Tool Compensation in Walk-Through Programming for Admittance-Controlled Robots

Chiara Talignani Landi, Federica Ferraguti, Cristian Secchi and Cesare Fantuzzi

***Abstract*— This paper describes a *walk-through programming* technique, based on admittance control and tool dynamics compensation, to ease and simplify the process of trajectory learning in common industrial setups. In the walk-through programming, the human operator grabs the tool attached at the robot end-effector and "walks" the robot through the desired positions. During the teaching phase, the robot records the positions and then it will be able to interpolate them to reproduce the trajectory back. In the proposed control architecture, the admittance control allows to provide a compliant behavior during the interaction between the human operator and the robot end-effector, while the algorithm of compensation of the tool dynamics allows to directly use the real tool in the teaching phase. In this way, the setup used for the teaching can directly be the one used for performing the reproduction task. Experiments have been performed to validate the proposed control architecture and a pick and place example has been implemented to show a possible application in the industrial field.**

I. INTRODUCTION

In the industrial field a lot of tasks can be executed automatically by means of robotic manipulators, e.g. pick and place, welding, assembly or product inspection, for this reason the way the trajectory is "taught" to the robot becomes crucial. Teaching techniques involving manual guidance of the robot [1] maximize the possibilities of re-programming the robot in a intuitive and easy way, because they don't need the knowledge of the robot programming language. To supervise the physical human-robot interaction (pHRI) during tasks requiring cooperation between the human operator and the robotic manipulator, with an efficient and cheap setup, compliant motion control strategies (i.e. admittance/impedance control) can be implemented [2]. The forces/torques arising during the interaction are measured by a Force/Torque (F/T) sensor typically mounted on the robot wrist flange. Thanks to this kind of controllers, the "walk-through programming" can be exploited: the operator becomes like a teacher that physically follows ("walks") the end-effector through the desired positions [3]. At the same time the robot's controller records the joints coordinates, "learning" the desired trajectory, and can play the trajectory back thereafter. In order to create an industrial setup, the tool attached at the robot end-effector has to be considered and its dynamics has to be compensated.

Several studies based on pHRI have been proposed in literature. In [4], [5] and [6] the weight of the payload mounted on the end-effector is considered and shared between the arm-manipulator and the human operator, implementing a compliant control. However, this work disregards the dynamics forces/torques generated by the motion of the tool. In [7] and [8] the idea of virtual tool to emulate the real one has been introduced and impedance is used to control the system. The walk-through technique for a welding process is addressed in [9] and [10], exploiting the impedance control with zero stiffness but without taking into account the tool emulation or compensation. In [11] an admittance control is implemented, considering a nonlinear model of the virtual tool's dynamics, with the same weight and inertial properties of the real tool. However, in some industrial applications, the teaching phase has to be performed directly using the real tool, in order to see the final result of the operation. In this case the end-effector of the robot may have to carry a not negligible payload. This condition requires to consider the dynamic model of the tool and to compensate its effects from the measurements of the F/T sensor. A solution to this problem was proposed in [12], where impedance control schemes are adopted and directly modified to embed the dynamic of the load. However, impedance control requires the knowledge of the robot dynamics and thus it is not suitable for industrial applications, where typically the dynamic model of the robot is unknown.

In this paper, we exploit the results obtained in [13] to directly compute and eliminate the non-contact forces due to the load dynamics and we include the tool compensation in the control architecture to obtain a walk-through programming technique, based on admittance control, that allows the use of the real tool in the teaching phase. During the teaching phase, the robot controller records all the significant poses of the trajectory followed by the human operator, and thus it will be able to interpolate them and play the trajectory back. In order to validate the control architecture, we implemented it on a robotic setup and we performed experiments that emulate a typical industrial application of pick and place as proof of concept.

II. CONTROL ARCHITECTURE AND PROBLEM STATEMENT

The walk-through programming is based on the physical human-robot interaction (pHRI), one of the most revolutionary and challenging features of the new generation of robots. This kind of robots have to safely interact with the human operator, thus interaction control strategies have to be implemented in order to guarantee a compliant behavior

Chiara Talignani Landi, Federica Ferraguti, Cristian Secchi and Cesare Fantuzzi are with the Department of Science and Methods for Engineering, University of Modena and Reggio Emilia, Italy. E-mail: {chiara.talignanilandi, federica.ferraguti, cristian.secchi, cesare.fantuzzi}@unimore.it

978-1-5090-3474-1/16/$31.00 ©2016 IEEE

of the robot. Impedance and admittance control strategies [2] describe the dynamical relations between force and displacement, but while the impedance control is more suitable for backdrivable robots, the admittance control is preferred when the robot has a stiff and non-backdrivable mechanical structure, which is the case of industrial robots.

In the admittance control, the desired position and orientation, together with the measured contact wrench, are input to an admittance equation which, via a suitable integration, generates the position and orientation to be used as a reference for a low-level motion controller. We assume that the motion is designed and tuned to minimize the tracking error and optimize the dynamic response. Thus we can assume that the actual pose $\boldsymbol{x} \in \mathbb{R}^n$, $n \leq 6$, of the robot end-effector is the same as the reference position $\boldsymbol{x}_{ref}$ [14] and in the following we will consider $\boldsymbol{x} = \boldsymbol{x}_{ref}$.

Formally, consider the Euler-Lagrange dynamic model of a n-DOF robotic manipulator, in the task space:

$$\boldsymbol{\Lambda}(\boldsymbol{x})\ddot{\boldsymbol{x}} + \boldsymbol{\mu}(\boldsymbol{x}, \dot{\boldsymbol{x}})\dot{\boldsymbol{x}} + \boldsymbol{F}_g(\boldsymbol{x}) = \boldsymbol{F}_\tau + \boldsymbol{F}_{ext} \tag{1}$$

where $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{q})$ is the end-effector pose, obtained from the joint values $\boldsymbol{q} \in \mathbb{R}^m, m \geq n$, through the direct kinematics $\boldsymbol{f}(\cdot)$. $\boldsymbol{F}_{ext} \in \mathbb{R}^n$ is the external wrench applied to the robot end-effector, while $\boldsymbol{F}_\tau \in \mathbb{R}^n$ is the torque due to the controlled joint torques $\boldsymbol{\tau} \in \mathbb{R}^m$. $\boldsymbol{\Lambda}(\boldsymbol{x}) \in \mathbb{R}^{n \times n}$ is the positive definite inertial matrix, $\boldsymbol{\mu}(\boldsymbol{x}, \dot{\boldsymbol{x}}) \in \mathbb{R}^{n \times n}$ is the matrix describing the centrifugal-Coriolis terms and $\boldsymbol{F}_g(\boldsymbol{x}) \in \mathbb{R}^n$ is the gravity vector.
The goal of the admittance control is to establish a desired dynamical relationship between the motion of the robot and the force applied by the environment. Thus, it is possible to force the robot to behave compliantly with the environment, according to a given mass-spring-damper system.

The elastic part of the mass-spring-damper system is used to attract the robot end-effector towards the desired pose. In this paper we want to address the case of an industrial robotic manipulator manually driven by the human operator. In this case, the user guides the robot by means of the force applied at its end-effector, without directly specifying a desired pose. Thus, the admittance control law of a mass-damper system can be written as:

$$\boldsymbol{\Lambda}_d\ddot{\boldsymbol{x}} + \boldsymbol{D}_d\dot{\boldsymbol{x}} = \boldsymbol{F}_{ext} \tag{2}$$

where $\dot{\boldsymbol{x}}(t)$ and $\ddot{\boldsymbol{x}}(t)$ are the velocity and acceleration, while $\boldsymbol{D}_d$ and $\boldsymbol{\Lambda}_d$ are, respectively, the desired damping and inertia n-dimensional symmetric and positive definite matrices.

The external force $\boldsymbol{F}_{ext}$ in (2) is assumed to be measured by a 6-DOF F/T sensor attached at the robot wrist flange. Moreover, we are considering case studies in which the robot is guided by the human through a desired trajectory, e.g. to pick objects by using a grasping tool of non-negligible weight attached to the robot after the F/T sensor, as shown in Fig. 1. In this case, the tool exerts a non-contact effect on the sensor, with both static (i.e. gravity) and dynamics (i.e. inertial, centrifugal/Coriolis) terms. Thus, the external force measured by the F/T sensor contains the sum of two terms: the non-contact wrench $\boldsymbol{F}_{nc}$ due to the load dynamics, and

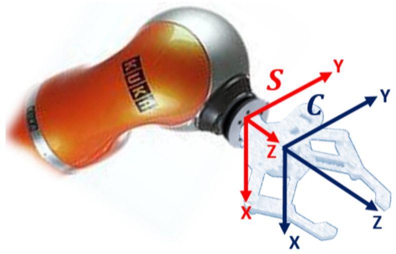


Fig. 1. Reference frames of the F/T sensor $\boldsymbol{S}$ and of the load (e.g. a grasping tool) $\boldsymbol{C}$. The axes of the tool reference frame are parallel to those of the sensor reference frame.

the contact wrench $\boldsymbol{F}_c$, arising from the interaction between the robot and the human operator or the environment.

Typically, the teaching phase in the walk-through programming is performed without considering the real tool, but exploiting a virtual tool that exhibits the same mechanical properties of the real tool [11] or compensating only the static part of the non-contact wrench [5]. Indeed, while the static terms of $\boldsymbol{F}_{nc}$ could be easily computed and eliminated from F/T measurements with a knowledge of the load mass, center-of-mass (COM) and orientation, compensating the dynamic effects of the load is much more complex, since the load inertia tensor and an estimate of linear acceleration/velocity must be available.

However, in some industrial applications the teaching phase has to be performed directly using the real tool, in order to see the final result of the operation, e.g. in welding applications, or to verify that the grasping tool accurately grasps the objects in pick and place applications. In these situations, where the real tool is involved and it is attached at the robot wrist flange, the effect of the non-contact wrench $\boldsymbol{F}_{nc}$ has to be computed and eliminated, since in the admittance model (2) the term $\boldsymbol{F}_{ext}$ embodies the force that has to be tracked by the admittance control (i.e. the force applied by the human when moving the robot end-effector to walk through the trajectory).

The aim of this paper is to propose a walk-through programming technique, based on admittance control, that includes the compensation of the static and dynamic terms of the load, and thus directly allowing the use of the real tool attached at the robot end-effector. The overall control scheme is shown in Fig. 2. The algorithm of tool compensation requires the knowledge of the robot linear acceleration ${}^S\hat{\boldsymbol{a}}$, angular acceleration ${}^S\hat{\boldsymbol{\alpha}}$ and angular velocity ${}^S\hat{\boldsymbol{\omega}}$, that can be estimated by using Kalman filters starting from the current pose of the robot end-effector $\boldsymbol{x}_{rob}$, as suggested in [13]. The superscript S indicates that all the values are referred to the reference frame of the F/T sensor. Then, the algorithm estimates the contact forces and torques $\hat{\boldsymbol{F}}_c$ starting from the raw measurements $\boldsymbol{F}_{ext}$ of the F/T sensor and $\left[\, {}^S\hat{\boldsymbol{a}},\ {}^S\hat{\boldsymbol{\omega}},\ {}^S\hat{\boldsymbol{\alpha}} \right]$ estimated by the Kalman filter. The admittance control, with input $\hat{\boldsymbol{F}}_c$, provides the reference position $\boldsymbol{x}_{ref}$ which the position-controlled robot must follow. If the industrial controller (low-level position control in Fig. 2) does not allow to directly feed the Cartesian position/orientation set-

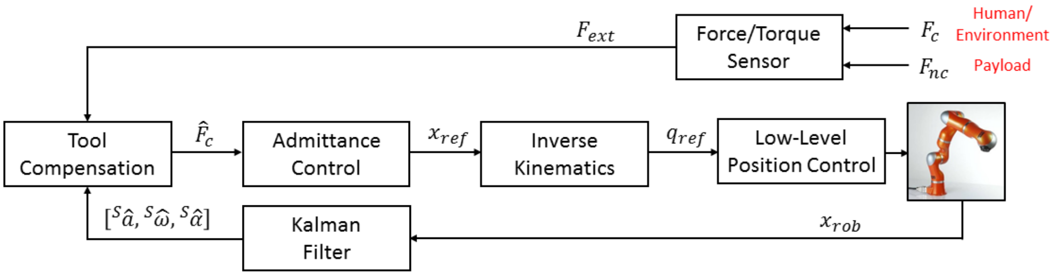


Fig. 2. The overall control scheme of the proposed walk-through programming technique based on admittance control and tool compensation.

points, a kinematic inversion has to be performed in order to convert the outputs of the admittance control into the joint position set-points $\boldsymbol{q}_{ref}$. During the teaching phase, the robot controller records all the significant poses of the trajectory followed by the human operator, and thus it will be able to interpolate them and play the trajectory back.

III. TOOL COMPENSATION

In order to compensate the static and dynamic effects of the tool mounted on the robot end-effector, it is required to relate the dynamic of the tool and its inertial parameters, i.e. mass, center of mass and inertia tensor, with the forces/torques measured by the F/T sensor. Even if the dynamic parameters may be calibrated online, as described in [15], we consider that in a typical industrial application the characteristics of the tool are known in advance, whether it is a tool designed by the system integrator of the robotic cell or an off-the-shelf object produced on a massive scale.

The non-contact forces/torques $\boldsymbol{F}_{nc} = [\,{}^S\boldsymbol{f}_{nc},\ {}^S\boldsymbol{\tau}_{nc}]^T$ in the sensor reference frame can be computed, according to the Newton-Euler formulation and to the motion of a rigid body due to external forces and torques, as follows [13]:

$$
\begin{aligned}
{}^S\boldsymbol{f}_{nc} &= m\,{}^S\boldsymbol{a} - m\,{}^S\boldsymbol{g} + {}^S\boldsymbol{\alpha}\times m\,{}^S\boldsymbol{c} + {}^S\boldsymbol{\omega}\times({}^S\boldsymbol{\omega}\times m\,{}^S\boldsymbol{c}) \\
{}^S\boldsymbol{\tau}_{nc} &= {}^S\boldsymbol{I}\,{}^S\boldsymbol{\alpha} + {}^S\boldsymbol{\omega}\times {}^S\boldsymbol{I}\,{}^S\boldsymbol{\omega} + m\,{}^S\boldsymbol{c}\times {}^S\boldsymbol{a} - m\,{}^S\boldsymbol{c}\times {}^S\boldsymbol{g}
\end{aligned} \tag{3}
$$

where m is the mass of the tool, ${}^S\boldsymbol{c} = [\,{}^Sc_x,\ {}^Sc_y,\ {}^Sc_z]$ is its center of mass, ${}^S\boldsymbol{I}$ is the inertia tensor, ${}^S\boldsymbol{g}$ is the gravity vector, ${}^S\boldsymbol{a}$ is the linear acceleration, ${}^S\boldsymbol{\omega}$ and ${}^S\boldsymbol{\alpha}$ are the angular velocity and acceleration, respectively. The inertia tensor in the sensor reference frame is defined as

$$
{}^S\boldsymbol{I} = \begin{bmatrix} {}^SI_{xx} & {}^SI_{xy} & {}^SI_{xz} \\ {}^SI_{xy} & {}^SI_{yy} & {}^SI_{yz} \\ {}^SI_{xz} & {}^SI_{yz} & {}^SI_{zz} \end{bmatrix} \tag{4}
$$

and it can be computed by directly applying the Huygens-Steiner theorem.

The equations (3) can be rewritten as follows

$$
\begin{bmatrix} {}^S\boldsymbol{f}_{nc} \\ {}^S\boldsymbol{\tau}_{nc} \end{bmatrix} = {}^S\boldsymbol{V}(\,{}^S\boldsymbol{a},\ {}^S\boldsymbol{\alpha},\ {}^S\boldsymbol{\omega},\ {}^S\boldsymbol{g})\ {}^S\boldsymbol{\varphi} \tag{5}
$$

in order to highlights the linear dependency of the non-contact forces/torques from the load inertial vector ${}^S\boldsymbol{\varphi}$, which is defined as:

$$
{}^S\boldsymbol{\varphi} = [m, m\,{}^Sc_x, m\,{}^Sc_y, m\,{}^Sc_z,\ {}^SI_{xx},\ {}^SI_{xy},\ {}^SI_{xz},\ {}^SI_{yy},\ {}^SI_{yz},\ {}^SI_{zz}]^T \tag{6}
$$

The matrix ${}^S\boldsymbol{V}(\,{}^S\boldsymbol{a},\ {}^S\boldsymbol{\alpha},\ {}^S\boldsymbol{\omega},\ {}^S\boldsymbol{g})$ is a 6×10 matrix including velocities, accelerations and elements of the gravity vector, and it is defined in (7). For sake of clarity, in the rest of the paper the dependency of ${}^S\boldsymbol{V}$ from $(\,{}^Sa,\ {}^S\alpha,\ {}^S\omega,\ {}^Sg)$ will be omitted. The matrix ${}^S\boldsymbol{V}$ does not consider the force drift typical of strain gage-based devices [17], which is a slowly time-varying and typically temperature-dependent contribution that may deteriorate the compensation results. To handle the force offset that could arise, the device can be periodically zeroed using a *tare* function embedded in the F/T sensor. To perform the tare operation, the tool should be detached from the end-effector. Since this operation has to be performed several times, it can be useful to include the compensation of the thermal drift into the compensation algorithm, in order to avoid the detachment of the tool for the taring. To this aim, the F/T sensor is zeroed when the tool is attached and in an initial static condition with a known orientation. Then, a pseudo-gravitational term (8) is added to subsequent readings in order to eliminate the gravitational forces and the associated torques.

$$
\boldsymbol{F}_{g_{init}} = {}^S\boldsymbol{V}(\boldsymbol{0},\boldsymbol{0},\boldsymbol{0},\ {}^S\boldsymbol{g_{init}})\ {}^S\boldsymbol{\varphi} \tag{8}
$$

As highlighted in (8) the components of the pseudo-gravitational vector $\boldsymbol{F}_{g_{init}}$ are computed as the product of the matrix ${}^S\boldsymbol{V}$, with only the gravity terms and zero accelerations and velocities, and the load inertial vector ${}^S\boldsymbol{\varphi}$, as follows:

$$
\begin{bmatrix} {}^S\boldsymbol{f}_{g_{init}} \\ {}^S\boldsymbol{\tau}_{g_{init}} \end{bmatrix} = \begin{bmatrix} m\ {}^Sg_x \\ m\ {}^Sg_y \\ m\ {}^Sg_z \\ m\ {}^Sc_y\ {}^Sg_z - m\ {}^Sc_z\ {}^Sg_y \\ -m\ {}^Sc_x\ {}^Sg_z + m\ {}^Sc_z\ {}^Sg_x \\ m\ {}^Sc_x\ {}^Sg_y - m\ {}^Sc_y\ {}^Sg_x \end{bmatrix} \tag{9}
$$

This zeroing operation should be repeated periodically, during a pause of the robotic task, in order to cope with the temperature variations and forces drift.

Since perfect knowledge of the inertial parameters of the tool cannot be achieved, all of the previous quantities should be considered as estimates, affected by a parametric uncertainty (i.e. ${}^S\hat{\varphi} = {}^S\varphi \pm \Delta\ {}^S\varphi$) and by estimation errors of the Kalman filter. Finally, the computation of contact F/T estimation can be performed as follows:

$$
\hat{\boldsymbol{F}}_c = \boldsymbol{F}_{ext} - \hat{\boldsymbol{F}}_{nc} + \hat{\boldsymbol{F}}_{g_{init}} \tag{10}
$$

$$
{}^{s}\boldsymbol{V} = \begin{bmatrix}
a_x - g_x & -\omega_y^2 - \omega_z^2 & \omega_x\omega_y - \alpha_z & \omega_x\omega_z + \alpha_y & 0 & 0 & 0 & 0 & 0 & 0 \\
a_y - g_y & \omega_x\omega_y + \alpha_z & -\omega_x^2 - \omega_z^2 & \omega_y\omega_z - \alpha_x & 0 & 0 & 0 & 0 & 0 & 0 \\
a_z - g_z & \omega_x\omega_z - \alpha_y & \omega_y\omega_z + \alpha_x & -\omega_y^2 - \omega_x^2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & a_z - g_z & g_y - a_y & \alpha_x & \alpha_y - \omega_x\omega_z & \alpha_z + \omega_x\omega_y & -\omega_y\omega_z & \omega_y^2 - \omega_z^2 & \omega_y\omega_z \\
0 & g_z - a_z & 0 & a_x - g_z & \omega_x\omega_z & \alpha_x + \omega_y\omega_z & \omega_z^2 - \omega_x^2 & \alpha_y & \alpha_z - \omega_x\omega_y & -\omega_x\omega_z \\
0 & a_y - g_y & g_x - a_x & 0 & -\omega_x\omega_y & \omega_x^2 - \omega_y^2 & \alpha_x - \omega_y\omega_z & \omega_x\omega_y & \alpha_y + \omega_x\omega_z & \alpha_z
\end{bmatrix} \quad (7)
$$

In order to cope with the uncertainties on the estimation of the contact input, a simple threshold-based logic is applied: if the estimate is lower than a fixed threshold it is considered as zero. However, fixed thresholds may require a trade-off between sensitivity and wrong contact detection. Adaptive thresholds, instead, would allow a better evaluation of model-based residuals (as essentially is the output of (10)), directly taking into account modeling errors [18]. In this case, we can exploit the linear structure of (5) to define an adaptive threshold as follows:

$$
\boldsymbol{F}_{th} = \mid {}^{s}\boldsymbol{V}(\,{}^{s}\boldsymbol{a},\ {}^{s}\boldsymbol{\alpha},\ {}^{s}\boldsymbol{\omega},\ {}^{s}\boldsymbol{g})\Delta\,{}^{s}\boldsymbol{\varphi} \mid + \boldsymbol{F}_{min} \qquad (11)
$$

where $\boldsymbol{F}_{min}$ is a fixed and empirically chosen safety margin. Thus, the threshold-based evaluation of the contact force can be written as follows:

$$
\hat{F}_c^i = \begin{cases} \hat{F}_c^i - F_{th}^i & \text{if } \hat{F}_c^i > F_{th}^i \\ \hat{F}_c^i + F_{th}^i & \text{if } \hat{F}_c^i < -F_{th}^i \\ 0 & \text{otherwise} \end{cases} \qquad (12)
$$

where $\hat{F}_c^i$ is the i-th component of the 6-DOF contact wrench $\hat{\boldsymbol{F}}_c$. As shown in Fig. 2, the vector computed in (12) is then used as the force input for the admittance control, which has to compute the pose set-point for the low-level controller.

IV. EXPERIMENTAL RESULTS

The proposed control scheme has been implemented and tested on a KUKA LWR 4+[1], provided with a 6 axis F/T sensor developed by the IIT (Italian Institute of Technology)[2]. Even though compliant control modes (i.e. cartesian and joint impedance) have been developed for this kind of manipulators, we used the embedded joint position control in order to simulate an industrial setup, where typically the low-level controller is a position control. All the software components have been developed using the Orocos real-time framework[3] and they run with a 500 Hz frequency: every 2 ms the external PC sends the current joint positions and the robot's controller records the trajectory. The object used to simulate the real tool is a hollow steel cylinder of 1.075 Kg of mass, 4 cm height, with 2.6 cm and 7.1 cm of internal and external diameters. A video clip showing the following experiments can be found at www.arscontrol.unimore.it/iecon2016.

A. Admittance control with and without tool compensation

Preliminary experiments have been performed to test the performance of the admittance control with and without the tool compensation.

[1]http://www.kuka-robotics.com/en
[2]http://wiki.icub.org/wiki/FT_sensor
[3]http://www.orocos.org/toolchain

After many experimental evaluations, the inertial and damping parameters of the desired interaction model (2) were empirically chosen as the following constant diagonal matrices:

$$
\begin{aligned}
\Lambda_d &= diag\{0.3, 0.3, 0.3, 0.005, 0.005, 0.005\} \quad [Kg] \\
D_d &= diag\{D_{d1}, D_{d2}\}
\end{aligned}
$$

where

$$
\begin{aligned}
D_{d1} &= diag\{60, 60, 60\} \quad [Ns/m] \\
D_{d2} &= diag\{0.2, 0.2, 0.2\} \quad [Nms/rad]
\end{aligned}
$$

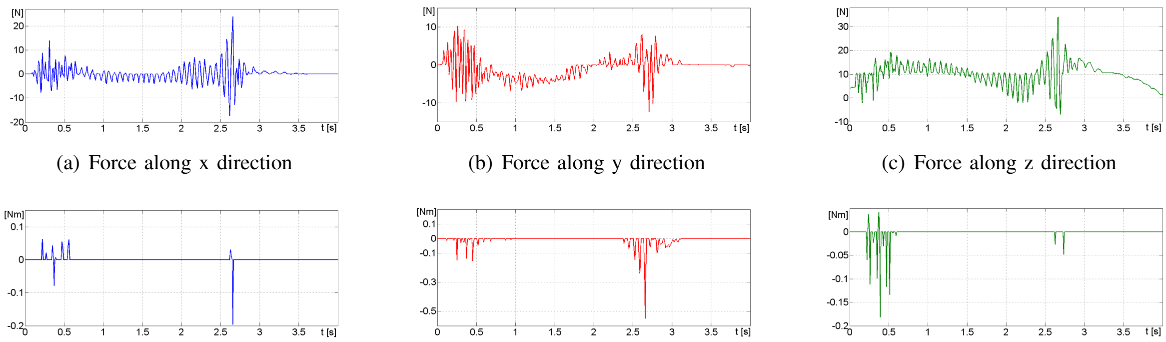
At first, the tool is simply attached at the robot end-effector and the standard admittance control is activated: the robot cannot support the weight of the tool, which falls down revealing an unstable behavior. Figure 3 shows the forces/torques (expressed in the world frame) measured during the test: the oscillating trend causes a lot of effort to the operator that is moving the robot.

Then, the admittance control with tool compensation is activated. The estimated contact forces and torques are smooth and damped (Fig. 4), resulting in a stable behavior of the robot end-effector and a smooth motion. Indeed, since the human interaction (detected by the F/T sensor) is the only input in the mass-damper model (2), the smooth F/T profile brings to a smooth motion of the robot.

As described in Section II, the non-contact forces/torques (shown in Fig. 5) are due to the tool mass, inertia, Coriolis and centrifugal terms and they have to be subtracted from the forces/torques measured by the F/T sensor in order to obtain the estimated contact wrench.

B. Walk-through programming for a pick and place application

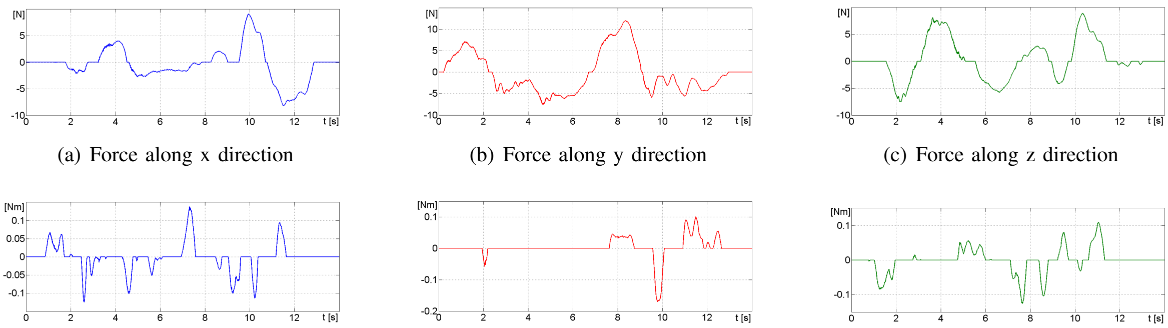
In order to provide an example of a typical industrial application which would benefit from the walk-through programming, based on admittance control and tool compensation, we emulate a pick and place application. The human operator grabs the tool, which in our experiment is the hollow steel cylinder while in the real case would be the grasping tool, and walk the robot through the poses where it has to take the objects and the location where they have to be placed. During the teaching phase, the robot records the end-effector poses and then it is able to play the trajectory back. Figure 6 shows a comparison between the poses recorded during the teaching phase (red lines) and those reached by the end-effector during the following reproduction (blue lines). To minimize the tracking error, the parameters of the low-level control have to be properly tuned.



(a) Force along x direction (b) Force along y direction (c) Force along z direction

(d) Torque around x axes (e) Torque around y axes (f) Torque around z axes

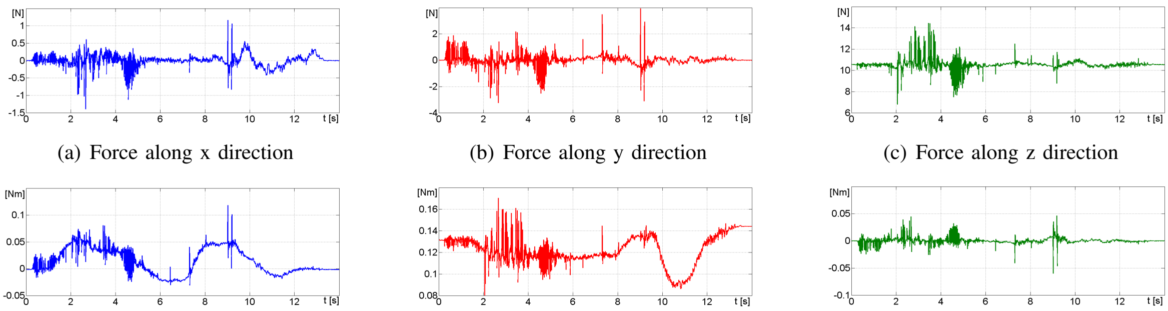
Fig. 3. Forces and torques without tool compensation.



(a) Force along x direction (b) Force along y direction (c) Force along z direction

(d) Torque around x axes (e) Torque around y axes (f) Torque around z axes

Fig. 4. Contact forces and torques with tool compensation.



(a) Force along x direction (b) Force along y direction (c) Force along z direction

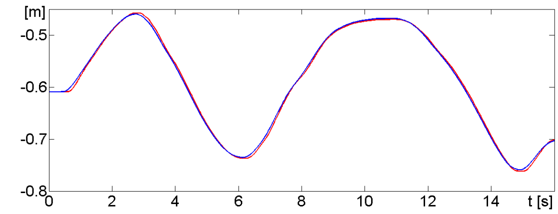
(d) Torque around x axes (e) Torque around y axes (f) Torque around z axes

Fig. 5. Non-contact forces and torques with tool compensation.

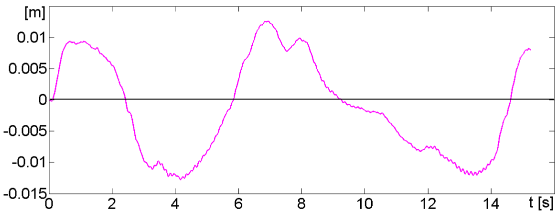
V. CONCLUSION

Common robot programming techniques involve defining the relevant robot positions and configurations, and writing the robot program to execute the task. In order to ease and simplify the programming process, new strategies of *walk-through programming* have been developed. In the walk-through programming, the human operator grabs the tool attached at the robot end-effector and physically moves the robot through the desired positions within the robot workspace. During this time, the robot's controller records the coordinate values on a fixed-time interval basis. These values and other functional information are then replayed in the automatic mode.

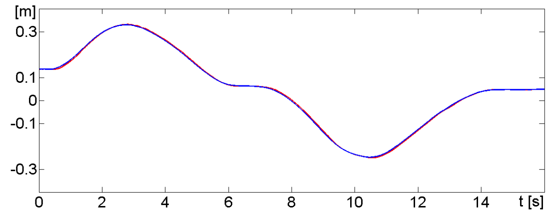
The teaching phase is currently performed by using the concept of virtual tool or compensating only the static effect of the tool attached at the robot end-effector. In this paper, we proposed a walk-through programming technique, based on admittance control with compensation of the tool dynamics. The admittance control guarantees a compliant behavior during the human-robot interaction, while the compensation of the tool dynamics allows to use the real tool directly in the teaching phase. The proposed control architecture has been validated on an experimental setup and an example of pick and place application has been provided. Future works aim at implementing the control architecture on a real industrial setup. To this aim, all the safety issues will be addressed in order to guarantee that the interaction between the user and the robot will be compliant with the current regulations.



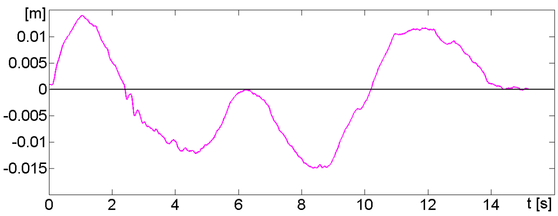
(a) Trajectory (x component)



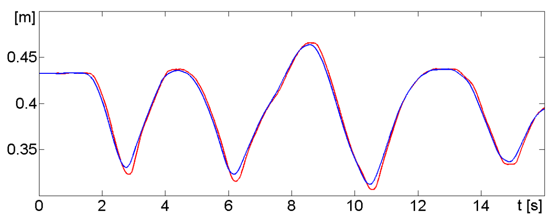
(b) Tracking error (x component)



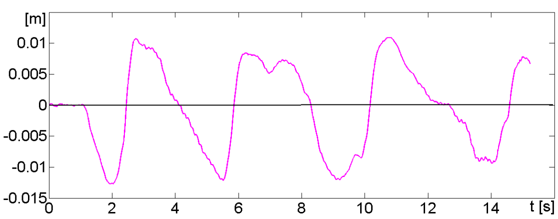
(c) Trajectory (y component)



(d) Tracking error (y component)



(e) Trajectory (z component)



(f) Tracking error (z component)

Fig. 6. Trajectory learned (red lines) and reproduced (blue lines) in the pick and place example and corresponding tracking errors.

Future works could investigate on the performance differences between the proposed method and the tool compensation directly implemented inside the admittance control.

References

[1] G. Ferretti, G. Magnani, and P. Rocco, "Assigning virtual tool dynamics to an industrial robot through an admittance controller," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 2009, pp. 1–6.

[2] L. Villani and J. De Schutter, "Force control," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer Berlin Heidelberg, 2008, ch. 7.

[3] A. K. Gupta and S. K. Arora, *Industrial Automation and Robotics*. Laxmi Publications, Daryaganj, New Delhi, 2007.

[4] O. M. Al-Jarrah and Y. F. Zheng, "Arm-manipulator coordination for load sharing using compliant control," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, MN, Apr. 1996, pp. 1000–1005.

[5] ——, "Arm-manipulator coordination for load sharing using reflexive motion control," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, Apr. 1997, pp. 2326–2331.

[6] ——, "Arm-manipulator coordination for load sharing using variable compliance control," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, Apr. 1997, pp. 895–900.

[7] K. Kosuge, Y. Fujisawa, and T. Fukuda, "Control of robot directly maneuvered by operator," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Tokyo, Japan, Jul. 1993, pp. 49–54.

[8] K. Kosuge and N. Kazamura, "Control of robot handling an object in cooperation with a human," in *Proceedings of the IEEE International Workshop on Robot and Human Communication*, Sendai, Japan, Sep. 1997, pp. 142–147.

[9] M. H. Ang Jr, W. Lin, and S. Lim, "A walk-through programmed robot for welding in shipyards," *Industrial Robot: An International Journal*, vol. 26, no. 5, pp. 377–388, 1999.

[10] M. H. Ang Jr, L. Wei, and L. S. Yong, "An industrial application of control of dynamic behavior of robots-a walk-through programmed welding robot," in *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, USA, Apr. 2000, pp. 2352–2357.

[11] L. Bascetta, G. Ferretti, G. Magnani, and P. Rocco, "Walk-through programming for robotic manipulators based on admittance control," *Robotica*, no. 7, pp. 1143–1153, Oct. 2013.

[12] F. Aghili and M. Namvar, "Scaling inertia properties of a manipulator payload for 0-g emulation of spacecraft," *International Journal of Robotics Research*, no. 7, pp. 883–894, 2009.

[13] D. Kubus, T. Kröger, and F. Wahl, "Improving force control performance by computational elimination of non-contact forces/torques," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Pasadena, USA, May 2008, pp. 2617–2622.

[14] F. Ferraguti, N. Preda, A. Manurung, M. Bonfe, O. Lambercy, R. Gassert, R. Muradore, P. Fiorini, and C. Secchi, "An energy tank-based interactive control architecture for autonomous and teleoperated robotic surgery," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1073–1088, 2015.

[15] D. Kubus, T. Kröger, and F. Wahl, "On-line estimation of inertial parameters using a recursive total least-squares approach," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, Sep. 2008, pp. 3845–3852.

[16] J. M. Selig, *Geometric Fundamentals of Robotics*. Springer, 2005.

[17] R. M. Voyles Jr, J. D. Morrow, and P. K. Khosla, "A comparison of force sensors," *Advanced manipulators laboratory, the Robotic Institute Carnegie Mellon University*, 1994.

[18] 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*, no. 6, 2014.