A model-based residual approach for human-robot collaboration during manual polishing operations☆

Claudio Gaz, Emanuele Magrini, Alessandro De Luca*

Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, Via Ariosto 25, Roma 00185, Italy

ABSTRACT

A fully robotized polishing of metallic surfaces may be insufficient in case of parts with complex geometric shapes, where a manual intervention is still preferable. Within the EU SYMPLEXITY project, we are considering tasks where manual polishing operations are performed in strict physical Human-Robot Collaboration (HRC) between a robot holding the part and a human operator equipped with an abrasive tool. During the polishing task, the robot should firmly keep the workpiece in a prescribed sequence of poses, by monitoring and resisting to the external forces applied by the operator. However, the user may also wish to change the orientation of the part mounted on the robot, simply by pushing or pulling the robot body and changing thus its configuration. We propose a control algorithm that is able to distinguish the external torques acting at the robot joints in two components, one due to the polishing forces being applied at the end-effector level, the other due to the intentional physical interaction engaged by the human. The latter component is used to reconfigure the manipulator arm and, accordingly, its end-effector orientation. The workpiece position is kept instead fixed, by exploiting the intrinsic redundancy of this subtask. The controller uses a F/T sensor mounted at the robot wrist, together with our recently developed model-based technique (the residual method) that is able to estimate online the joint torques due to contact forces/torques applied at any place along the robot structure. In order to obtain a reliable residual, which is necessary to implement the control algorithm, an accurate robot dynamic model (including also friction effects at the joints and drive gains) needs to be identified first. The complete dynamic identification and the proposed control method for the human-robot collaborative polishing task are illustrated on a 6R UR10 lightweight manipulator mounting an ATI 6D sensor.

1. Introduction

One challenging objective in the next generation of smart factory floors is to bring humans and robots close together, working efficiently and collaborating safely in the same shared manufacturing environment. This is being pursued in several national and international industry-oriented programs under different names, such as Industry 4.0, Cyber-Physical Systems, or Internet of Things [1].

In this context, recent research progresses dealing with physical Human-Robot Interaction (pHRI) have covered mechanical, actuation, sensing, planning, and control issues in an integrated way, with the goal of increasing safety and dependability of robotic systems [2]. In [3], we have originally proposed a control architecture devoted to pHRI, which is organized in three nested functional layers addressing, respectively, human-robot safety, coexistence, and collaboration.

Safety is the inherent and most important feature of a robot that has to work close to or collaborate with human beings. Due to the limits of

sensors and robot motion capabilities, e.g., when the human moves faster than the robot can sense or counteract, undesired contacts or collisions with humans may occur. This is handled in the lowest (direct) layer of the control architecture, which implements sensorless collision detection, isolation, and reflex reaction based on our residual method [4–6], a model-based scheme that monitors the generalized momentum of the robot.

By coexistence, we mean the robot capability of sharing a dynamic environment with humans, without requiring mutual contact or co-ordination of actions. The intermediate layer in our architecture realizes coexistence by featuring online collision avoidance capabilities, based on workspace monitoring by external sensors such as cameras or RGB-Depth devices (see, e.g., [7,8]), while human safety requirements are still being imposed.

Finally, the top control layer addresses collaboration, in which the robot performs a complex task with direct human interaction and co-ordination. We refer to safe collaboration when this activity does not

rule out a safe robot behavior, namely when both safety and coexistence features are guaranteed during the collaboration phase.

As a result, upper layers in this architecture will prescribe only robot reactive behaviors that are consistent with the objectives and constraints of lower layers. Interestingly enough, the three layers of the proposed control architecture can be easily mapped to the most recent requirements of safety standards and recommendations for collaborative robots, such as the ISO technical specification TS-15066 [9,10].

The top control layer handles both contact-less human-robot coordination and, most importantly here, physical collaboration tasks, in which a continuous and intentional contact takes place with a controlled exchange of forces/torques, as activated by multimodal communication such as voice, gestures, or touch [11,12].

In common industrial applications, the robot interacts with the environment mainly with a tool mounted on the end-effector. A force/torque (F/T) sensor placed on the final robot flange provides then the exchanged forces/torques. When the contact point is known in advance, the problem of sensor-less contact force estimation without the use of a F/T sensor has been addressed in [13], based on motor torque/current measurements. A dithering feedforward torque is used to decrease uncertainties due to friction, so as to improve force estimation when the robot is not in motion. Under similar operative conditions, learning-based approaches have also been applied for estimating exchanged external forces [14,15]. The estimated external force and the saturation of motor control torques can be used to keep the actual exchanged forces under a safety threshold [16].

In order to execute a wider range of activities involving Human-Robot Collaboration (HRC), physical contacts should not be limited in advance to a designated tool at the end-effector level, but rather whole-arm manipulation conditions should be considered. This raises the additional issue of reconstructing the exchanged forces at generic contact points along the robot structure, either by measuring them (e.g., using tactile sensitive skin in different locations [17]) or by estimating them in an indirect way, possibly combining model-based methods with other less invasive external sensors.

In [18], a first example of a method that estimates contact forces occurring at generic points (a priori unknown) on the robot arm was given. This was obtained by a so-called virtual force sensing approach, combining proprioceptive information of the residual method with localization of the contact location provided by a Kinect camera. The framework has been enriched in successive works [19,20], where other generalized control laws have been developed thanks to the improved knowledge of the dynamic model obtained on the KUKA LWR 4 + robot [21,22], which is exploited for the computation of the residual.

The H2020 European research project SYMPLEXITY [23] aims at developing solutions for complex surface finishing operations to be accomplished by collaborative work of humans and robots. Within this project, we are currently transferring our know-how on physical human-robot interaction to industrial set-ups proposed by end users.

In this paper, we present the core control algorithm that allows a human operator to kinesthetically reorient a workpiece held by the robot. In this context, the user should be able to change the orientation of the part while performing a surface polishing operation on it. It is desirable to achieve this in a natural way, by manually pushing or pulling the robot structure, rather than by commanding the robot via a separated interface (such as a teach-pendant, a pedal, or a keyboard). To this purpose, we need to decouple the effects of the two types of contact forces exerted by the operator against the robot: one due to the execution of the manual polishing task on the workpiece mounted on the robot end effector, the other due to the intentional reconfiguration force applied to the robot body. While the polishing force can be directly measured thanks to a F/T sensor mounted on the end-effector, the residual signals can estimate the torques at robot joint level resulting from both types of contact forces. By a suitable but simple elaboration, the intentional component can be isolated and will be used by the controller to command the robot motion. In particular, the

control law exploits the redundancy of the robot by specifying joint velocities in the null-space of an appropriate kinematic task, keeping the position of the end-effector fixed, while allowing relaxation of its orientation.

The paper is organized as follows. Section 2 briefly introduces the SYMPLEXITY laser polishing cell in which a HRC task has to be accomplished in a manual station. Section 3 presents the proposed control algorithm to achieve the HRC task. For the robot used in the experiments, a Universal Robots UR10, Section 4 presents the complete identification of the dynamic model, which is needed for the computation of residuals —the core signals in our model-based solution approach. This dynamic model is improved in accuracy with respect to the one supplied by the manufacturer, and includes also the novel estimation of friction terms and current-to-torque motor gains. Section 5 reports on the obtained experimental results, illustrated also in a video accompanying the paper. Finally, conclusions and on-going work are summarized in Section 6.

2. Human-Robot collaboration in a manual polishing station

Three main technologies are used for the robotized smoothing of metallic surfaces down to micrometer levels of roughness: Abrasive Finishing (AF), Laser Polishing (LP), and Fluid-jet Polishing (FP). All three technologies have been considered in industrial case studies within the SYMPLEXITY project, especially from the point of view of robot collaboration with a human operator both in execution and learning. We focus here on the LP cell. The safety requirements in this case do not allow a human to interfere at any time with the laser source, which is used with a robot moving inside a closed machine. However, this fast and precise technology for surface finishing can be used only if the roughness state of the workpiece at the start is already acceptable. In order to check this, metrology should be used to assess the initial quality of the part and, in case, manual abrasive polishing should be performed on selected parts. In this situation, HRC represents a handy solution.

A virtual rendering of the various components of the LP cell is shown in Fig. 1. Access to the LP machine occurs through an automatic airlock. A robot is used to move around the workpiece, from the operator desk to the ultrasonic cleaning bath, or from the measurement tool to the airlock, and so on. When the part is in need of a manual polishing before being ready to be processed by the LP machine, it will be carried by the robot in front of the human operator. HRC will occur then, with the human using a tool to polish selected surfaces of the part (those indicated graphically on a Human-Machine Interface) and the robot holding the workpiece in a specified sequence of poses. The correct and safe operation of the manual polishing station will be monitored by two RGB-D (Kinect) sensors, measuring in real time minimum distances between selected control points on the robot and the whole environment (including the dynamically moving human operator) [8]. This will prevent accidental collisions and avoid unintended contacts when the robot is moving around (safe coexistence). On the other hand, an obvious exchange of forces should occur at the end-effector level during execution of the manual polishing, with forces in the order of 20 N and peaks up to 35 N. In addition, extra intentional forces will be applied by the human operator to the robot body to reorient the end-effector and complete the task in a ergonomic and natural way. Operation and control of these physical HRC events is presented in the next section.

The cell will be used for polishing small bio-compatible metallic parts of complex shape for physiological uses in medicine. The weight of the parts is in the order of 10 grams —negligible with respect to the part holder and gripper mounted on the robot end-effector.

3. Decoupling the external forces when a robot holds a workpiece

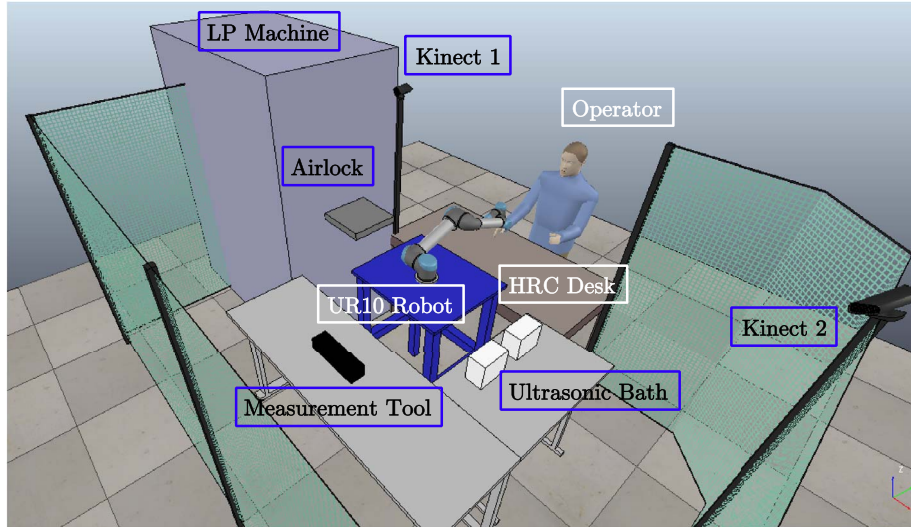Consider a rigid robot manipulator with n joints and generalized

Fig. 1. Layout of the manual polishing station in the SYMPLEXITY LP cell (in V-REP [24]).

coordinates $q \in \mathbb{R}^n$, modeled by Euler–Lagrange equations in the usual form

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + \tau_f(\dot{q}) = \tau + \tau_{ext}, \qquad (1)$$

where $M(q)$ is the positive definite, symmetric robot inertia matrix, Coriolis and centrifugal forces (quadratic in $\dot{q}$) are factorized using a square matrix $C(q, \dot{q})$ such that $\dot{M} - 2C$ is skew-symmetric, $g(q)$ are the gravity terms, $\tau_f(\dot{q})$ contains frictional forces, τ are the motor torques at the joints, and $\tau_{ext} \in \mathbb{R}^n$ is the vector of joint torques resulting from possible external Cartesian forces.

When the robot is holding a workpiece to be manually polished, the forces and moments that the human operator exerts on the surface of the part are reflected along the robot structure from the end-effector to the robot joints. In some cases, these generalized forces could even move the joints, if brakes were not activated or high-gain positional feedback was not used on the motors[1] This situation has to be avoided in order not to compromise the quality of the operation carried out on the part. Nevertheless, it may be desirable, or even necessary, during such an operation to reorient the workpiece in order to achieve better results. When the robot is firmly holding an object, though, exerting a force on the workpiece would produce no motion. Conversely, intentional forces exerted by the operator on the manipulator structure could usefully affect the reorientation of the part. Thus, such a contact situation should be detected, the original control task should be relaxed accordingly, and a new suitable command generated, by exploiting the degrees of kinematic redundancy that the robot gains from the relaxation of the original task. On the other hand, contacts occurring occasionally on the manipulator body when the robot is in motion should always be considered as accidental collisions and handled differently for safety (see, e.g., [6]).

The forces exerted on the workpiece at the end-effector and on the manipulator structure are both reflected at the joint level as τ_{ext} in Eq. (1). In view of the above discussion, it is however necessary to separate these two contributions, in such a way that only the intentional force acting on the robot links is fed into the robot motion control law. Let

$$\tau_{ext} = \tau_e + \tau_c = J_e^T(q)F_e + J_c^T(q)F_c, \qquad (2)$$

with vectors $\tau_e \in \mathbb{R}^n$ and $\tau_c \in \mathbb{R}^n$ being, respectively, the joint torque due to the generalized force $F_e = (f_e^T \ m_e^T)^T \in \mathbb{R}^6$ exerted on the end-

effector, and the joint torque due to the generalized force at the contact $F_c = (f_c^T \ m_c^T)^T \in \mathbb{R}^6$. Moreover, $J_e \in \mathbb{R}^{6 \times n}$ and $J_c \in \mathbb{R}^{6 \times n}$ are, respectively, the (geometric) Jacobian associated to the robot end-effector and the contact Jacobian associated to the contact area along the structure. Indeed, for specific types of interaction some components of the generalized forces may vanish (e.g., for a point-wise contact, at most only the three linear components f_c of F_c will be different from zero).

With reference to Fig. 2, in order to evaluate τ_{ext}, and then to separate it into the two quantities τ_e and τ_c as in (2), the values of F_e and F_c have to be known, as well as the link where the contact occurs, which is necessary to compute the Jacobian matrix J_c. The generalized force F_e can be measured thanks to a force/torque sensor mounted at the robot wrist/end-effector, like in our experimental set-up, and the related Jacobian J_e will be the one associated with the reference frame attached to the calibrated sensor. On the other hand, it is quite difficult to reconstruct the generalized force F_c, especially if we don't make any further assumption on the type of contact and on the link at which this type of intentional interaction occurs (which may be anywhere along the manipulator structure). An estimate of the linear components f_c of F_c for point-wise contacts could be obtained as in [19], but it would require the identification of the actual contact position on the link (and thus of J_c), by means of an external sensor, like a depth camera.

However, the solution is relatively easy when a F/T wrist sensor is available. In fact, for our purposes we only need to obtain an estimate of the τ_c component, and not of its original source F_c and location. Since τ_e can be obtained from the F/T sensor measurement, it is sufficient to resort to a slight modification of the model-based approach originally proposed in [5] for sensorless collision detection, which generates a residual vector $r \in \mathbb{R}^n$ by monitoring the generalized momentum $p = M(q)\dot{q}$ of the robot. Thus, we can define a modified (and computable) residual as

$$r(t) = G\left(M(q)\dot{q} - \int_0^t (C^T(q, \dot{q})\dot{q} - g(q) - \tau_f(\dot{q}) + \tau + \tau_e + r)ds\right), \qquad (3)$$

where $G \in \mathbb{R}^{n \times n}$ is a positive, diagonal gain matrix, and we set $r(0) = 0$ when the robot initially at rest. All terms in (3) can be evaluated efficiently, once a reliable robot dynamic model (1) is available. Note in particular that:

- No joint acceleration measurements nor inversion of the inertia matrix are needed in the residual formula;
- The joint torque contribution τ_e of the generalized force acting on the robot end-effector (due to the manual polishing) is subtracted

---

[1] High-gain positional feedback is preferred over braking, in order not to interrupt the operation flow for too long, as well as to transit between successive phases in a smoother way (i.e., by control).

Fig. 2. Human-Robot collaboration in the presence of a F/T sensor mounted at the robot wrist. A stiff positional behavior can be achieved during manual polishing in reaction to a generalized force $\boldsymbol{F}_e$ (left), while a generalized contact force $\boldsymbol{F}_c$ can be estimated separately and in parallel, and used to reorient the robot (right).

from the residual —a component which was absent in [5];

- The applied command torques $\boldsymbol{\tau}$, which are not restricted to any specific control law, are obtained from the measured motor currents $\boldsymbol{i}$, as $\boldsymbol{\tau} = \boldsymbol{Ki}$, with $\boldsymbol{K} = \text{diag}\{K_1, \cdots, K_n\}$ and where $K_j > 0$ is the current-to-torque gain of the jth motor;
- The friction term $\boldsymbol{\tau}_f(\dot{\boldsymbol{q}})$ is indeed critical, being usually the most difficult to be modeled and correctly identified;
- At very low robot speed (and at rest), the residual can be simplified by eliminating all terms that vanish together with the joint velocity $\dot{\boldsymbol{q}}$, leaving essentially the gravity torque $\boldsymbol{g}(\boldsymbol{q})$, the joint torques $\boldsymbol{\tau}$ and $\boldsymbol{\tau}_e$, and the static friction part of the torque $\boldsymbol{\tau}_f$, if any;
- Residuals have the same dimensional units [Nm] of motor torques at the joints; they can also be evaluated at the level of motor currents (in [A] units), scaling each term by the relative gain K_j.

Using the properties of the Christoffel symbols [25] for the conservative velocity terms in the dynamic model (1), it is easy to check that the residual vector $\boldsymbol{r}$ will automatically satisfy the following relation, which is useful for analysis:

$$\dot{\boldsymbol{r}} = \boldsymbol{G}(\boldsymbol{\tau}_c - \boldsymbol{r}), \text{ or, equivalently, } \dot{r}_i = G_i(\tau_{c,i} - r_i), \ i = 1, \cdots, n. \quad (4)$$

Therefore, the residual $\boldsymbol{r}$ will stay at zero (up to noise and model uncertainties) as long as there is no external joint torque applied along the manipulator structure. Once $\boldsymbol{\tau}_c$ is present, the residual will react to it as a first-order stable linear filter. As a result, $\boldsymbol{r}$ will converge exponentially to the true value of $\boldsymbol{\tau}_c$, if this is constant at least for a short period of time. The residuals will also return exponentially to zero when the contact is removed —an interesting feature for our application. In general, we will use sufficiently large gains G_i's and, based on (4), consider in our application the residual $\boldsymbol{r}$ and the joint torque $\boldsymbol{\tau}_c$ as superposed, i.e., $\boldsymbol{r} \simeq \boldsymbol{\tau}_c$. An example of behavior of residuals in practice is reported in Section 4.5, for a UR10 robot having $n = 6$ (see Fig. 10).

The control concept at the basis of this work is to produce self-motions of the robot in response to the human operator pushing by hand the manipulator structure, generating thus a $\boldsymbol{\tau}_c$ that is estimated by $\boldsymbol{r}$. The target is to keep the position of the end-effector fixed while relaxing its orientation. This will allow the operator to easily reorient the part to be manually polished, without pushing a button, touching a screen, or using other separate HMI devices, and eventually not even needing to suspend the manual polishing task as we shall see. To this purpose, we rewrite the Jacobian $\boldsymbol{J}_e(\boldsymbol{q})$ into its linear and angular components as

$$\boldsymbol{J}_e(\boldsymbol{q}) = \begin{pmatrix} \boldsymbol{J}_p(\boldsymbol{q}) \\ \boldsymbol{J}_o(\boldsymbol{q}) \end{pmatrix}, \qquad \boldsymbol{J}_p \in \mathbb{R}^{3 \times n}, \quad \boldsymbol{J}_o \in \mathbb{R}^{3 \times n}.$$

Assuming that the robot joints are controlled by velocity references (kinematic control), in order to provide the proper self-motion, the following control law is adopted

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_p^{\#}(\boldsymbol{q})\boldsymbol{K}_p(\boldsymbol{p}_d - \boldsymbol{p}(\boldsymbol{q})) + (\boldsymbol{I} - \boldsymbol{J}_p^{\#}(\boldsymbol{q})\boldsymbol{J}_p(\boldsymbol{q}))\boldsymbol{K}_r\boldsymbol{r}, \quad (5)$$

where the $n \times 3$ matrix $\boldsymbol{J}_p^{\#}$ is the unique pseudoinverse of the linear Jacobian $\boldsymbol{J}_p$, $\boldsymbol{p}_d \in \mathbb{R}^3$ is the desired (constant) position of the end-effector, $\boldsymbol{p} \in \mathbb{R}^3$ is its actual position, as computed through the robot direct kinematics, and $\boldsymbol{K}_p$ and $\boldsymbol{K}_r$ are (diagonal) gain matrices of dimension 3×3 and $n \times n$, respectively. In (5), the residual $\boldsymbol{r}$ will drive the null-space joint velocities through the $n \times n$ projection matrix $\boldsymbol{I} - \boldsymbol{J}_p^{\#}\boldsymbol{J}_p$ of the positional task. Since an equivalent joint torque generates a joint velocity command, the controller (5) belongs to the class of *admittance* laws at the joint level. The Cartesian gains $\boldsymbol{K}_p$ will be chosen as large as possible in order to realize a very stiff position control. Note finally that the choice of a kinematic control law, as opposed to a torque control law for $\boldsymbol{\tau}$, is mandatory when the robot has a closed control architecture and no direct access to motor currents/torques is allowed (as in the case of the UR10 manipulator).

4. Dynamic modeling and identification of the UR10 manipulator
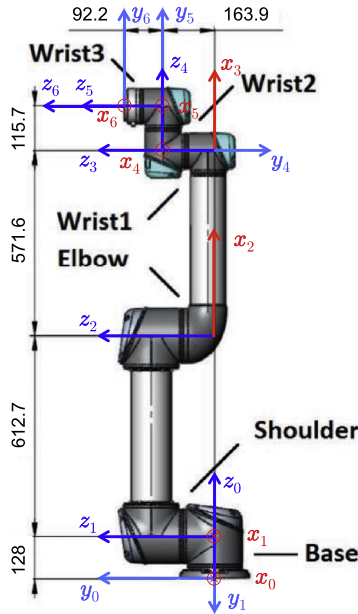
4.1. Motivation

As target robot for our HRC control task, we have considered the 6-dof lightweight manipulator UR10 by Universal Robots. Fig. 3 shows the kinematic frames and the associated table of Denavit-Hartenberg (DH) parameters.

Summarizing the result of Section 3, a good knowledge of the dynamic model of the robot is mandatory in order to have a reliable residual signal (3), necessary to implement the control action (5). The manufacturer distributes a software simulator on its website [26], which includes data files with the numerical values of all dynamic parameters of the UR10 manipulator. We exploited those data in order to reconstruct all dynamic terms in the model (1) of this robot.

Unfortunately, we found that the dynamic model reconstructed in this way is not sufficiently reliable, both in static and dynamic conditions. Extensive comparisons between measured motor currents[2] and currents computed via the nominal data provided by the manufacturer on a (relatively slow) sinusoidal motion trajectory showed quite different behaviors on all joints, with deviations as large as 9 [A] for the second joint. These differences are due in part also to the friction term

---

[2] In all experiments, measurements of joint positions $\boldsymbol{q}$ and motor currents $\boldsymbol{i}$ were collected using the URScript Programming Language [27].

Fig. 3. The UR10 manipulator with the chosen (classical) DH frames and the associated table of kinematic parameters. Lengths are expressed in [mm].

i	a_i	α_i	d_i	θ_i
1	0	$-\pi/2$	$d_1 = 128$	q_1
2	$a_2 = 612.7$	0	0	q_2
3	$a_3 = 571.6$	0	0	q_3
4	0	$-\pi/2$	$d_4 = 163.9$	q_4
5	0	$\pi/2$	$d_5 = 115.7$	q_5
6	0	0	$d_6 = 92.2$	q_6

Table 1

Extended dynamic coefficients of the gravity term. Comparison between the coefficients computed using the parameters supplied by the manufacturer [26] and the coefficients estimated with our method. Units are [A].

Dynamic coefficients	Using UR10 nominal parameters	Our estimation
ξ_1/K_2	0	6.73×10^{-4}
ξ_2/K_2	0	-2.88×10^{-3}
ξ_3/K_2	0.0094	0.0038
ξ_4/K_2	0	0.0068
ξ_5/K_2	0.0395	0.0201
ξ_6/K_2	0	-0.002
ξ_7/K_2	0	0.0036
ξ_8/K_2	0.611	0.2661
ξ_9/K_2	0	7.52×10^{-4}
ξ_{10}/K_2	1.284	0.5557
ξ_1/K_3	0	0.0014
ξ_2/K_3	0	-8.64×10^{-4}
ξ_3/K_3	0.0075	0.0022
ξ_4/K_3	0	-7.4×10^{-4}
ξ_5/K_3	0.0317	0.0238
ξ_6/K_3	0	0.0011
ξ_7/K_3	0	2.61×10^{-4}
ξ_8/K_3	0.4896	0.3264
ξ_1/K_4	0	2.7×10^{-5}
ξ_2/K_4	0	-3.9×10^{-4}
ξ_3/K_4	0.0078	0.005
ξ_4/K_4	0	0.0012
ξ_5/K_4	0.0329	0.0275
ξ_6/K_4	0	-0.0011
ξ_1/K_5	0	8.06×10^{-4}
ξ_2/K_5	0	0.0031
ξ_3/K_5	0.0078	0.0032
ξ_4/K_5	0	-0.0029
ξ_1/K_6	0	3.58×10^{-5}
ξ_2/K_6	0	-4.07×10^{-4}

τ_f, whose parameters are in fact not provided by the manufacturer. Even in static conditions, we found differences in the evaluation of the currents associated with the gravity term $g(q)$.

As a result, when used within the residual (3), the dynamic data of the robot manufacturer would lead to a poor discrimination of contacts, e.g., with multiple false positives or false negatives (depending on the

thresholds of detection). This motivated us to perform a new identification procedure in order to improve the accuracy of the terms in the dynamic model.

4.2. Identification in the presence of unknown motor gains

We present here the basic identification procedure that is needed for the UR10 manipulator. The specific features of this robot are the unknown current-to-torque motor gains (*aka*, drive gains), the presence of a relatively large friction component, and the availability of joint position and motor current measurements only (no joint torque sensor). This combination calls for some adaptation of otherwise standard robot dynamic identification methods (see, e.g., [28]). From now on, we set specifically $n = 6$ in (1).

Given the linear dependence of the dynamic model on the dynamic parameters of the robot links [25], it is possible to rearrange the left-hand side of (1) in terms of a full rank regressor matrix $Y \in \mathbb{R}^{6 \times p}$ that multiplies a vector $\pi \in \mathbb{R}^p$ of dynamic coefficients, i.e., a number p of linear combinations of the link dynamic parameters [29], as

$$Y(q, \dot{q}, \ddot{q})\pi = \tau = Ki. \tag{6}$$

The vector $\tau \in \mathbb{R}^6$ of motor torques is expressed as the product of the drive gains matrix $K = \text{diag}\{K_1, \cdots, K_6\}$ times the vector of motor currents $i \in \mathbb{R}^6$. Since the drive gains are also unknown, we can formally divide each of the six equations in (6) by the corresponding K_j, thus obtaining the new linear equations

$$Y_j(q, \dot{q}, \ddot{q})\,\pi^{(j)} = i_j, \ j = 1, \cdots, 6, \tag{7}$$

where $Y_j \in \mathbb{R}^p$ is the jth *row* of the regressor Y, i_j is the current of motor j, and we have extended the vector π of dynamic coefficients, by replicating it in a scaled form relative to each joint as

$$\pi^{(j)} = \left(\frac{\pi_1}{K_j} \ \cdots \ \frac{\pi_p}{K_j} \right)^T \in \mathbb{R}^p, j = 1, \cdots, 6. \tag{8}$$

In matrix form, Eqs. (7) and (8) are rewritten as

$$Y^{[e]}(q, \dot{q}, \ddot{q})\pi^{[e]} = i, \tag{9}$$

where the extended vector of dynamic coefficients $\pi^{[e]} \in \mathbb{R}^{6p}$ is

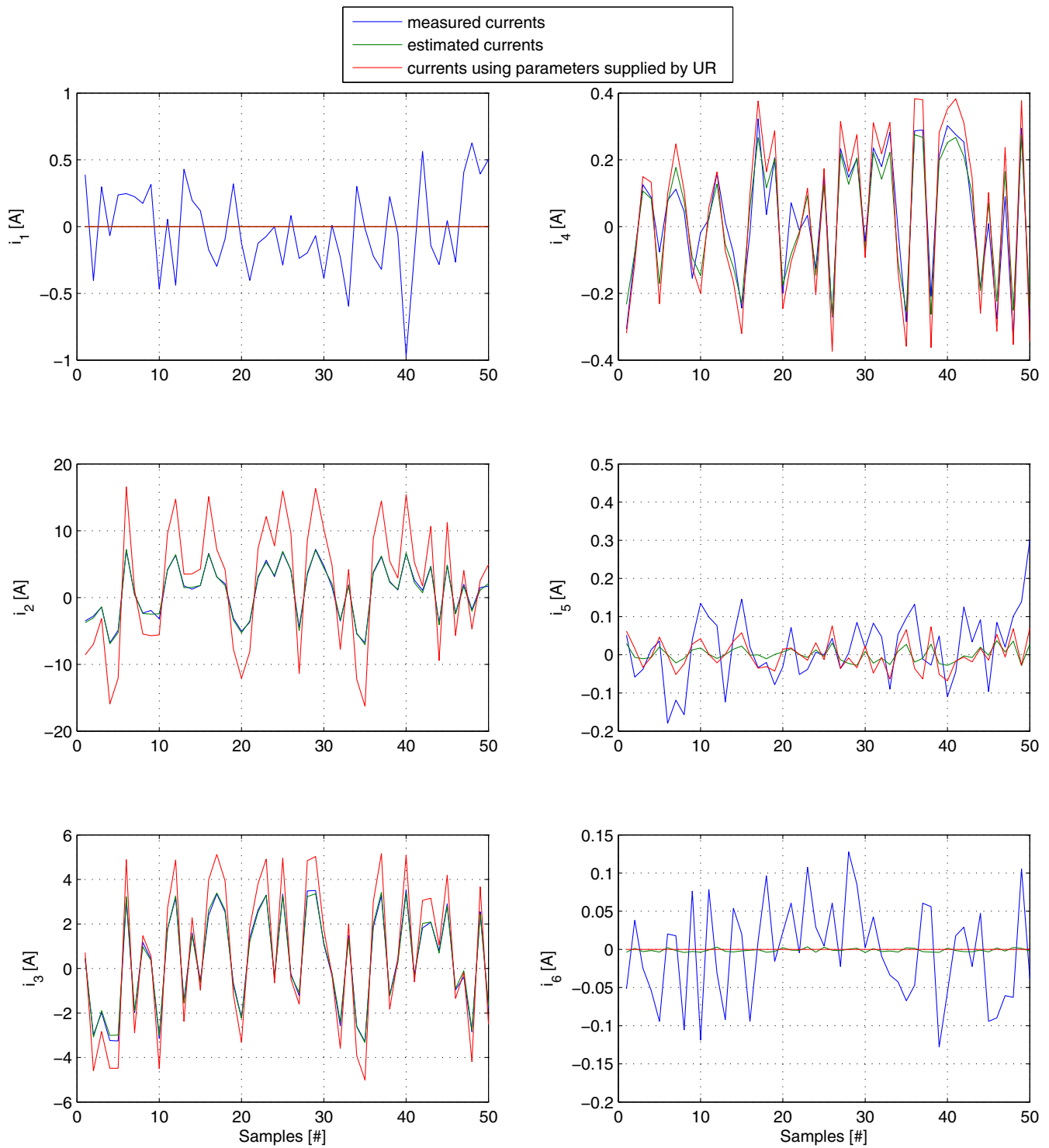$$\pi^{[e]} = \left(\pi^{(1)T} \ \pi^{(2)T} \ \cdots \ \pi^{(6)T} \right)^T$$

Fig. 4. Comparison between the sets of 50 measured and computed motor currents for the six joints of the UR10 robot in static conditions: measured currents (blue); currents computed using the dynamic parameters supplied by the manufacturer (red); currents estimated with our identification process (green). For the main joints 2 and 3, the green curves are practically superposed to the blue ones. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

and the extended regressor matrix $Y^{[e]} \in \mathbb{R}^{6 \times 6p}$ has a block diagonal structure

$$Y^{[e]} = \text{blockdiag}\{\; Y_1(q, \dot{q}, \ddot{q}), \;\; Y_2(q, \dot{q}, \ddot{q}), \;\; \cdots, Y_6(q, \dot{q}, \ddot{q})\}.$$

Thanks to this structure of $Y^{[e]}$, which is in fact a *stretching* of the rows of the original matrix Y, the problem is decomposed into six sub-problems that can be solved in parallel.

Collecting the actual positions $q(t)$ and the associated motor currents $i(t)$ along a sufficiently exciting nominal trajectory, one can proceed as usual with an off-line numerical differentiation and obtain accurate approximations for $\dot{q}(t)$ and $\ddot{q}(t)$ on the recorded motion. Extracting from these quantities a sufficiently large number M of samples $\{q_k, \dot{q}_k, \ddot{q}_k, i_k\}$, $k = 1, \cdots, M$, we can build for each joint j the following stacks of M rows:
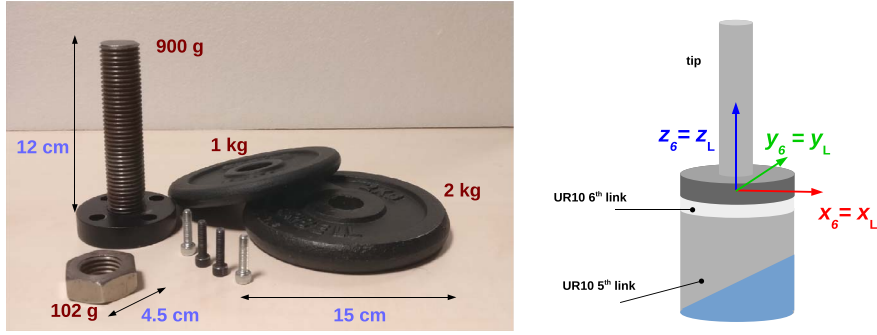
Fig. 5. The payload parts used for identifying the motor gains (left). A scheme with the payload reference frame, which is coincident with the frame of the sixth link (right).
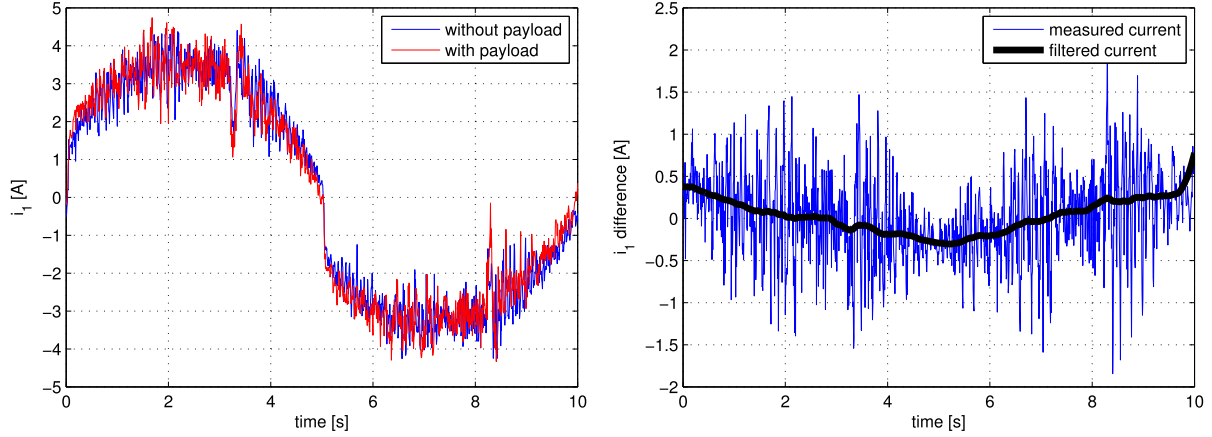

Fig. 6. Motor currents at joint 1 during an exciting trajectory, without (blue) and with (red) a payload [left]. Difference of the measured currents and its filtered version [right]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$\overline{Y}_j = \begin{pmatrix} Y_j(\boldsymbol{q}(t_1), \dot{\boldsymbol{q}}(t_1), \ddot{\boldsymbol{q}}(t_1)) \\ \vdots \\ Y_j(\boldsymbol{q}(t_M), \dot{\boldsymbol{q}}(t_M), \ddot{\boldsymbol{q}}(t_M)) \end{pmatrix}, \quad \bar{\boldsymbol{i}}_j = \begin{pmatrix} i_j(t_1) \\ \vdots \\ i_j(t_M) \end{pmatrix}, j = 1, \cdots, 6. \tag{10}$$

Then, for each separate linear problems $\overline{Y}_j \pi^{(j)} = \bar{\boldsymbol{i}}_j$, $j = 1, \cdots, 6$, defined via (10), we can solve for the extended dynamic coefficients using standard pseudoinversion:

$$\hat{\pi}^{(j)} = \overline{Y}_j^\# \bar{\boldsymbol{i}}_j, j = 1, \cdots, 6. \tag{11}$$

The values $\hat{\pi}^{[e]}$ retrieved with this modified identification procedure would be sufficient for the design of model-based trajectory control laws (e.g., feedback linearization), if we assume that motor currents can be directly imposed to the robot. However, the original dynamic coefficients π would be estimated only up to the current-torque gains. Moreover, since frictional phenomena are difficult to model in general and are quite relevant for the considered manipulator, an integrated identification procedure would not lead to robust results. On the other hand, we found that the inertial parameters provided by the manufacturer [26], as opposed to the gravity parameters, were already sufficiently good for our purposes, not deserving a new separate identification.

Therefore, we will pursue here a different, more structured two-step identification approach. First, we estimate in static conditions both the dynamic coefficients in the gravity vector $\boldsymbol{g}(\boldsymbol{q})$ and the drive gains $\boldsymbol{K}$, by using two sets of experiments with and without a known payload [22,30]. Next, we use this information to estimate the friction term $\tau_f(\dot{\boldsymbol{q}})$ in dynamic conditions.

4.3. Estimating gravity coefficients and motor gains

With reference to Fig. 3, we consider the UR10 manipulator mounted on a horizontal table. The gravity acceleration is then

$\boldsymbol{\gamma} = (0 \quad 0 \quad -g_0)^T$, with $g_0 = 9.81$ m/s². In static conditions, $\dot{\boldsymbol{q}} = \ddot{\boldsymbol{q}} = \boldsymbol{0}$, Eqs. (6)–(9) specialize to

$$\boldsymbol{g}(\boldsymbol{q}) = \text{col}\{g_j(\boldsymbol{q})\} = Y_g(\boldsymbol{q})\pi_g = K\boldsymbol{i}$$

$$\Rightarrow \boldsymbol{g}^{[e]}(\boldsymbol{q}) = \text{col}\left\{\frac{g_j(\boldsymbol{q})}{K_j}\right\} = Y_g^{[e]}(\boldsymbol{q})\pi_g^{[e]} = \boldsymbol{i}. \tag{12}$$

The gravity vector $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^6$ (in [Nm] units) is computed symbolically using a Lagrangian approach, then linearly parametrized in terms of the dynamic coefficients π_g, and finally scaled in the form $\boldsymbol{g}^{[e]}(\boldsymbol{q}) \in \mathbb{R}^6$ (in [A] units) of Eq. (12). In this way, we found $p_g = \dim \pi_g = 10$ independent coefficients. However, because of the many structural zeros present in the different rows $Y_{g,j}$ of Y_g, the expanded vector $\pi_g^{[e]}$ has only 30 non-vanishing components (rather than $6 \cdot p_g = 60$). Moreover, the drive gain K_1 is not appearing in any of these expressions, because the first component of the gravity vector $\boldsymbol{g}(\boldsymbol{q})$ is $g_1 = 0$ (the first joint axis is vertical). As a result, we have:

$$\pi_g^{[e]} = \begin{pmatrix} \xi_1/K_2 \\ \vdots \\ \xi_{10}/K_2 \\ \xi_1/K_3 \\ \vdots \\ \xi_8/K_3 \\ \xi_1/K_4 \\ \vdots \\ \xi_6/K_4 \\ \xi_1/K_5 \\ \vdots \\ \xi_4/K_5 \\ \xi_1/K_6 \\ \xi_2/K_6 \end{pmatrix} \in \mathbb{R}^{30}, \quad \text{with} \quad \begin{aligned} \xi_1 &= c_{6y}m_6 \\ \xi_2 &= c_{6x}m_6 \\ \xi_3 &= c_{5z}m_5 + c_{6z}m_6 + d_6m_6 \\ \xi_4 &= c_{5x}m_5 \\ \xi_5 &= c_{5y}m_5 + c_{4z}m_4 + d_5m_5 \\ \xi_6 &= c_{4x}m_4 \\ \xi_7 &= c_{3y}m_3 \\ \xi_8 &= a_3(m_3 + m_4 + m_5 + m_6) + c_{3x}m_3 \\ \xi_9 &= c_{2y}m_2 \\ \xi_{10} &= a_2(m_2 + m_3 + m_4 + m_5 + m_6) + c_{2x}m_2. \end{aligned}$$
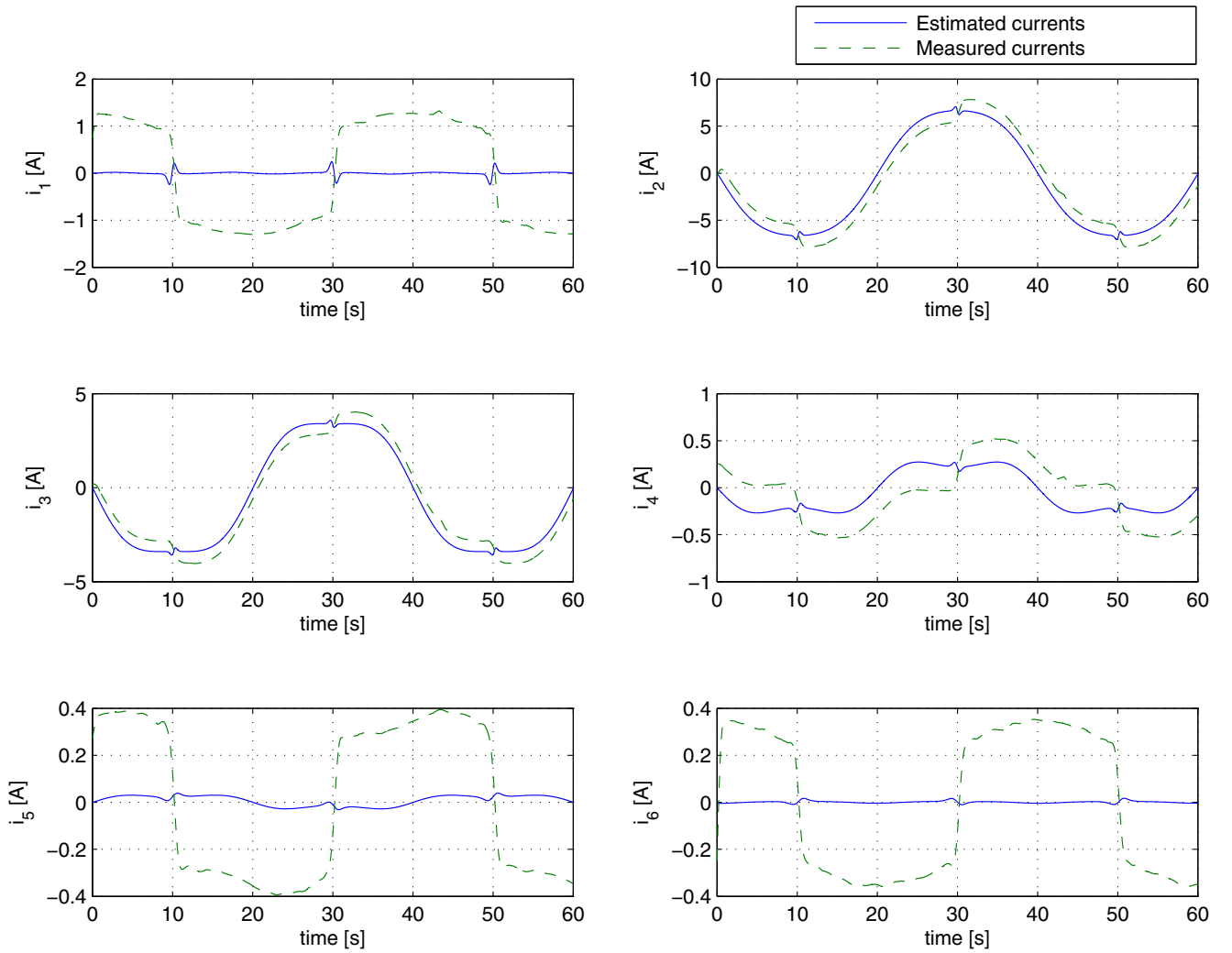
$$\tag{13}$$

Fig. 7. Comparison between measured (green, dashed) and estimated (blue, continuous) motor currents, neglecting friction. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The robot was moved to $M = 500$ different configurations spanning the entire workspace, and motor currents were retrieved statically when each desired position in the list was reached. Stacking these data, the extended gravity coefficients in (13) were estimated, and the associated motor currents computed as

$$\widehat{\pi}_g^{[e]} = \overline{Y}_g^{[e]\#} \, \overline{\boldsymbol{\imath}} \qquad \Rightarrow \qquad \widehat{\boldsymbol{g}}^{[e]}(\boldsymbol{q}) = Y_g^{[e]}(\boldsymbol{q}) \, \widehat{\pi}^{[e]}. \tag{14}$$

In (14), $\overline{\boldsymbol{\imath}} \in \mathbb{R}^{6 \cdot 500}$ is the stacked vector of measured motor currents, while the numerical regressor $\overline{Y}_g^{[e]} \in \mathbb{R}^{6 \cdot 500 \times 30}$ is again block diagonal and has full column rank. The obtained results are listed in Table 1, together with the values computed using the manufacturer's data for a comparison.

The estimation has been validated by placing the robot in 50 new joint configurations. The results are shown in Fig. 4. In all these static conditions, the measured motor currents match now quite well with the estimated currents $\widehat{\boldsymbol{g}}^{[e]}(\boldsymbol{q})$, whereas they are quite different from the currents computed from the manufacturer data (the peaks of these differences are in the order of 5–6 [A] on the main joints). Note also that for joint 1 we always get a zero estimate for the gravity term, while the current measures are slightly different from zero. This may be caused by an unmodeled static friction effect at $\dot{\boldsymbol{q}} = \boldsymbol{0}$. A similar mismatch is present for joint 6 as well.

The same identification procedure has been performed in order to estimate the drive gains K_j's, mounting on the end-effector the complete payload shown in Fig. 5 (for a total of 4.002 [kg]), and repeating the

same 500 positioning tasks. In the presence of a payload, the extended vector $\pi_g^{[e]}$ is modified to the loaded one $\pi_{g,L}^{[e]}$, using the following substitutions in the dynamic coefficients

$$c_{6v} \, m_6 \; \rightarrow \; c_{6v} \, m_6 + c_{Lv} \, m_L, \qquad v = x, y, z, \tag{15}$$

where the mass and the position of the center of mass of the sixth (last) link of the UR10 and of the payload are denoted as m_k and $\boldsymbol{c}_k = (c_{kx} \quad c_{ky} \quad c_{kz})^T$, respectively for $k = 6$ and $k = L$. At the end of the procedure, a new estimation $\widehat{\pi}_{g,L}^{[e]}$ is obtained, and the numerical difference $\widehat{\boldsymbol{\varepsilon}}_{g,L}^{[e]} = \widehat{\pi}_{g,L}^{[e]} - \widehat{\pi}_g^{[e]}$ is computed.

Since the symbolic difference vector $\boldsymbol{\varepsilon}_{g,L}^{[e]} = \pi_{g,L}^{[e]} - \pi_g^{[e]}$ contains the five scalars K_j, $j = 2, \cdots, 6$, as the only unknowns[3], their estimation is set up as follows. Define

$$\boldsymbol{K}^{\dagger} = \left(\frac{1}{K_2} \;\; \cdots \;\; \frac{1}{K_6} \right)^T. \tag{16}$$

By linearity, we can introduce a Jacobian matrix $\boldsymbol{J}_{\varepsilon_g}$, evaluated with the known payload data, and write

$$\boldsymbol{J}_{\varepsilon_g} \boldsymbol{K}^{\dagger} = \boldsymbol{\varepsilon}_{g,L}^{[e]}, \qquad \text{with} \quad \boldsymbol{J}_{\varepsilon_g} = \frac{\partial \boldsymbol{\varepsilon}_{g,L}^{[e]}}{\partial \boldsymbol{K}^{\dagger}}. \tag{17}$$

---

[3] Based on the structure of the coefficients ξ_j in (13) and on their modifications (15), the symbolic difference vector $\boldsymbol{\varepsilon}_{g,L}^{[e]}$ will have only 17 non-vanishing components. Numerical values for estimation are assigned only to these.
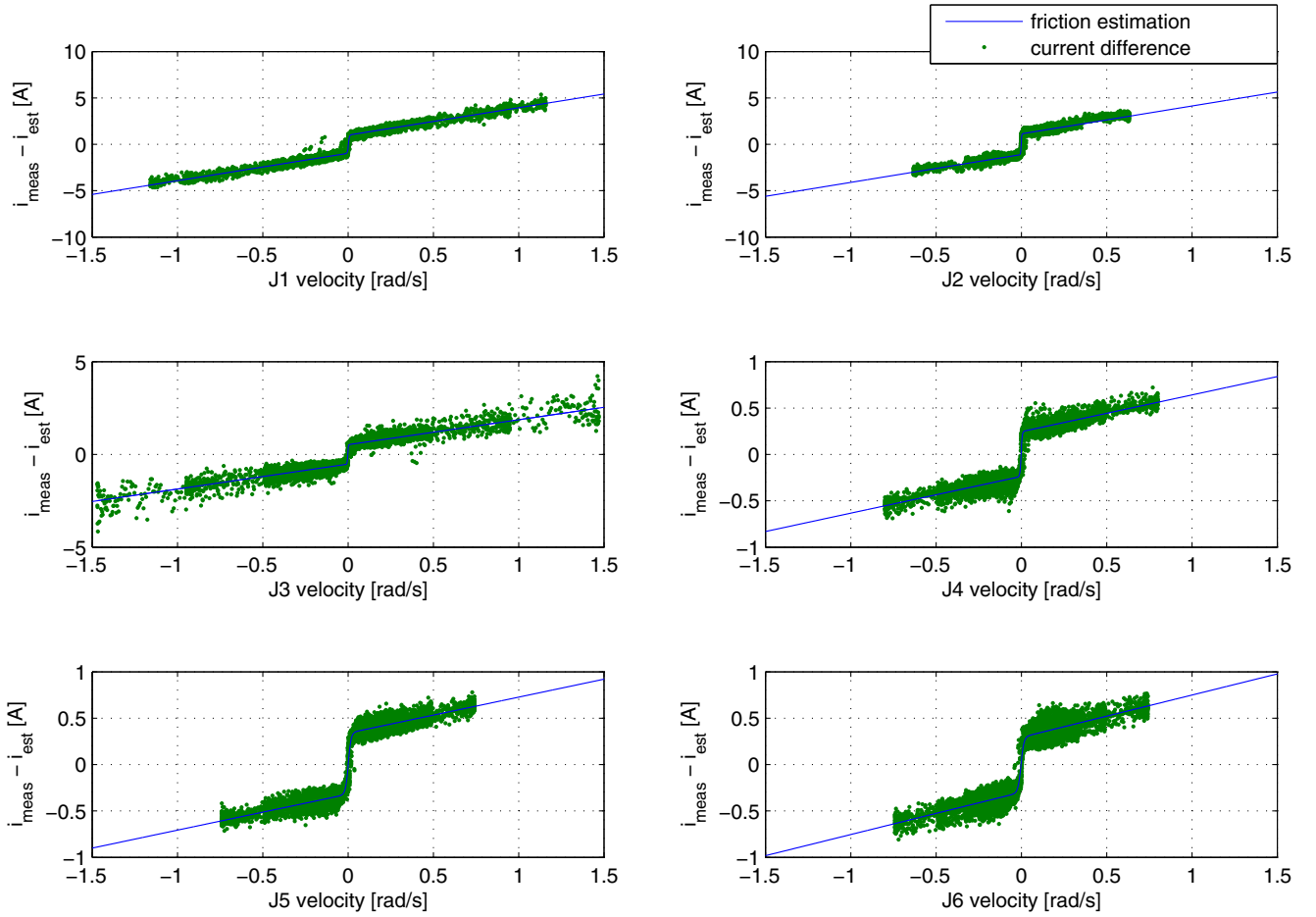
Fig. 8. The estimated friction functions (23) for the six UR10 robot joints. The data to fit (green dots) are the differences between measured and estimated currents. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The (inverse of the) drive gains are estimated by pseudoinversion as

$$\widehat{\boldsymbol{K}}^{\dagger} = \boldsymbol{J}_{\varepsilon}^{\#} \hat{\boldsymbol{\varepsilon}}_{g,L}^{[e]} \qquad \Rightarrow \qquad \widehat{\boldsymbol{K}} = \left(\frac{1}{\widehat{K}_{2}^{\dagger}} \quad \cdots \quad \frac{1}{\widehat{K}_{6}^{\dagger}} \right)^{T}. \tag{18}$$

The following estimates were obtained for the drive gains of the joints that are subject to gravity in the UR10 manipulator[4]:

$$\widehat{K}_2 = 13.26, \quad \widehat{K}_3 = 11.13, \quad \widehat{K}_4 = 10.62, \quad \widehat{K}_5 = 11.03, \quad \widehat{K}_6 = 11.47 \quad [\text{Nm/A}]. \tag{19}$$

In order to estimate the missing drive gain K_1 of joint 1, where no gravity is felt, a dynamic rather than static procedure was needed, including again pairs of experiments with and without payload. To maximize the inertial effects of the payload on joint 1, trajectories were designed that move only the first joint and keep all the others at rest, with the arm straight and almost parallel to the ground. Fig. 6 shows the measured motor currents during one of these trajectories, with and without the known payload, and their filtered difference. The motor gain of joint 1 was finally estimated as:

$$\widehat{K}_1 = 14.87 \quad [\text{Nm/A}]. \tag{20}$$

4.4. Estimating motor friction

After estimating the gravity vector and the diagonal matrix of drive gains, we have used the inertial parameters of the six links of the UR10 robot reported by the manufacturer in [26], suitably modified in order

to be expressed consistently with the DH frames of Fig. 3, to obtain an estimate $\widehat{\boldsymbol{M}}(\boldsymbol{q})$ of the inertia matrix. From this, by means of the Christoffel symbols [25], we derive also an estimate of the Coriolis and centrifugal matrix $\widehat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})$. Since we have an estimate of the drive gains, we can now determine the contribution of these dynamic terms to the motor currents associated to a desired trajectory. In fact, the jth equation in (7) can be rewritten more explicitly as

$$\frac{\boldsymbol{M}_j(\boldsymbol{q})}{K_j} \ddot{\boldsymbol{q}} + \frac{\boldsymbol{C}_j(\boldsymbol{q}, \dot{\boldsymbol{q}})}{K_j} \dot{\boldsymbol{q}} + \frac{g_j(\boldsymbol{q})}{K_j} = i_j, \tag{21}$$

where $\boldsymbol{M}_j$ and $\boldsymbol{C}_j$ are 6-dimensional row vectors, while g_j, K_j and i_j are scalar quantities. Combining the assumed inertial data with the previously estimated gravity vector $\hat{\boldsymbol{g}}^{[e]}(\boldsymbol{q})$ in (14) and estimated drive gains $\widehat{\boldsymbol{K}}$ in (19) and (20), we get the following estimation of the motor currents along a twice differentiable motion $\boldsymbol{q}(t)$:

$$\hat{i}_j = \frac{\widehat{\boldsymbol{M}}_j(\boldsymbol{q})}{\widehat{K}_j} \ddot{\boldsymbol{q}} + \frac{\widehat{\boldsymbol{C}}_j(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\widehat{K}_j} \dot{\boldsymbol{q}} + \frac{\hat{g}_j(\boldsymbol{q})}{\widehat{K}_j}, j = 1, \cdots, 6. \tag{22}$$

As a first validation experiment, we imposed the sinusoidal joint trajectories $q_j(t) = -(\pi/2) + (\pi/4)\sin((\pi/20)t)$, for $j = 2, 4$, and $q_j(t) = (\pi/4)\sin((\pi/20)t)$, for $j = 1, 3, 5, 6$, and performed a comparison between the motor currents estimated by (22) and the measured currents, filtered through a 4th-order zero-phase digital Butterworth filter with a cutoff frequency of 1 Hz. Fig. 7 shows that the remaining differences are still non-negligible, namely of the order of 0.4–1.4 [A] depending on the joint. We also noticed that the motor currents typically change sign together with the relative joint velocities, a fact that is especially evident for the first and last two joints, while they display in

---

[4] Note that an asymmetric payload has been used additionally for estimating the gain K_6 of the last motor, in order to produce a gravitational torque around the last joint axis.
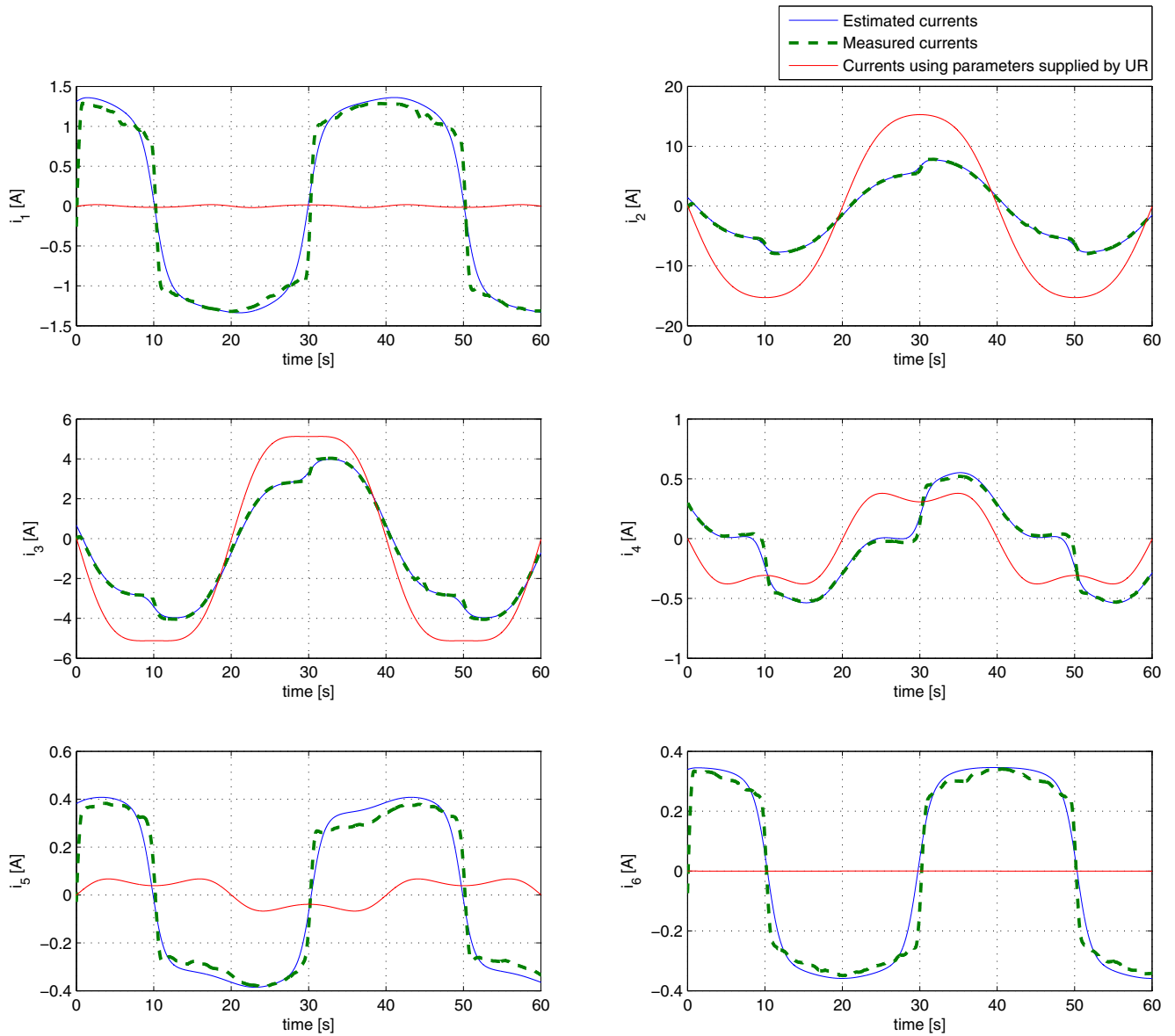
Fig. 9. Measured (green, dashed) and estimated (blue, continuous) motor currents, including friction. For comparison, we report also the motor currents (red, continuous) computed using the dynamic parameters supplied by the manufacturer [26]; these currents are clearly different from the measured and our estimated ones. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

general a small but sharp discontinuity close to the zero-velocity regime. These behaviors can be attributed to friction effects at the motors/transmissions that are velocity dependent.

Therefore, we proceeded with an estimation of the motor currents (in [A]) associated to the friction torque $\tau_f(\dot{q})$ (in [Nm]) in Eq. (1), assuming this last missing term as a function of the joint velocity. In order to derive a functional model for the motor friction, we executed several rest-to-rest cubic trajectories having different maximum speed, moving one joint at the time and keeping the others at rest. For every joint $j \in \{1, \cdots, 6\}$, we collected on the average 10k samples of the motor current. Eventually, we found that the best model that fits the data was given by a sigmoidal function added to an affine function as

$$\frac{\tau_{f,j}(\dot{q}_j)}{K_j} = (a_j \dot{q}_j + b_j) + \frac{S_j}{1 + e^{-\alpha_j\left(\dot{q}_j + \nu_j\right)}}.$$
(23)

This function is characterized by five parameters a_j, b_j, S_j, α_j, and ν_j. These were estimated solving a nonlinear least squares problem by means of a Nelder–Mead routine, using as fitting data the differences

between the measured motor currents and the estimated currents $\hat{i}_j$ given by (22). For the six joints of the UR10 robot, we have identified a total of 30 structural parameters for friction. The plots of the obtained friction functions are shown in Fig. 8.

Adding the estimated current components due to friction, we finally obtained the following estimation of the total motor current for each joint j:

$$\hat{i}_j = \frac{\widehat{M}_j(q)}{\widehat{K}_j}\ddot{q} + \frac{\widehat{C}_j(q, \dot{q})}{\widehat{K}_j}\dot{q} + \frac{\widehat{g}_j(q)}{\widehat{K}_j} + \frac{\widehat{\tau}_{f,j}(\dot{q})}{\widehat{K}_j}, j = 1, \cdots, 6.$$
(24)

As a final validation, we used (24) to estimate the motor currents along the same sinusoidal joint trajectories used to obtain Fig. 7. The plots in Fig. 9 show that the motor currents are now accurately predicted, meaning that an accurate dynamic model of the UR10 manipulator has been obtained.
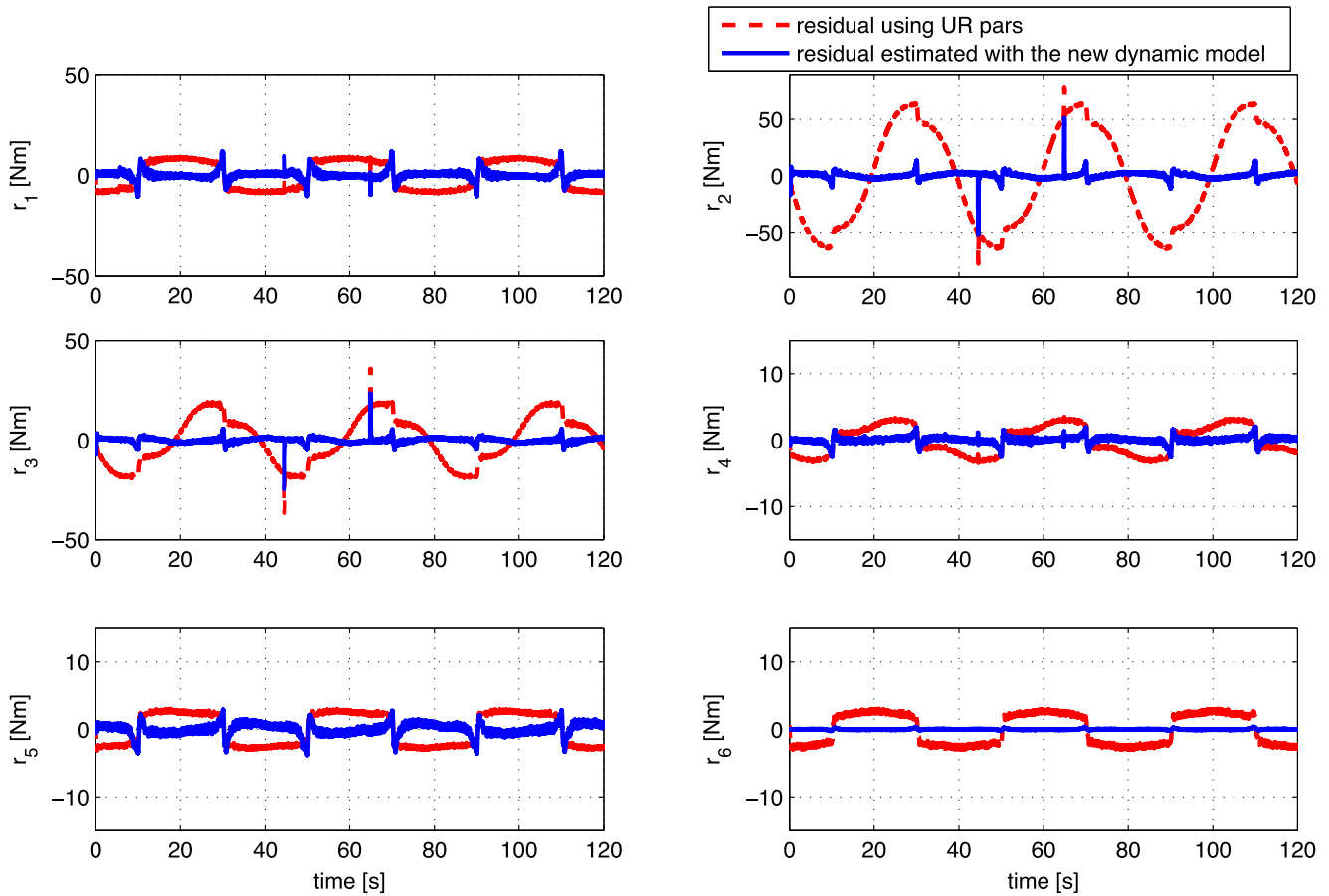
Fig. 10. Evolution of the estimated residuals (blue, continuous) during a nominal robot motion, while instantaneous but mild contact forces are exerted on link 3 at $t = 45$ s and $t = 65$ s. A collision can be correctly detected when at least one scalar component of the residual vector exceeds in absolute value a small threshold. This occurs for the residuals at joints 2 and 3. We report also the residuals (red, dashed) computed using the dynamic parameters supplied by the manufacturer. These signals are clearly inappropriate for collision detection, since they are dramatically non-zero even in the absence of a contact. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
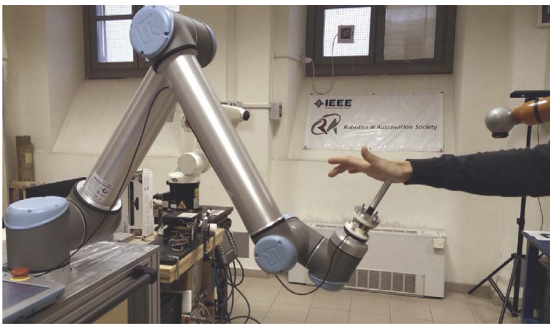

Fig. 11. End-effector position and orientation are both kept when the human is only pushing on the finger tool, mimicking an abrasive polishing task on a part mounted on the robot.

4.5. Collision detection test

The quality of the outcome of the identification process described in Section 4.3–4.4 has been further tested by verifying if and how one is able to detect possible collisions between a human operator and the UR10 manipulator while the robot is in motion, when using our estimated dynamic model.

Fig. 10 shows the evolution of the six components of the residual $\boldsymbol{r}$ in Eq. (3), when 2 almost instantaneous and mild contact forces are exerted along opposite Cartesian directions on link 3, respectively at the time instants $t = 45$ s and $t = 65$ s. The nominal sinusoidal motion of the joints was the same used to obtain the currents in Fig. 7. The two peaks

at joints 2 and 3 are clearly distinguishable from the residual noise, so that collisions can be safely detected by moderate thresholding. We remark that the residual method allows also isolation of the colliding link in the manipulator chain: its index j_c equals in fact the largest index among the components of the residual vector that exceed a detection threshold —in this case, it is $j_c = 3$.

For comparison, Fig. 10 shows also the residuals evaluated using the original dynamic parameters provided by the robot manufacturer [26]. In this case, there are permanently large non-zero values also in the absence of contacts, e.g., with oscillations going up to ± 60 [Nm] at joint 2, making the peaks in correspondence to the actual collisions hardly distinguishable from the background signal. It is quite evident that these untuned residuals are completely useless, since collision events are almost completely masked and intentional contact forces applied by the user would not be correctly recognized.

We note finally that the estimated residuals still exhibit small peaks whenever the joint velocity changes sign. This could be due to yet unmodeled dissipative effects, such as a larger static and Stribeck friction, with respect to those captured by the model equation (23).

5. HRC experimental results

We present here some experimental results of the proposed control law (5) applied to the Universal Robot UR10 manipulator. The hardware platform was a 64-bit Intel Core i7-6700K CPU, equipped with 16GB DDR4 RAM. The complete process of residual computation and motion control runs on the robot at 8 ms cycle time. In order to measure the external force $\boldsymbol{F}_e$ applied by the human, the robot is equipped with
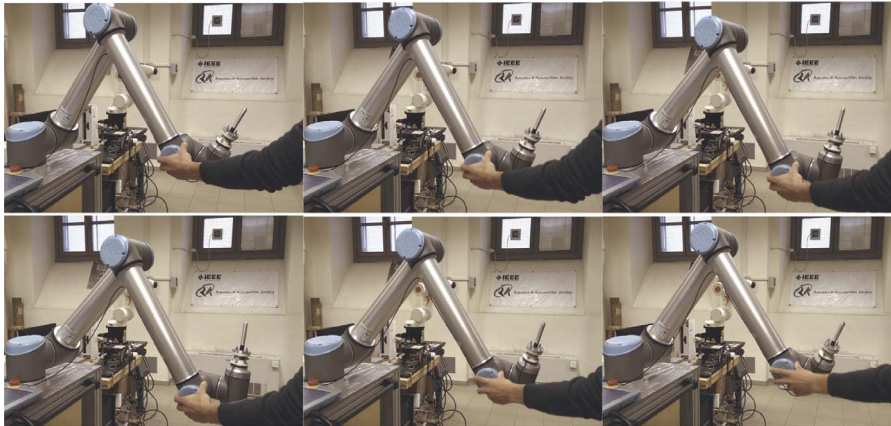
Fig. 12. Snapshots from the phase when the human only reorients the part by pushing on the robot structure.
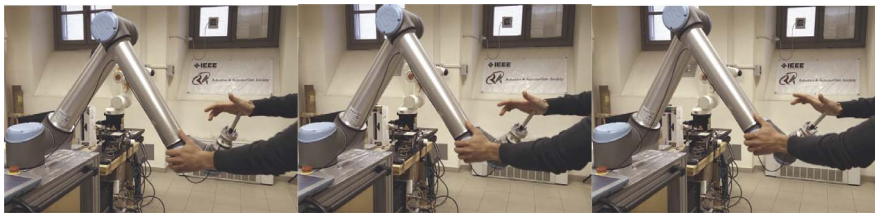

Fig. 13. Snapshots from the phase when the human pushes on the end-effector tool, while reorienting at the same time the part by pushing on the robot structure.

an ATI Mini45 6D Force/Torque sensor, mounted on its end-effector. With reference to the residual computation (3), the gain matrix G has been chosen as $G = 3.6 \cdot I_6$, where I_k denotes the $k \times k$ identity matrix. The other parameters in (5) were set to $K_p = 2.3 \cdot I_3$ and $K_r = 0.16 \cdot I_6$. Finally, the desired constant position p_d for the mimicked polishing task operation has been chosen as $p_d = (-0.07 \quad 0.89 \quad 0.31)^T$ [m] with respect to the base frame in Fig. 3.

During the proposed interaction experiment, the human pushes the robot at the end-effector level and/or along its structure to emulate a polishing process and the need to reorient the polished part. Figs. 11–13 show snapshots of the various phases of the experiment, while Fig. 14 reports the time behavior of different variables during the task: the robot end-effector position $p \in \mathbb{R}^3$, the joint position variables $q \in \mathbb{R}^6$, only the end-effector contact force $f_e \in \mathbb{R}^3$ measured by the F/T sensor, and the residual vector $r \in \mathbb{R}^6$ computed as (3). Due to uncertainties left in the dynamic model or unmodeled effects, and in order to keep limited the effects of noise in the motor current sensor, the residual vector has been considered to be zero if $\|r\| \leq r_{th}$, a threshold set to $r_{th} = 10$ [Nm]. The reported experiment can be seen in the accompanying video clip (also on YouTube at https://youtu.be/bjZbmlAclYk).

By comparing the plots of the variables of interest in Fig. 14 with the snapshots/video, one can distinguish different human-robot collaboration phases.

- In the interval $4\,s \leq t \leq 8\,s$, only the polishing force is exerted, which is completely represented by the force sensor measure, since no contact along the robot structure occurs (Fig. 11). As expected, position and orientation of the end-effector do not change and there is no joint motion. In fact, the control law counterbalances all manual polishing forces/torques applied by the operator. Note that, the residual vector remains equal to zero during this phase.
- When $10\,s \leq t \leq 15\,s$, the operator pushes only on the robot structure. The contact is detected by the residual and the robot joints move according to Eq. (5), relaxing the end-effector orientation. Thanks to the null-space projection, the end-effector position remains constant even if the joint configuration changes. In this way, the robot arm can be reconfigured by the operator so as to reorient the held workpiece in the most comfortable way. Fig. 12 shows this

situation.
- When $16\,s \leq t \leq 22\,s$, both a polishing force and an extra contact force/moment (on the robot structure) are exerted at the same time. Here, the joints move only due to the resulting extra joint torque detected by the residual. Still, the control law counterbalances all manual polishing forces/torques applied by the operator, as shown in Fig. 13.

Similar considerations can be done for the remaining parts of the experiment.

6. Conclusions

We have addressed a human-robot collaboration problem that arises in the robot-aided industrial task of abrasive surface polishing for metallic workpieces with complex geometry. We presented a control scheme that allows a human operator to physically move the robot arm that is holding the part to different configurations, reorienting in this way the workpiece on the end-effector while keeping instead fixed its position so as to better accomplish manual polishing. This robot behavior can be realized without extra interfaces (e.g., a touch screen or teach-pendant), by simply pushing the arm structure in a very natural way and implementing an admittance control law with a null-space algorithm to exploit kinematic redundancy of the task. The robot reconfiguration is obtained in the same way both if the operator has paused the activity or if is still exerting at the same time a polishing force on the workpiece. This decoupling is made possible by the simultaneous use of a standard force/torque sensor mounted on the robot wrist and of our model-based dynamic method of residuals, which estimates the joint torques associated to contact forces/moments applied at any location of the manipulator arm. We proved our algorithm with good results on a UR10 lightweight robot, mounting a small-size 6D F/T sensor and using otherwise the original low-level motion controller that accepts user-defined kinematic commands. A major subtask was to estimate an accurate dynamic model of the manipulator, including the significant friction effects and the motor current-to-torque gains, in order to compute a reliable residual signal.

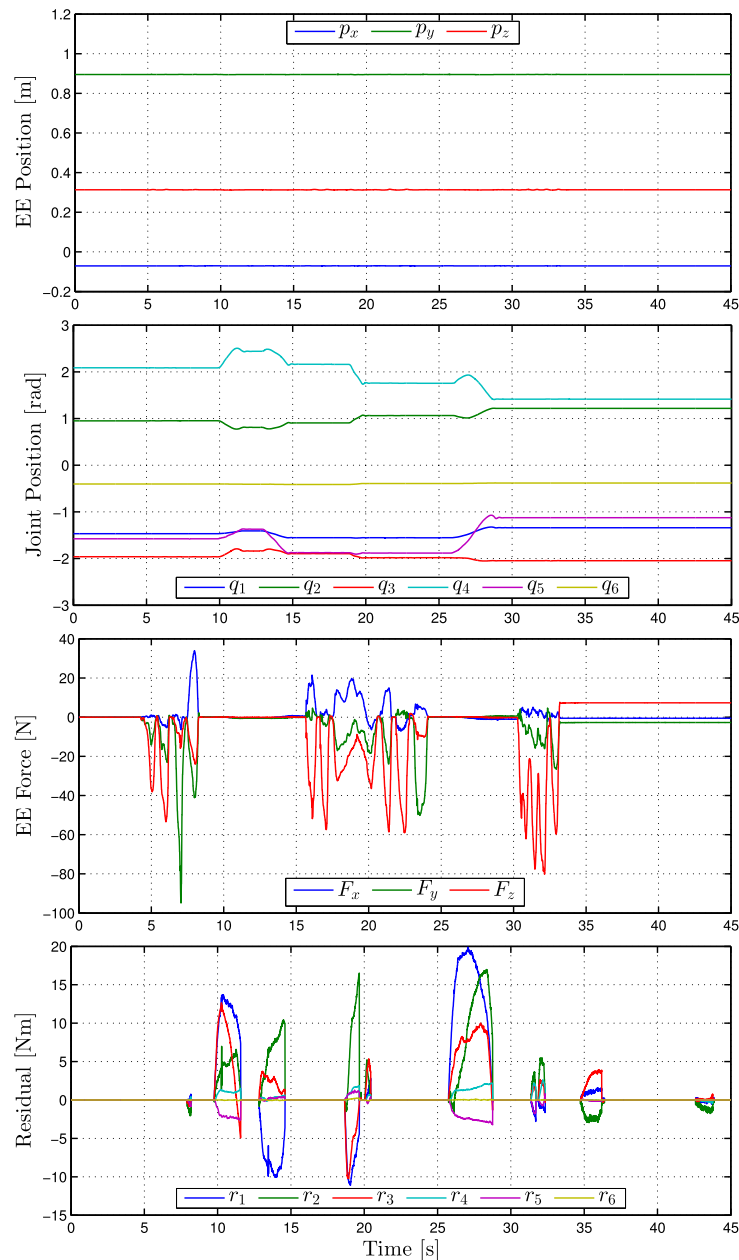Our current work within the SYMPLEXITY project is devoted to

Fig. 14. Human-robot collaborative experiment. From the top: end-effector Cartesian position, joint position variables, contact force at the end-effector measured by the F/T sensor, and components of the residual vector.

integration activities of this human-robot collaborative scheme into the actual manual abrasive polishing station of the industrial cell for robotized laser polishing, incorporating it in the overall process control flow and the available HMI and protocols, and using the two Kinects that monitors the safe human-robot coexistence also for improving the physical collaboration phase.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.mechatronics.2018.02.014.

References

[1] Lu Y. Industry 4.0: a survey on technologies, applications and open research issues. J Ind Inf Integr 2017;6:1–10.
[2] De Santis A, Siciliano B, De Luca A, Bicchi A. An atlas of physical human-robot interaction. Mech Mach Theory 2008;43(3):253–70.
[3] De Luca A, Flacco F. Integrated control for pHRI: collision avoidance, detection, reaction and collaboration. Proc. IEEE Int. Conf. on Biomedical Robotics and Biomechatronics. 2012. p. 288–95.
[4] De Luca A, Mattone R. Sensorless robot collision detection and hybrid force/motion control. Proc. IEEE Int. Conf. on Robotics and Automation. 2005. p. 1011–6.
[5] De Luca A, Albu-Schäffer A, Haddadin S, Hirzinger G. Collision detection and safe reaction with the DLR-III lightweight robot arm. Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems. 2006. p. 1623–30.
[6] Haddadin S, De Luca A, Albu-Schäffer A. Robot collisions: a survey on detection, isolation, and identification. IEEE Trans Rob 2017;33(6):1292–312.
[7] Flacco F, Kröger T, De Luca A, Khatib O. A depth space approach for evaluating distance to objects – with application to human-robot collision avoidance. J Intell Rob Syst 2015;80, Suppl. 1:7–22.
[8] Flacco F, De Luca A. Real-time computation of distance to dynamic obstacles with multiple depth sensors. IEEE Rob Autom Lett 2017;2(1):56–63.
[9] ISO 10218-1-2011. Robots and robotic devices – Safety requirements for industrial robots. Part 1: Robots; Part 2: Robot systems and integration. since July 1, 2011. URL http://www.iso.org.
[10] ISO TS 15066:2016. Robots and robotic devices – Collaborative robots. since February 15, 2016. URL http://www.iso.org.
[11] Lucignano L, Cutugno F, Rossi S, Finzi A. A dialogue system for multimodal human-

robot interaction. Proc. 15th ACM Int. Conf. on Multimodal Interaction. 2013. p. 197–204.

[12] Iengo S, Rossi S, Staffa M, Finzi A. Continuous gesture recognition for flexible human-robot interaction. Proc. IEEE Int. Conf. on Robotics and Automation. 2014. p. 4863–8.

[13] Stolt A, Robertsson A, Johansson R. Robotic force estimation using dithering to decrease the low velocity friction uncertainties. Proc. IEEE Int. Conf. on Robotics and Automation. 2015. p. 3896–902.

[14] Colomé A, Pardo D, Alenyà G, Torras C. External force estimation during compliant robot manipulation. Proc. IEEE Int. Conf. on Robotics and Automation. 2013. p. 3535–40.

[15] Berger E, Grehl S, Vogt D, Jung B, Amor H. Experience-based torque estimation for an industrial robot. Proc. IEEE Int. Conf. on Robotics and Automation. 2016. p. 144–9.

[16] Vick A, Surdilovic D, Krüger J. Safe physical human-robot interaction with industrial dual-arm robots. Proc. 9th IEEE Int. Work. on Robot Motion and Control. 2013. p. 264–9.

[17] Cirillo A, Ficuciello F, Natale C, Pirozzi S. A conformable force/tactile skin for physical human-robot interaction. IEEE Rob Autom Lett 2016;1(1):41–8.

[18] Magrini E, Flacco F, De Luca A. Estimation of contact forces using a virtual force sensor. Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems. 2014. p. 2126–33.

[19] Magrini E, Flacco F, De Luca A. Control of generalized contact motion and force in physical human-robot interaction. Proc. IEEE Int. Conf. on Robotics and Automation. 2015. p. 2298–304.

[20] Magrini E, De Luca A. Hybrid force/velocity control for physical human-robot collaboration tasks. Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems. 2016. p. 857–63.

[21] Gaz C, Flacco F, De Luca A. Identifying the dynamic model used by the KUKA LWR: a reverse engineering approach. Proc. IEEE Int. Conf. on Robotics and Automation. 2014. p. 1386–92.

[22] Gaz C, De Luca A. Payload estimation based on identified coefficients of robot dynamics –with an application to collision detection. Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems. 2017. p. 3033–40.

[23] SYMPLEXITY. Symbiotic human-robot solutions for complex surface finishing operations. www.symplexity.eu.

[24] Coppelia Robotics. V-rep virtual robot experimentation platform. 2015. URL http://www.coppeliarobotics.com.

[25] Siciliano B, Sciavicco L, Villani L, Oriolo G. Robotics: modeling, planning and control. 3rd London: Springer; 2008.

[26] Universal Robots. Universal robots offline simulator. URL https://www.universal-robots.com/download/?option=28545#section16632.

[27] The URScript Programming Language. Universal Robots; 2015. Version 3.1.

[28] Swevers J, Verdonck W, De Schutter J. Dynamic model identification for industrial robots. IEEE Control Syst Mag 2007;27(5):58–71.

[29] Khalil W, Dombre E. Modeling, identification and control of robots. Hermes Penton London; 2002.

[30] Gautier M, Briot S. Global identification of drive gains parameters of robots using a known payload. Proc. IEEE Int. Conf. on Robotics and Automation. 2012. p. 2812–7.