}}} &= \sqrt{(\frac{\delta\tau_z}{\tau_{z,\text{ref}}})^2 + (\frac{\delta\dot{\omega}_z}{\dot{\omega}_{z,\text{ref}}})^2} \end{aligned} \tag{10}$$

in which $I_{xx,\text{ref}}$ is the CAD-computed target value of I_{xx}, $\delta\tau_x$ and $\delta\dot{\omega}_x$ are calculated as the square root of the variance reported in Table 2 (i.e. 1-sigma uncertainty), $\dot{\omega}_{x,\text{ref}} = 4.8$ rad/s is the target acceleration and $\tau_{x,\text{ref}} = I_{xx,\text{ref}} \cdot \dot{\omega}_{x,\text{ref}}$ is the expected torque during the identification motion. The components of y and z axis are analogously defined.

The uncertainties on the inertial parameters are reported in Table 3, computed as the final values from of recursive procedure (8) for m and $m\mathbf{c}$ and from the uncertainty propagation formula (10) for I_{xx},I_{yy},I_{zz}. Such theorical uncertainties are a bit smaller than the experimental identification errors. However, it is worth to note that the previously described propagation analysis does not take into account systematic errors such as the misalignment of the robot basis with respect to the horizontal plane.

Table 3: Uncertainties on inertial parameters

Parameter	Reference value and uncertainty
m (kg)	$2.7830 \pm 7.9 \cdot 10^{-4}$ (0.03 %)
mc_x (kg·m)	$0 \pm 1.8 \cdot 10^{-6}$
mc_y (kg·m)	$-4.682 \cdot 10^{-2} \pm 1.4 \cdot 10^{-6}$ (0.003 %)
mc_z (kg·m)	$1.015 \cdot 10^{-1} \pm 2.0 \cdot 10^{-6}$ (0.002 %)
I_{xx} (kg·m^2)	$6.928 \cdot 10^{-3} \pm 2.6 \cdot 10^{-4}$ (3.7 %)
I_{yy} (kg·m^2)	$6.634 \cdot 10^{-3} \pm 1.6 \cdot 10^{-4}$ (2.4 %)
I_{zz} (kg·m^2)	$3.326 \cdot 10^{-3} \pm 1.2 \cdot 10^{-4}$ (3.6 %)

6. Conclusions

The paper has described the logical sequence applied to design a methodology for the identification of the robot payload, supporting human-robot collaborative applications. In particular, the design of the identification procedure started from considering that the effect on a wrist-mounted F/T sensor (or estimator) of a relevant inertial parameter (i.e. the mass, the center of mass or the diagonal terms of the inertia tensor) can be decoupled from the effects of other parameters by executing specific test motions. Such test motions can then be safely executed even in presence of human operators, provided that the robot configuration at the beginning of motion is properly selected (i.e. optimizing the distance from safe constrained zones or other obstacles and the manipulability of the robot in that configuration) and that both cartesian and joint velocities/accelerations are kept within appropriate bounds.

An aspect that is particularly relevant for future extensions of the proposed method, which has not been considered so far, is the adaptation of the

planned paths and trajectories to the real situation of the collaborative workcell. Indeed, in a really collaborative scenario humans should not be forced to operate within a limited part of the workspace. Dynamically changing safety zones would require online re-scheduling or even re-planning of the payload identification motion sequence. The design of an efficient and practically feasible solution for this issue is the main objective of future works. A further development could involve a quantitative analysis of the requirements in terms of payload identification accuracy and its effects on the detection of external forces and collision, aimed at the compliance with the international standard specifications on collaborative applications.

References

[1] Andrea Cherubini, Robin Passama, André Crosnier, Antoine Lasnier, and Philippe Fraisse. Collaborative manufacturing with physical human–robot interaction. *Robotics and Computer-Integrated Manufacturing*, 40:1–13, 2016.

[2] David A Duque, Flavio A Prieto, and Jose G Hoyos. Trajectory generation for robotic assembly operations using learning by demonstration. *Robotics and Computer-Integrated Manufacturing*, 57:292–302, 2019.

[3] Matteo Ragaglia, Andrea Maria Zanchettin, Luca Bascetta, and Paolo Rocco. Accurate sensorless lead-through programming for lightweight robots in structured environments. *Robotics and Computer-Integrated Manufacturing*, 39:9–21, 2016.

[4] ISO 10218-1:2011. Robots and robotic devices – Safety requirements for industrial robots. Part 1: Robots, 2011.

[5] ISO 10218-2:2011. Robots and robotic devices – Safety requirements for industrial robots. Part 2: Robot systems and integration, 2011.

[6] ISO/TS 15066:2016. Robots and robotic devices – Collaborative robots, 2016.

[7] F. Platbrood and O. Gornemann. *Safe robotics - Safety in collaborative robot systems*. SICK AG, Waldkirch, Germany, 2017.

[8] Jeremy A Marvel and Rick Norcross. Implementing speed and separation monitoring in collaborative robot workcells. *Robotics and computer-integrated manufacturing*, 44:144–155, 2017.

[9] F. D'Ippolito, F. Alonge, and E. Cucco. Contact estimation in robot interaction. *International Journal of Advanced Robotic Systems*, 11(96), 2014.

[10] C.-N. Cho and J.-B. Song. Collision detection algorithm robust to model uncertainty. *International Journal of Control, Automation and Systems*, 11(4):776–781, August 2013.

[11] M. Geravand, F. Flacco, and A. De Luca. Human-robot physical interaction and collaboration using an industrial robot with a closed control architecture. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4000–4007, Karlsruhe, Germany, May 2013.

[12] M. Makarov, A. Caldas, M. Grossard, P. Rodriguez-Ayerbe, and D. Dumur. Adaptive filtering for robust proprioceptive robot impact detection under model uncertainties. *IEEE/ASME Transactions on Mechatronics*, 19(6), December 2014.

[13] M.S. Erden and T. Tomiyama. Human-intent detection and physically interactive control of a robot without force sensors. *IEEE Transactions on Robotics*, 26(2), April 2010.

[14] Federica Ferraguti, Chiara Talignani Landi, Cristian Secchi, Cesare Fantuzzi, Marco Nolli, and Manuel Pesamosca. Walk-through programming for industrial applications. *Procedia Manufacturing*, 11:31–38, 2017.

[15] C. Talignani Landi, F. Ferraguti, L. Sabattini, C. Secchi, and C. Fantuzzi. Admittance control parameter adaptation for physical human-robot interaction. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2911–2916, Singapore, 2017.

[16] C. Talignani Landi, F. Ferraguti, L. Sabattini, C. Secchi, M. Bonfè, and C. Fantuzzi. Variable admittance control preventing undesired oscillating behaviors in physical human-robot interaction. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 3611–3616, Vancouver, Canada, 2017.