Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 11 (2017) 31 – 38

Procedia MANUFACTURING

27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27-30 June 2017, Modena, Italy

Walk-through programming for industrial applications

Federica Ferraguti[a,*], Chiara Talignani Landi[a], Cristian Secchi[a], Cesare Fantuzzi[a], Marco Nolli[b], Manuel Pesamosca[b*]

[a]*University of Modena and Reggio Emilia, via Amendola 2, 42122 Reggio Emilia, Italy*
[b]*Gaiotto Automation, via Toscana 1, 29122 Piacenza, Italy*

Abstract

Collaboration between humans and robots is increasingly desired in several application domains, including the manufacturing domain. The paper describes a software control architecture for industrial robotic applications allowing human-robot cooperation during the programming phase of a robotic task. The control architecture is based on admittance control and tool dynamics compensation for implementing walk-through programming and manual guidance. Further steps to integrate this system on a real set-up include the robot kinematics and a socket communication that sends a binary file to the robot.

Keywords: physical human robot interaction; cooperative robotics; admittance control; walk-through programming

1. Introduction

In the recent years, robots are becoming key elements to achieve manufacturing competitiveness, especially if they are able to collaborate with humans in a shared workspace, creating a co-working partnership. The initial paradigm for robot usage has changed during the years from an idea in which robots work with complete autonomy

* Corresponding author. Tel.: +39-0522-523531.
E-mail address: federica.ferraguti@unimore.it

doi:10.1016/j.promfg.2017.07.126

in a separate cell, to a scenario in which robots and humans can work together and cooperate. Traditional methods for robot programming involve the simulation/offline programming and the teach pendant use. In the first case, the knowledge of the robot language or of a 3D CAD program is needed and the programming phase can be long and time wasting, although different approaches to the same problem can be tested in a more efficient way. In the second case, a training for the use of the teach pendant is needed and its point-to-point programming becomes efficient only for simple movements.

New methods for robot programming can be exploited to ease the way the trajectory is taught to the robot. In particular, teaching techniques involving manual guidance of the robot [1] maximize the possibilities of re-programming the robot in an intuitive and easy way, because they don't need the knowledge of the robot programming language. Moreover, a high-level robot programming based on body and hand gestures recognition [2] has been developed to move the robot translating human movements into the robot ones.

Compliant motion strategies (i.e. admittance/impedance control) [3] can be implemented to supervise the physical human-robot interaction (pHRI) and to allow manual guidance of the robotic manipulator. Thanks to this kind of controllers, the "walk-through programming" can be exploited: the operator becomes like a teacher that physically guides the end-effector of the robot through the desired positions. During the teaching, the robot controller records the trajectory followed by the human and it will be able to play it 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. For example, in [4] and [5] the walk-trough programming has been exploited to to teach the robot welding paths in shipyards.

In this context, the safety of the human operator during the interaction must be preserved: in [6] three virtual walls are built around the robot to avoid collisions between the robot and the human body, superimposing virtual forces to the forces/torques measured by the sensor. Moreover, in [7] they present a safety framework for assembly tasks based on multiple Kinects for a 3D workspace simulation. This approach allows a real-time replication of the human and manipulator movements, generating safe goal positions for the robot.

In order to support the increasing request of pHRI introduction in the production industry, a new generation of industry-oriented robots has been launched on the market (e.g. KUKA LBR iiwa). Such robots embed advanced control functionalities, not available on standard industrial robots, including compliant control modes (from simple gravity compensation to Cartesian impedance control) and contact forces/torques estimation, possibly exploiting direct joint torque sensing. However, small and medium size enterprises may not have access to these technologies mainly due to the high costs of buying new robots dedicated to pHRI. Thus, a challenging issue would be to adapt the standard industrial manipulators with their non-backdrivable structures and stiff position controllers installed, to behave as collaborative robots, in order to extend pHRI to a larger part of industrial applications.

In [8, 9] a walk-through programming technique, based on admittance control and tool dynamics compensation is presented, to ease and simplify the process of trajectory learning and robot programming in common industrial setups. However, the control scheme has been implemented and tested on a KUKA LWR 4+ which allow manual guidance thanks to the embedded impedance controller and has a backdrivable structure. In this paper we exploit the results obtained in [8, 9] to implement a control architecture that allows human-robot cooperation during the programming phase of a robotic task. However, the whole architecture has been implemented in a real industrial setup, including an existing industrial robot that has to perform, e.g., a painting task.

2. 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. In pHRI taks humans and robots coexist in the same workspace and cooperate to perform a particular task. These robotic applications are suitable for the implementation of either impedance or admittance control schemes [3] since they require to regulate the interaction of the robot with its environment and also to cope with both free motion and "in contact" phases. Admittance control is more suitable when the robot has a stiff and non-backdrivable mechanical structure or is made purposefully stiff by an inner motion control loop, which is the common case in industrial robotics.

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 controller is designed and tuned to minimize the tracking error and optimize the dynamic response. Thus, we assume that the actual pose $\boldsymbol{x}(t) \in \mathbb{R}^n$ of the robot end-effector is the same as the reference position $\boldsymbol{x}_{\boldsymbol{ref}}(t) \in \mathbb{R}^n$ computed by the admittance control and in the following we will consider $\boldsymbol{x} = \boldsymbol{x}_{\boldsymbol{ref}}$.

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

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

where $\boldsymbol{x} = f(\boldsymbol{q})$ is the end-effector pose, obtained from the joint values $\boldsymbol{q} \in \mathbb{R}^m, m \geq n$, through the direct kinematics $f(\cdot)$. $\boldsymbol{F}_{\boldsymbol{ext}} \in \mathbb{R}^n$ is the external wrench applied to the robot end-effector, while $\boldsymbol{F}_{\boldsymbol{\tau}} \in \mathbb{R}^n$ 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 inertia matrix, $\boldsymbol{\mu}(\boldsymbol{x}, \dot{\boldsymbol{x}}) \in \mathbb{R}^{n \times n}$ is the matrix describing the centrifugal-Coriolis terms and $\boldsymbol{F}_{\boldsymbol{g}} \in \mathbb{R}^n$ 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 that can be expressed by the following differential equation:

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

where $\tilde{\boldsymbol{x}}(t) = \boldsymbol{x}(t) - \boldsymbol{x}_d(t)$ is the pose error, while $\boldsymbol{\Lambda}_d, \mathbf{D}_d$ and $\mathbf{K}_d$ are, respectively, the desired inertia, damping and stiffness n-dimensional symmetric and positive definite matrices.

The elastic part of the mass-spring-damper system in (2) is used to attract the robot end-effector towards the desired pose $\boldsymbol{x}_d(t)$. 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, we want to implement a particular case of the interaction model (2), where the robot end-effector behaves as a mass-damper system characterized by:

$$\boldsymbol{\Lambda}_d\ddot{\boldsymbol{x}} + \mathbf{D}_d\dot{\boldsymbol{x}} = \boldsymbol{F}_{\boldsymbol{ext}} \tag{3}$$

The external force $\boldsymbol{F}_{\boldsymbol{ext}}$ in (3) is assumed to be measured by a 6-DOF F/T sensor attached to the robot wrist flange. Moreover, we are considering case studies in which the end-effector is manually guided by the human to teach the robot a specific task, e.g. painting of an object. The example in Fig. 1 shows a spray gun attached to the robot end-effector, and a handle of non-negligible weight attached to the same end-effector, after the F/T sensor. 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 $\boldsymbol{F}_{\boldsymbol{nc}}$ due to the load dynamics, and the contact wrench $\boldsymbol{F}_{\boldsymbol{c}}$, arising from the interaction between the robot and the human operator or the environment.

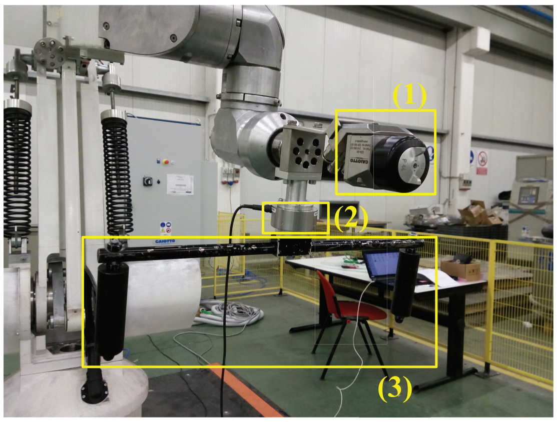

Fig. 1. Example of a tool that can be attached to the robot end-effector to perform an industrial task. (1) Spray gun (2) F/T sensor (3) Handle

Typically when the robot is programmed by using the walk-through techinque, the teaching phase is performed without considering the real tool, but exploiting a virtual tool that exhibits the same mechanical properties of the real tool [6] or compensating only the static part of the non-contact wrench [10]. 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 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 a painting task the amount of paint has to be directly monitored in order to verify the coverage of the whole surface. 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. Indeed, in the admittance model (3) the term $\boldsymbol{F_{ext}}$ is the force that has to be tracked and thus should embody only the force applied by the human when moving the robot end-effector to walk through the trajectory.

The goal of this paper is to present a software control architecture to obtain human-robot cooperation during the programming phase of a robotic task. The proposed control scheme allows the so-called walk-through programming or hand guidance, where the operator guides the end-effector of the robot through the desired positions to teach him the trajectory that will be played back thereafter. The walk-through programming has been implemented on industrial robots by other authors [6] and by companies like FANUC (i.e. by adding the “hand guidance” option to its collaborative robot CR-35iA) or ABB (i.e. by means of its FC Programming Handle add-on). However, these examples require the user to interact with the robot by means of specific handles embedding a F/T sensor, but the latter is not sensitive to the robot payload. In our case, instead, the setup including the real tool could also be used since the static and dynamic effects of the tool are compensated. In this way, the control architecture is suitable to execute industrial tasks requiring a controlled interaction between payload and environment (e.g. polishing, assembling, etc.) or to perform tasks that require to constantly check the final result of the operation (e.g. painting). In order to validate the presented strategy, the integration steps between the theoretical system and the real robot are presented, and customized for the GA-2000 Gaiotto Robot.

The overall control scheme is shown in Fig. 2. The algorithm of tool compensation requires the knowledge of the robot linear acceleration $\boldsymbol{\hat{a}}$, angular acceleration $\boldsymbol{\hat{\alpha}}$ and angular velocity $\boldsymbol{\hat{\omega}}$. These data can be estimated by using Kalman filters starting from the current pose of the robot end-effector $\boldsymbol{x}$, as suggested in [3]. Then, the algorithm estimates the contact forces and torques $\boldsymbol{\hat{F}_c}$ starting from the raw measurements $\boldsymbol{F_{ext}}$ of the F/T sensor and $[\boldsymbol{\hat{a}}, \boldsymbol{\hat{\alpha}}, \boldsymbol{\hat{\omega}}]$ estimated by the Kalman filter. The admittance control, with input $\boldsymbol{\hat{F}_c}$, 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 setpoints, 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 of the walk-through programming, 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.

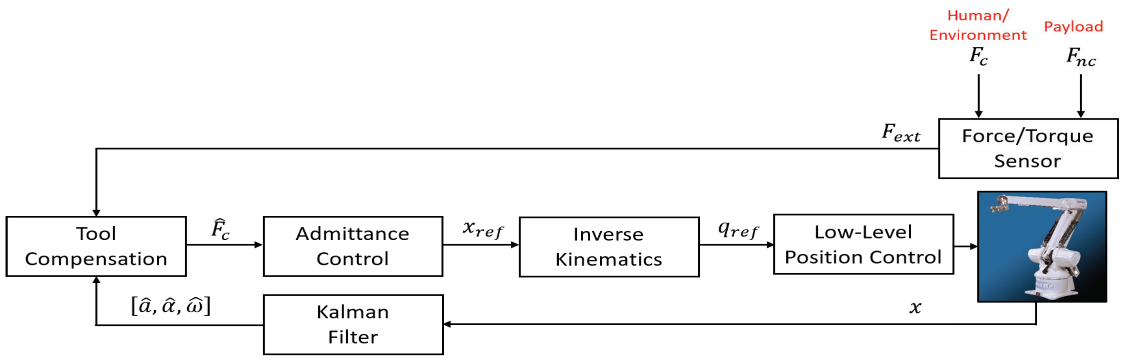

Fig. 2. Overall control architecture: walk-through programming technique based on admittance control and compensation of the tool dynamics

3. 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 [11], 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 $\boldsymbol{F_{nc}} = [\boldsymbol{f_{nc}}, \boldsymbol{\tau_{nc}}]$ 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 [12]:

$$\begin{aligned} f_{nc} &= m\,\boldsymbol{a} - m\,\boldsymbol{g} + \boldsymbol{\alpha}\times m\,\boldsymbol{c} + \boldsymbol{\omega}\times(\boldsymbol{\omega}\times m\,\boldsymbol{c}) \\ \boldsymbol{\tau}_{nc} &= \boldsymbol{I}\,\boldsymbol{\alpha} + \boldsymbol{\omega}\times\boldsymbol{I}\,\boldsymbol{\omega} + m\,\boldsymbol{c}\times\boldsymbol{a} - m\,\boldsymbol{c}\times\boldsymbol{g} \end{aligned} \tag{4}$$

where m is the mass of the tool, $\boldsymbol{c} = [c_x, c_y, c_z]$ is its center of mass, $\boldsymbol{I}$ is the 3×3 inertia tensor, $\boldsymbol{g}$ is the gravity vector, $\widehat{\boldsymbol{a}}$ is the linear acceleration, $\widehat{\boldsymbol{\omega}}$ and $\widehat{\boldsymbol{\alpha}}$ are the angular velocity and acceleration, respectively.

The equations (4) can be rewritten as follows

$$\begin{bmatrix} \boldsymbol{f_{nc}} \\ \boldsymbol{\tau_{nc}} \end{bmatrix} = \boldsymbol{V}(\boldsymbol{a}, \boldsymbol{\alpha}, \boldsymbol{\omega}, \boldsymbol{g})\,\boldsymbol{\varphi} \tag{5}$$

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

$$\boldsymbol{\varphi} = [m, m\,c_x, m\,c_y, m\,c_z, I_{xx}, I_{xy}, I_{xz}, I_{yy}, I_{yz}, I_{zz}] \tag{6}$$

The matrix $\boldsymbol{V}(\boldsymbol{a}, \boldsymbol{\alpha}, \boldsymbol{\omega}, \boldsymbol{g})$ 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 $\boldsymbol{V}$ from $(\boldsymbol{a}, \boldsymbol{\alpha}, \boldsymbol{\omega}, \boldsymbol{g})$ will be omitted.

The matrix $\boldsymbol{V}$ does not consider the force drift typical of strain gage-based devices [13], 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 and not always the tool can be easily detached from the end-effector, it can be useful to include the compensation of the thermal drift into the compensation algorithm. 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 (7) is added to subsequent readings in order to eliminate the gravitational forces and the associated torques.

$$\boldsymbol{F}_{\boldsymbol{g}_{init}} = \boldsymbol{V}(\boldsymbol{0}, \boldsymbol{0}, \boldsymbol{0}, \boldsymbol{g}_{init})\,\boldsymbol{\varphi} \tag{7}$$

This offset zeroing operation should also be repeated periodically, during a pause of the robotic task, to cope with temperature variations and slow drifts.

Since perfect knowledge of the inertial parameters cannot be achieved and there could be estimation errors of the Kalman filter, the computation of contact F/T is based on uncertain quantities. These uncertainties can be compensated by applying threshold-based logics as, e.g., the one provided in [3]. Therefore, the expression of contact F/T can be written as follows:

$$\widehat{\boldsymbol{F}}_{\boldsymbol{c}} = \boldsymbol{F}_{\boldsymbol{ext}} - \widehat{\boldsymbol{F}}_{\boldsymbol{nc}} + \widehat{\boldsymbol{F}}_{\boldsymbol{g}_{init}} \tag{8}$$

4. Software Implementation

From a practical point of view, the implementation of the described software architecture is not as immediate as it can be for a collaborative robot, whose interface with the low level controller is meant to be easily programmable and accessible.

A few steps have to be integrated in the system described in Fig. 2, and they strictly depend on the robot model and kinematics. In our case study, all the parameters were tuned to integrate the control architecture on the GA-2000 Gaiotto robot. To reduce the computational effort required to the robot controller, we implemented the overall control architecture on an external pc provided with Ubuntu 14.04. We exploited Orocos Real-Time Toolkit (RTT)[1] and Kinematics/Dynamics Library (KDL) to create software components, whose update frequency was limited to 200 Hz, due to robot hardware limitations. We chose Orocos since it is an open source project, modular and flexible, whose advantage consists of creating a component-based architecture, where the software components are independent and they can be arbitrarily linked through data ports.

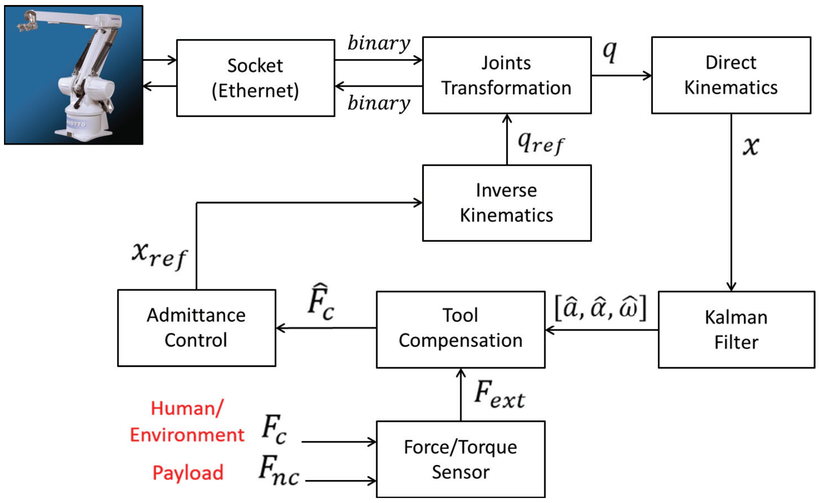

Fig. 3. Software control architecture: walk-through programming technique implemented on Gaiotto GA-2000 robot

As depicted in Fig. 3, the robot sends its current joint position to a Joints Transformation component through the Ethernet socket communication. The robot sends its data in a binary form, which correspond to the encoder steps per revolution of each motor. Hence each value has to be divided by a factor of $(4096 * 61)/360$ for the first three joints, and by a factor of $(4096 * 31)/360$ for the other three, in order to obtain joint angles, expressed in degrees and then converted to radians to execute the following steps.

Here, preprocessing operations on the six joints array are performed: the robot offsets have to be subtracted from the resulting angles and the values referred to the second/third and four/fifth joint have to be decoupled.

From a mechanical point of view, the coupling/decoupling operation is necessary since the second and third axis are mechanically bounded together, as well as the fourth and fifth ones. This means that a movement on the fourth joint reflects in a movement on the fifth one and, to avoid that, the motor on the fifth joint has to be moved about the opposite quantity to keep it fixed. The same procedure has to be applied to the second and third axis.

The Direct Kinematics has to be performed to obtain the corresponding Cartesian pose. Since Gaiotto kinematics returns a value referred to the fifth axis, a translation matrix has to be added as a final step to obtain the Cartesian pose with respect to the end-effector, making it coherent with the rest of the system.

[1] http://www.orocos.org/

This pose is then used by the Force Sensor component to execute the rotation of the contact wrench from the sensor reference frame to the world one, and by the admittance control to set the reference initial position. Furthermore, the Kalman Filter [9] uses the Cartesian position to compute the linear acceleration, angular acceleration and angular velocity. Then, the Sensor component, the Tool Compensation and the Admittance Control perform as described in Section 2.

The Cartesian pose computed by the Admittance Control has to be transformed in a joint position by the Inverse Kinematics of the robot GA-2000 and, through the Joint Transformation, robot offsets and joint coupling are performed, and a final binary transformation on each joint value is done. Finally, these set points are transmitted to the robot through the socket communication.

The robot controller itself handles the recording of the trajectory and the painting data (i.e. control of the spray gun, the fan and the flow of the paint), which are directly managed by the human operator through the buttons placed on the handle.

Since the focus of the system resides in the force/torque sensor and the robot is primarily used for spray painting, we chose the ATI Mini45 with a 68 International Protection, rated for submersion in fresh water to a depth of 4 m[1].

To obtain a better monitoring of the device, the robot and the external pc exchange some useful information about the sensor state, the maximum force that can be applied and if there is coherence between the force and the movement (i.e. if the force perceived is zero, the movement has to be null as well, and vice versa).

Since safety is of paramount importance when dealing with physical Human-Robot Interaction, a dedicated PLC system and the “dead man” on the handle are integrated, with respect to the ISO EN 10218 regulations.

5. Conclusions

In this work we presented a way to adapt and integrate the control architecture of a walk-through programming technique on an industrial robot. First, the overall control scheme is presented in combination with the payload compensation, to allow the human operator to directly use the real tool attached to the robot end-effector during the teaching phase. To integrate the system on an industrial robot, several considerations have to be taken into account: first of all the robot controller must communicate with the external system with a minimum frequency of 250 Hz, in order to guarantee a stable behavior of the admittance control. To reduce the computational effort required to the robot, we developed the overall architecture on an external pc: the inverse kinematics has been moved from the robot to the external system, along with the joints coupling, coherently with the mechanical bounds of the axis, and the binary conversion for the electric motors. Hence, the binary values are directly sent to the robot through the Ethernet socket, reducing its elaboration time and guaranteeing a real-time application.

A crucial aspect of the industrial painting is to perform the task with high velocities (2 m/s): the main challenge of our walk-through system will be to create a stable and reliable set-up that can satisfy this requirement.

Future works aim at implementing the overall system in real time on the GA-2000 Gaiotto robot, creating a generalizable software architecture that can convert any industrial robot into a collaborative one.

References

[1] G. Ferretti, G. Magnani, P. Rocco, “Assigning virtual tool dynamics to an industrial robot through an admittance controller”, IEEE International Conference on Robotics and Automation. Kobe, Japan. May, 2009.

[2] P. Tsarouchi, A. Athanasatos, S. Makris, X. Chatzigeorgiou, G. Chryssolouris. “High Level Robot Programming Using Body and Hand Gestures”. Procedia CIRP, 2016.

[3] L. Villani, J. De Schutter. “Force control”, Springer Handbook of Robotics, B. Siciliano and O. Khatib, Eds. Springer Berlin-Heidelberg, 2008, chapter 7.

[4] M. H. Ang Jr., W. Lin, S.Y. Lim. "A walk-through programmed robot for welding in shipyards." Industrial Robot: An International Journal, pp. 377-388, 1999.

[5] 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.

[1] http://www.ati-ia.com

[6] L. Bascetta, G. Ferretti, G. Magnani, P. Rocco. “Walk-through programming for robotic manipulators based on admittance control”, Robotica, pp. 1143-1153, July 2013.

[7] C. Morato, K. N. Kaipa, B. Zhao, S. K. Gupta. “Toward Safe Human Robot Collaboration by using Multiple Kinects based Real-time Human Tracking”, Journal of Computing and Information Science in Engineering, January 2014.

[8] C. Talignani Landi, F. Ferraguti, C. Secchi, C. Fantuzzi. "Tool compensation in walk-through programming for admittance-controlled robots", 2016 Annual Conference of the IEEE Industrial Electronics Society. Firenze, Italy. October, 2016.

[9] S. Farsoni, C. Talignani Landi, F. Ferraguti, C. Secchi, M. Bonfè. "Compensation of load dynamics for admittance controlled interactive industrial robots using a quaternion-based Kalman filter", IEEE Robotics and Automation Letters, vol. PP, no.99, pp.1-1

[10] O. M. Al-Jarrah, Y. F. Zheng, "Arm-manipulator coordination for load sharing using reflexive motion control", IEEE International Conference on Robotics and Automation. Albuquerque, New Mexico. April, 1997.

[11] D. Kubus, T. Kröger, F. Wahl. “On-line estimation of inertial parameters using a recursive total least-squares approach”, IEEE/RSJ International Conference on Intelligent Robots and Systems. Nice, France. September, 2008.

[12] D. Kubus, T. Kröger, F. Wahl. “Improving force control performance by computational elimination of non-contact forces/torques”, IEEE International Conference on Robotics and Automation. Pasadena, USA. May, 2008.

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