

# PC BASED CONTROL SYSTEMS FOR COMPLIANCE CONTROL AND INTUITIVE PROGRAMMING OF INDUSTRIAL ROBOTS

**D. Colombo**  
**I.T.I.A. C.N.R.**  
**Milano (Italy)**

**D. Dallefrate**  
**I.T.I.A. C.N.R.**  
**Milano (Italy)**

**L. Molinari Tosatti**  
**I.T.I.A. C.N.R.**  
**Milano (Italy)**

**Speaker: Davide Colombo, ITIA – CNR, Italy, d.colombo@itia.cnr.it**

**Topic: Robots for SME manufacturing: SMErobot**

**Keywords: PC-based control, open source, compliance control**

## **Abstract**

Nowadays, high costs of automation technologies, sustainable only in manufacturing markets with large volume production and intensive capitals, do not allow a widespread application of Robotics to Small Medium Enterprises (SME). To face such a problem the need of low-cost and not-proprietary solutions at all level of the robot work cell is mandatory. Focusing on the robot control system, two aspects can be addressed to reduce its life cycle costs. Firstly, during the Research & Development activities, more openness allows to design, develop and integrate functionalities quickly, reacting to market changes with few engineering efforts both in term of cost and time; secondly, during the common use of the control system, providing intuitive interfaces for robot programming let the robot end user not to rely on high skilled operators who require specific knowledge for each specific system with expensive efforts in personnel training. In this paper, the above mentioned aspects are addressed proposing a PC and open source software based control architecture for robot controllers. Such an architecture has been used to test compliance strategies; specifically, this work shows the implementation of teaching by demonstration based on manual guidance using a force/torque (F/T) sensor and a wireless joy-pad. The same algorithms used on the PC-based controller have been applied to an industrial robot controller in order to accomplish a manual guidance based on the motor currents feedback.

## **1. Introduction**

During the last decade, the openness of control systems has been addressed in several ways by worldwide research projects both in the field of machine tools (i.e. OSEC/JOP, OMAC, OSACA) [1] and in the field of Robotics (i.e. OROCOS) [2]. Although a universal agreement on definition of Open Control Architecture (OAC) has been not yet achieved, a basic set of common requirements has been drawn and they can be summed up as follows:

- an OAC allows the portability of software on different hardware platforms and operating systems;
- an OAC is modular having the capability of easy replacement of modules with new ones, allowing scalability both in performances and functionalities
- an OAC allows easy integration of new functionalities. This means, firstly, that it is based on standards to let third parties develop hardware and software that meet new requirements; secondly, it means that an OAC provides plug-and-play mechanisms for fast integration of such new functionalities
- an OAC is reconfigurable providing mechanisms of easy adaptation of its parameters in order to customize the system to different application scenarios
- an OAC is economic since it is based on off-the-shelf modules both with reference to hardware and software

Even if many control vendors offer integrated development tools to customize their systems, such tools are often based on proprietary solutions. In this case, providing customizations for specific applications [3] requires great engineering efforts in term of knowledge, time and cost. On the other hand, controls based on PC hardware cover mostly the requirements of an OAC. Main reasons include the increasing power of PC microprocessors that make them suitable for real-time applications, economies of scale of the PC market that reduce their cost, the easy integration of PCs in enterprise network that fulfils the need of accurate manufacturing information at all enterprise level. Another key point for a modern OAC is the use of open source software (OSS), which has attracted recently also the field of numerical controls for machine tools [4]. One of the most promising use of OSS in OAC is at operating system level: firstly, OSS has the advantage of full code

sources availability that allows developers to provide their own customization [5]; secondly, OSS allows to reduce cost of the whole controller cutting the license fee of commercial RTOS.

For the above mentioned reasons, since some years Institute of Industrial Technologies and Automation (ITIA) of the Italian National Research Council (CNR) has been developing PC and OSS based control systems for robots [6] [7].

Specifically, exploiting the modularity and reconfigurability of the developed control, such system is now used to implement and test compliance control strategies in order to allow intuitive programming of robots, i.e. Programming by Demonstration (PbD), also known in literature as, 'Walk-through Approach', 'Computer Assisted Teach and Play' or 'Programming by Manual Guidance'. PbD is needed to transfer human skills to the robot. These abilities can be directly used to repeat a task as it was observed or combined with other similar performances of the task in order to learn a more general account of the skill.

Specifically, the focus is on manual guidance [8] which means that the human operator can drive the robot applying forces and torques to the tip or other parts of the robot itself. In this way when the robot is moved to each desired position, the controller records its joint coordinates. This information is collected to process a program composed by a sequence of vectors with joint coordinates and some activation signals for external equipment such as the gripper aperture. Then, when the robot is in automatic mode and the recorded program is executed, the robot moves through the specified sequence of joint coordinates, reproducing the indicated signals.

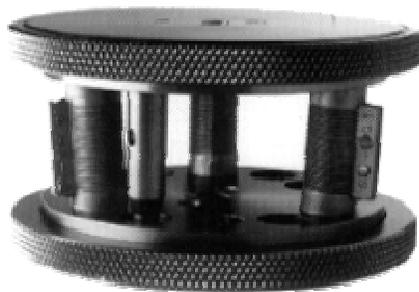
This work is organized as follows: in Section 2 compliance strategies for PbD are revised with reference to the literature underlying the algorithms adopted in the following sections. In section 3 an overall view of the robot control architecture designed by ITIA-CNR is shown. With reference to such an architecture, in Section 4 the software implementation is detailed focusing on the application of PbD. In Section 5 the specific results regarding PbD of, firstly, a 7 d.o.f robot arm controlled by means of an F/T sensor and, secondly, of a standard industrial robot controlled by means of motor currents feedback (F/T sensor-less manual guidance) are described; then in section 6 final conclusions and future works are drawn.

## 2. Compliance strategies for Manual Guidance

Manual guidance is a programming method that was widely used in the past especially for painting robots. Such systems were mechanically balanced; this means that the method used for their manual guidance did not use any external sensor: the robot was completely passive while the operator had to compensate its inertia.

Nowadays, standard industrial robots are not mechanically compensated, so their manual guidance needs the use of proper feed-back systems to get forces/torques applied by the operator to the robot. Furthermore, the success of PbD by means of Manual Guidance [10] and, generally speaking, the success of human-robot or environment-robot interaction, relies mainly on the planning accuracy and the control performance. The implementation of manual guidance needs the use of compliance control strategies. Such control algorithms transform the robot from a passive and hard mechanism only capable of repeating a programmed operation, to an active mechanism capable of adapting to environment modifications. To understand the property of the Compliance strategies it's easily to think how is essential the tactile information for a human, in the such way the force information managed from the compliance strategies gives to the robot a real intelligent way to have an entirely success in the task approach. Compliance control strategies are basically divided in two groups: active and passive compliance.

Passive compliance uses flexible mechanical devices placed on the tip of the robot. Related to an assembly task, these devices adjust the position of the manipulated object when an interaction force has been applied. The most common device is the remote centre of compliance (RCC) (Figure 1) [9]. The external forces and torques applied on this device produce a translation and a rotation about a point called "accommodation centre". The stiffness of the RCC is mechanically configurable. Improvements in spatial alignments of prismatic parts of general cross section have been obtained by using the Spatial Remote Centre of Compliance SRCC [11]. These passive devices give flexibility to the robot, but don't allow an on line reconfiguration of the stiffness parameters.



*Figure 1 Remote centre of compliance*

Active compliance methods are based on active device, such as F/T sensors, to measure the interaction with the robot. The information is collected and elaborated through force control strategies to make the robot compliant with the environment. In general the compliance could be defined according to a force or a stiffness value.

Several control schemes has been studied in scientific literature including generalised damping control stiffness control impedance and admittance control, hybrid position-force control, parallel force-position control and hybrid impedance control. A detailed analysis of compliance algorithms could be found in the work of De Schutter and Bruyninckx [13] and of Sciavicco and Siciliano [14], hereunder the most successful strategies are briefly described.

The Hybrid position-force control subdivides the Cartesian space in two subspace, i.e. constrained and unconstrained space, allowing, in the former, the force control and, in the latter, the position control. In the constrained space the force controller allows the robot to follow a desiderate force target and, as the position control, could be designed as a PID control. The parallel Force/Position makes use of a cascade of position and force control strategies. The force controller has a higher priority than the position controller, so an error in velocity would be tolerated in order to regulate the force level.

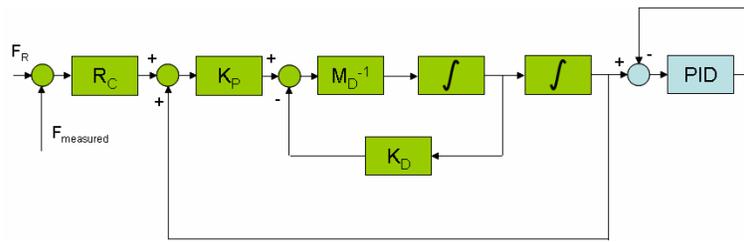


Figure 2 Force Control Strategy

Impedance and admittance control strategy provides a dynamic relationship between the robot position and the force it exerts. The basic idea of admittance control is that the arm can be controlled as it behaves as mechanical impedance (mass, spring and damper) with positional constraints imposed by the environment.

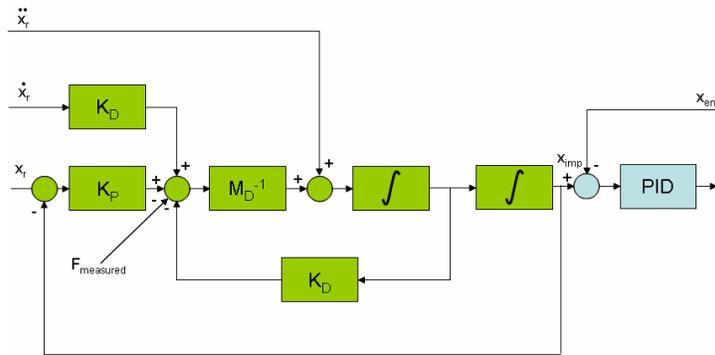


Figure 3 Impedance Control Strategy

The method of manual guidance based on admittance control strategy has been applied in our test case, the 7-d.o.f. robot arm PA-10 manufactured by Mitsubishi. Details of the implementation of this strategy are reported in section 5.1. The same strategy, but without using a force/torque sensor, has been applied to an industrial robot (NS16) manufactured by COMAU (section 5.2).

### 3. Architecture design of the control system

According with the OAC definition (section 1), the main design goals of the proposed architecture (Figure 4) are the follows.

- Portability. In order to run the software on different hardware platforms and operating system, a software engineering to decouple control functionalities and real time operating system (RTOS) dependent services has been adopted.
- Modularity. A functional decomposition of the control software allows easy replacement of functionalities with new ones just changing module implementations.

- Reconfigurability. The main modules have been parameterized then an easy customization of the control architecture with reference to different kinematics architectures is feasible.
- Cost effectiveness. Off-the-shelf modules both with reference to hardware and software have been used in order to reduce costs of the whole system.

The proposed architecture is basically divided in two layers: an application layer that contains modules needed by a robotic controller and a system layer where main functionalities of an RTOS are encapsulated. This kind of architecture allows to reach the first design goal. Specifically, the system layer encapsulates functionalities of task management and inter process communication, providing three modules: task, queue and shared memory. The Task provides functionalities to handle an RT process; it allows the task configuration, setting basic properties such as id, priority, sample time, the task activation and deactivation (i.e. make periodic, run, suspend, resume, kill); finally, it provides diagnostic functions to check the task state. The Queue and the Shared Memory modules encapsulate respectively the mechanisms of the same name for the communication between different tasks.

The application layer, according with the second design goal, is made up by 6 main modules: interpreter, trajectory generator, kinematics, F/T manager, axis control and axis driver. Furthermore, each module provides a set of parameters that allows the customization with reference to the kinematics architecture of the controlled robot. Hereunder such modules are described in details.

The Interpreter translates high-level motion commands specified by the user into an inner data format (i.e. motion data structure) suitable for further elaborations of the controller described below.

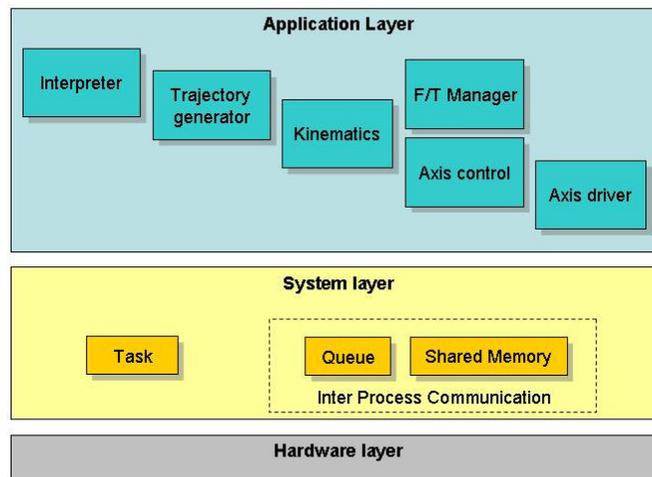


Figure 4 Reference architecture

The Trajectory Generator acts through three different levels: first, it provides the path planning generating a Cartesian trajectory according to the geometrical data specified in the motion data structure prepared by the Interpreter; then, it performs a sampling of such a path according to a given motion law; finally it calls services provided by the kinematics module to transform end effector movements to actuators coordinates set points.

The Kinematics includes algorithms to solve transformations from Cartesian to joint space of the controlled robot and vice versa and to check robot workspace and limits. An easy reconfiguration of this module is a basic requirement in order to apply rapidly the controller to different robots achieving one of the goals of an OAC (chapter 1); such a reconfiguration can usually be done in two ways: by means of the definition of a general set of parameters that describe a wide range of kinematics architectures; by means of the implementation of different kinematics modules, each one specific for one kind of architecture. Specifically, this is the design rule followed in our reference architecture. In this case reconfiguration is done by the substitution of a kinematics module with another one. This operation is feasible without effort only if all kinematics modules implement the same interfaces.

The F/T manager collects the information data given from the I/O board connected to the F/T, concerning the force, the torque and temperature values sensor. The information in raw-mode is translated and combined to describe the interaction forces on the robot tip.

The Axis Controller keeps the actual actuators coordinates as close as possible to the corresponding set points by means of proper control algorithms.

The Axis Driver is the software interface to the actual I/O board that allows bi-directional data communication between the controller and the robot; specifically, it allows the reading of sensors signal (encoders, analog and digital inputs) and the writing of command set points.

#### 4. Implementation of the control system

The implementation of the architecture described in section 3 is made in C++ on a PC based hardware; it follows the scheme proposed in Figure 5. Specifically, the architecture is composed by the following six main processes interacting in the robot system control:

- CMD interpreter process;
- TrjGen process;
- pControl process;
- MC server process;
- HMI Server process.
- F/T Sensor manager process;

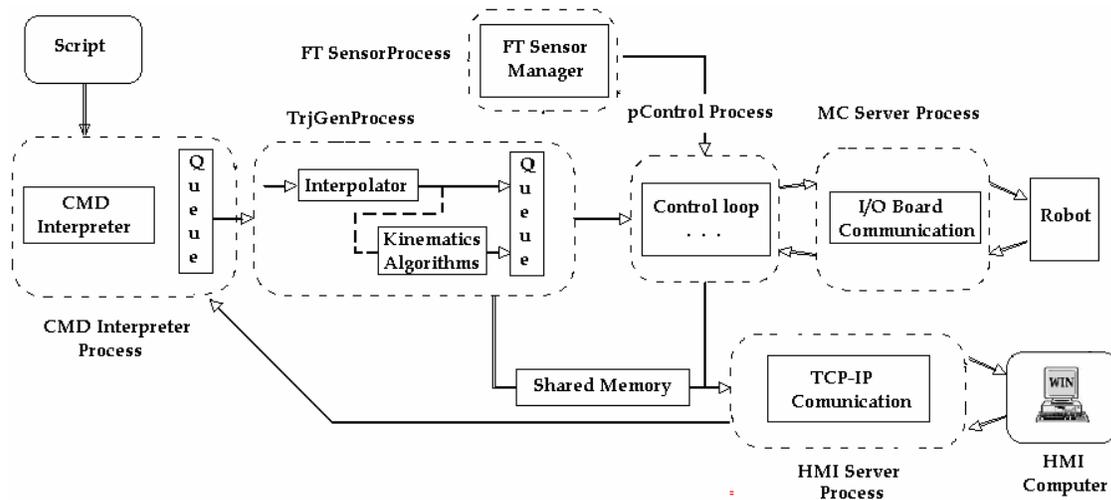


Figure 5 Control architecture

The CMD interpreter process receives from the shell or from the HMI Server process robot scripts. A robot script is a list of high level commands that can be expressed in Cartesian or joint space. The user has to specify the movement command type, the goal and the time needed to reach the goal. Movement command types can be linear, circular or spline. Then the main goal of the command interpreter process is to make an interpretation of the robot script message and to insert it into a command queue.

The TrjGen process reads a value from the above mentioned command queue then, using several interpolation routines (i.e. linear, circular, spline), it generates a set of points with a given sample time (i.e. 1 ms). If the robot script includes a Cartesian movement the TrjGen process calls the kinematics functionalities implemented in the kinematics library. Such a library is composed by the following main procedures: position transformations by means of direct and inverse kinematics algorithms, velocity transformations by means of Jacobian functions and workspace check by means of Is-Reachable function. Then, the planned points are filled into a queue that is named reference queue.

The pControl is the process with the highest priority in the system. It is the heart of the robot controller where axis control algorithms, alarm and synchronization procedures have been implemented. Firstly, it extracts one value from the reference queue at each cycle; secondly it executes the control algorithms for the interaction and the jog control detailed below; finally, it communicates with the hardware by means of the MC Server Process which encapsulates all hardware dependent functionalities for reading and writing I/O. A shared memory is used to synchronize the pControl Process with the TrjGen Process and to allow the communication with the HMI Server process. Furthermore, the pControl process has to manage the five possible states of the machine: initialization when parameters are loaded; the start when the robot is powered on; the control loop when control algorithms are executed; the stop when the robot is powered off; the alarm when an error occurs. Then, for monitoring purposes all information related to the machine status (e.g. position, velocity, motor current and so on), is written into a data structure stored in a shared memory by the pControl process.

The HMI Server process, polling such a shared memory with a fixed rate of 10HZ, sends the machine status through the Ethernet LAN using TCP/IP protocol to all client that have been connected to the HMI Server. The HMI Clients are also allowed to send high-level commands (i.e. above mentioned robot scripts) to the controller using Telnet protocol.

The F/T sensor manager is the process that reads the F/T sensor measurement. The F/T sensor data is composed of seven elements, six values are relative to the forces and torques the last one is relative to the temperature compensation. The data are read by the I/O board connected to the F/T sensor. This process includes the routine to communicate with the I/O board and mathematics function to transform the raw data in physical measures. A message consisting of the six values of forces and torques is transmitted via shared memory to the pControl Process.

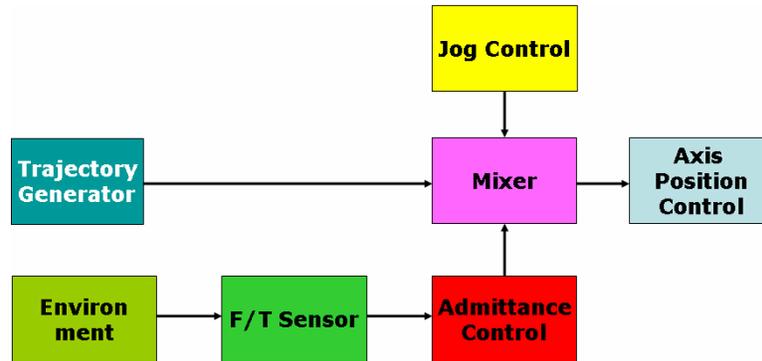


Figure 6 Compliance robot Control

The Figure 6 shows in details the control strategy implemented in the pControl process that makes possible the compliance control of the robot. The core of the system is the axis position control, a standard PID control that has the goal to minimize the gap between the encoder joint position and the reference joint position. Above it, a modular control strategy has been developed.

Before the axis position control, a manager of the compliance strategies, named the mixer in the scheme in Figure 6, is adopted in order to select and mix the control algorithms. Specifically, three type of strategies can be mixed by such a module in the pControl process: the robot path planned by the trajectory generator, a jog position strategy sensible to the joy-pad interaction and a force/compliance strategy based on the force measured from the F/T sensor mounted on the robot tip.

As input to the "mixer" there is an admittance control strategy that acts a balance between force and position control. Using the admittance control it is possible to define a specific behaviour (stiffness, damping) in Cartesian space when there is an interaction with the environment. Practically, this strategy is based on an ideal mass-spring system with damping.

Let  $F$  be the force of the contact,  $K_D$  the damping due to viscous friction effect,  $K_P$  the stiffness of a spring connected to the ground,  $M$  the mass. Then the following equation describes the behaviour of the above mentioned mechanical system.

$$\ddot{X}_{Admit\ tan\ ce} = \frac{F}{M} - \frac{K_D}{M} \cdot \dot{X}_{Admit\ tan\ ce} - \frac{K_P}{M} \cdot X_{Admit\ tan\ ce}$$

The admittance control strategy, implemented on the PA-10 robot controller, reproduces such an equation in Cartesian space regarding all 6 d.o.f.; furthermore, the behaviour of the robot can be tuned with a careful selection of three parameters ( $K_D$ ,  $K_P$ ,  $M$ ) for each d.o.f in the 6D Cartesian space.

Another input to the mixer module is provided by the jog control whose goal is the definition of reference positions for point-to-point motions programmed by the operator by means of a joy-pad. Specifically, a velocity control strategies has been implemented, using as input the position of the joy-pad levers. Furthermore, a low-pass filter has been adopted to have a smooth response.

## 5. Experimental test-beds for manual guidance

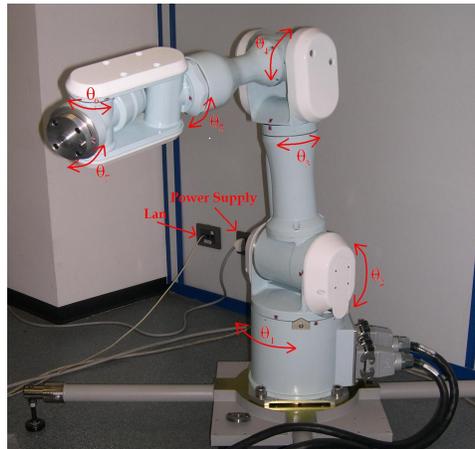
The control system described in section 3 and 4 has been used as test-bed of compliance control strategies addressed in section 2. Such algorithms are needed to have a natural robot programming by means of manual guidance as underlined in section 1. It is important to note that the adopted compliance strategy is constructed over an inner standard joint position controller. Then, as far as an industrial robot controller provides an openness at that level, the proposed control architecture can be easily integrated.

### 5.1. Manual guidance based on F/T sensor feedback

Mitsubishi PA-10 is a commercial robot with an open architecture that makes the manipulator suitable for research and development. The robot is composed by seven links that allow seven d.o.f. (three rotation and four pivot axes); its structure is shown in Figure 7. The geometrical characteristics of the PA-10 manipulator are the follows: the length of the manipulator is 1.37 m and the weight is about 35 kg. The robot joints are actuated through three-phase AC servomotors, while the power is given through the domestic 230V AC.

The setup that is normally sold consists of a robot arm, a servo controller, a motion control card and an upper control computer linked to the servo controller by means of a dedicated optic fibre network based on ARCNET. Several levels of control are allowed by the standard Mitsubishi axis control card, from an higher level controlling the movement in Cartesian space by using an inner interpolator up to a lower level controlling directly the position of the joints. Using the high-level Mitsubishi libraries provided in the upper controller, the maximum control frequency is 100Hz. In the control mode used in our application the Mitsubishi upper controller has been replaced by ours, bypassing the high level control libraries. In this mode it is possible to obtain a control of the robot with a frequency of 1kHz. The control of the servo may be carried out in torque mode or in velocity mode. At the moment, the second modality has been chosen, sending a velocity signal for each joint to the servo. This means that the position loops are closed within the upper controller. The servo has a digital PI feedback that allows the joint-torque control of the robot. The upper controller runs the Linux RTAI 3.2 operating system on a PC with an 2.4 Gbyte Intel Pentium 4 processor.

A CControl PCX20020 ARCNET card is mounted on the upper control computer to allow the communication (maximum bitrate of 10 Mb/s) with the servo controller. Regarding the upper controller software, the architecture described in section 3 has been applied. Firstly, some customizations have been done within the application layer. Specifically, they involve the kinematics, the interpreter and the trjGen in order to deal with a 7 d.o.f. serial robot. Then, at system level a specific RTAI kernel module has been used to provide the implementation of POSIX message queues. Finally, a custom implementation of Task and SharedMemory modules have been developed to encapsulate the same Linux RTAI functionalities. The communication with the hardware has been achieved by means of customization of ARCNET driver provided by standard Linux kernel 2.6.x; all such functionalities have been encapsulated in the MC server Process.



*Figure 7 Mitsubishi PA-10 Robot*

Furthermore, exploiting the openness of the adopted open source RTOS and the modular architecture of the control system, an external device has been added in order to be used as a robot teach pendant.

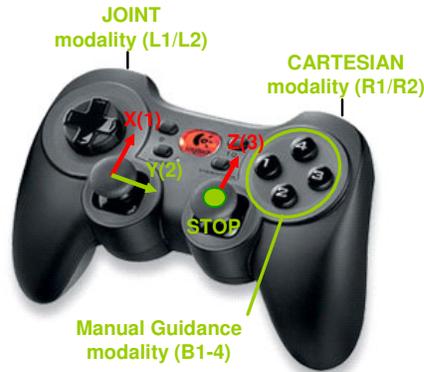


Figure 8 Logitech Cordless RumblePAD 2

Such a device is a USB wireless joy pad widely available on the consumer PC market. Its use has two main benefits: it is a low-cost devices (about 50€) and it is intuitive since it is well-known in popular game stations. Then, the teaching of the operator is very intuitive and need only few minutes. Specifically, the device has 2 analogical levers, 12 digital buttons and allows a dual rumble vibration.

When the programming by demonstration (PbD) of the robot is enabled, the operator could choose among three different way to command the robot: joint jog-teaching (L1, L2 buttons), Cartesian jog-teaching (R1, R2 buttons) or in manual guidance (B1, B2, B3 buttons).

Pressing the L1 button the operator selects the joint modality for the first 3 joints (base joints) and pressing the L2 button the operator selects to move the last 3 joints (wrist joints); in this case the left lever controls the first and the second axes; instead the right lever controls the third axis.

The Cartesian jog-teaching is enabled by pressing R1 button (robot translations) or R2 button (robot rotations); in this case the right lever controls the Z axis, the left lever controls the X and the Y axis.

B1 button allows the operator to select the manual guidance modality in all 6 d.o.f.; differently, manual guidance can be activated either with reference to translations (B2 button) or rotations (B3 button). The manual guidance modality is implemented using the admittance control strategy described in section 2, imposing the damper of the tip equal to zero. A relative value of the stiffness of the robot can be set in the Cartesian space.

The user moves manually the tip of the robot and during the motion if the forth bottom is pressed the end effector position is recorded. The results point-point positions are used to generate a robot program to move the robot arm through the same motion. An analogical six axis F/T sensor has been connected to the system using a acquisition board. The F/T sensor is made from ATI and distributed in Europe form Shunk. The device model is named FT-Mini 45 and the rate of force allowable for measuring are +/- 580 N and +/- 10 Nm. The F/T sensor is placed in the manipulator wrist and has an outside diameter of 45 mm.



Figure 9 Shunk - FT Sensors

Figure 10 shows the behaviour of the PA-10 robot controlled with the compliance control strategies in the Cartesian Space. On the first row the interaction forces in Newton exerted by the human on the robot tip are drawn, while on the second row the resulting target positions in meters are shown. Specifically the control algorithm has been set up in order to have three

different compliance behaviours on the three Cartesian axes. In details, the X-axis has been set to have an under damped response; in particular this behaviour is shown between time of 20 s and 25 s where several oscillations are generated after the application of the force in X direction. Between 10s and 20s a force in Y direction is applied and on the related target position plot an over damped response is shown. On Z direction a behaviour without spring has been imposed. Such behaviour is used generally for the manual guidance. The target position reported in the left-under corner plot shows the points generated during an interaction in Z direction. The admittance is set to have a purely damped reaction, so the obtained position is an integration of the force.

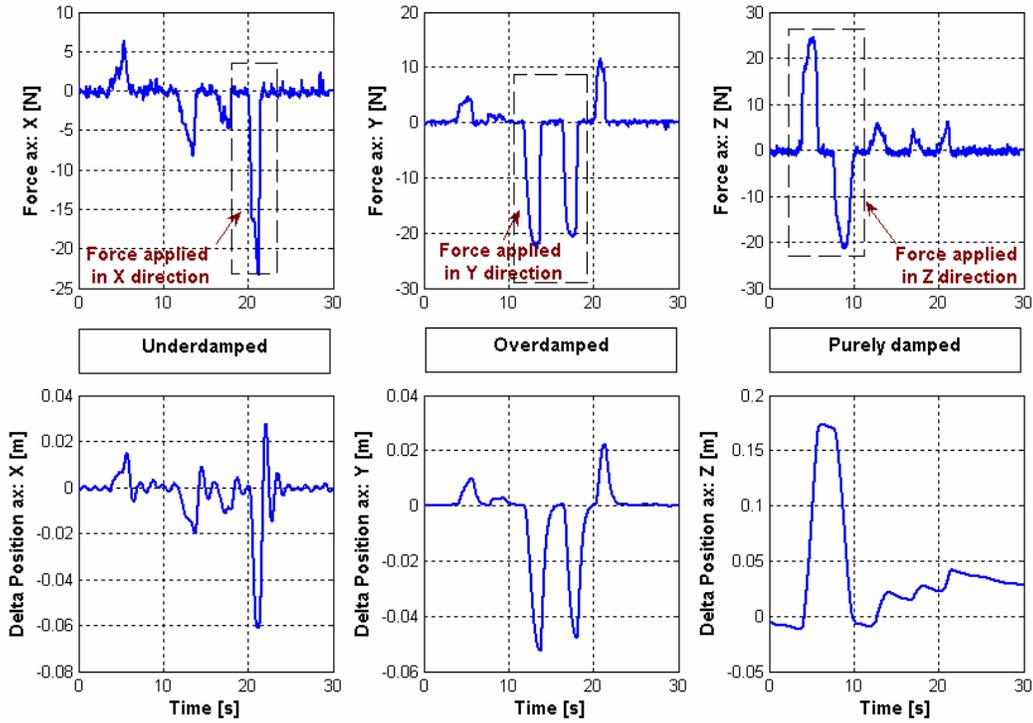


Figure 10 Different compliance behaviour of the PA-10 robot

## 5.2. Manual guidance with motor currents feedback

F/T sensors are quite expensive; their cost rates from a minimum of 4 K€ to more than 15 K€ and this is the main drawback of their use in industrial applications. Although the continuous improvements of electronic devices will give a reduction of the cost of these sensors in the next future, the use of sensor-less methods such as ones based on motor-current feedbacks could be a solution to develop cost-effective intelligent compliance strategies for robotic applications in SMEs.

The observation of motor current to estimate the force interaction was studied in the past [16] but the high computational resources needed by the algorithms were not suitable for the old robot controllers. Nowadays the situation is quite different due to new computer processors installed on the robot controllers that allow the integration of innovation and sophisticated control strategies. In details, the strategy that has been analysed is based on the observation of the change of the motor currents due to the interaction between the robot and the environment. The method does not use any F/T sensors but only the measure of the joint position, given from the robot encoder, the acceleration and the velocities, calculated with a numerical derivation and the joint torque, estimated with the measure of the motor current. Furthermore, a detailed dynamic model of the robot is used. It provides information on how a specific dynamic phenomena, such as inertial, Coriolis/centrifugal, gravitational and friction effects, influences the robot behaviour in a given motion task. The used dynamic model of the robot is described using the formalism shown in the following equation

$\hat{\tau} = \hat{B}(q) \cdot \ddot{q} + \hat{C}(q, \dot{q}) \cdot \dot{X}_p + \hat{G}(q) + \hat{F}_r(\dot{q})$  where  $\hat{B}(q)$  is the inertia matrix,  $\hat{C}(q, \dot{q})$  the Coriolis and centripetal matrix,  $\hat{G}(q)$  are the gravity forces,  $\hat{F}_r(\dot{q})$  are the friction forces and  $\hat{\tau}$  are the estimated joint torque.



Figure 11 Manual Guidance of a COMAU NS-16 robot

The parameters of the dynamic model are obtained, partially, from CAD simulation and, partially, from experimental identification. The approach makes use of model identification, with the estimation of a set of dynamic parameters starting from the response measure, achieved using a robot dynamic excitation experiment. The experimental identification is the only feasible way to obtain both accurate robot models and indications of their confidence accuracy and validity. The response of the robot when an interaction with the environment occurs can be summarized with the following equation:

$$\tau = B(q) \cdot \ddot{q} + C(q, \dot{q}) \cdot \dot{q} + G(q) + F(q) + J^T H$$

where H is the generalized Cartesian forces and torque, while J is the Jacobian matrix. The observer has the goal to minimize the uncertainty of the estimated parameters and the external disturbance, in this way in first approximation the values of the external forces/torques are obtained by the comparison of the real and the estimated joint torque using the follow relation.

$$\hat{h} = J(q)^{-T} (\tau - \hat{\tau})$$

The same control architecture drawn in Figure 6 has been integrated into the COMAU robot controller as shown in Figure 12. The COMAU teach pendant has been used instead of the wireless joy-pad, and the F/T sensor has been substituted by the force estimator.

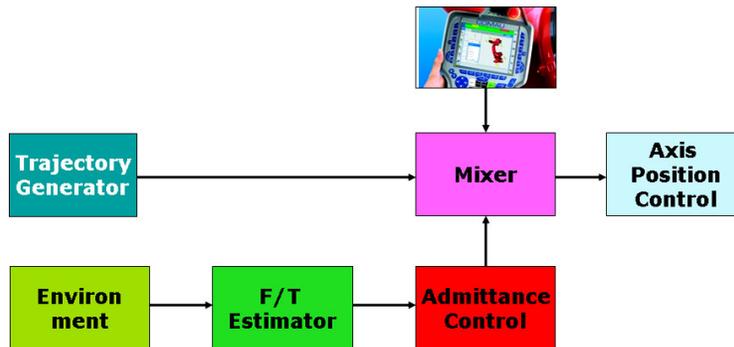


Figure 12 Control Architecture used for manual guidance

## 6. Conclusion and future work

In this paper, a reference architecture for the development of PC and Open Source based robot controllers has been illustrated. Our major goals includes the definition of a modular architecture which can be easily reconfigured with reference to the kinematics of the controlled robot and which can be easily applied to different RTOSs. Specifically, this

architecture has been implemented with reference to a 7-d.o.f. robot arm manufactured by Mitsubishi. With reference to such a robot, the reconfigurability and modularity of the control system have been exploited in order to implement Programming by Demonstration based on manual guidance and in order to use a low-cost programming device. The proposed architecture allows both the manual guidance based on the feedback provided by an F/T sensor (as shown in the PA-10 application) and the feedback of motor currents (as shown in COMAU NS-16 application).

Future works include the extension of the proposed control architecture with the objective to have a robust and interactive robot control. Such an activity is under development with the cooperation of the research group of the mechanical department of University of Brescia headed by Prof. Giovanni Legnani. This architecture consists of a core of compliance algorithms focusing on task frame definition (Figure 13). The target is to have an intelligent controller, safe and robust, able to understand complex tasks and to share them in simplex commands.

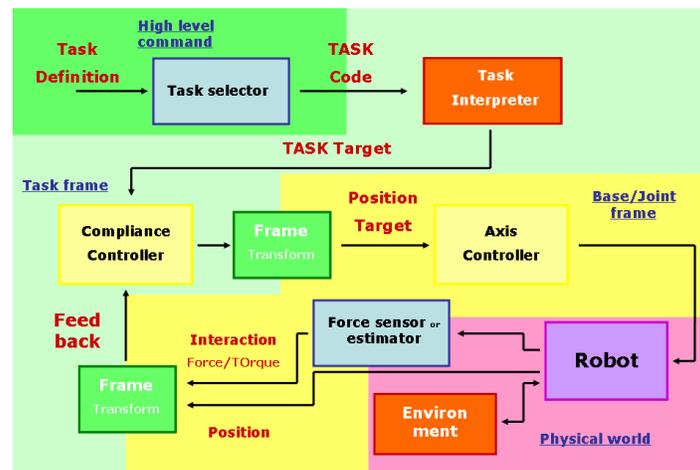


Figure 13 Scheme of the new proposed control strategies

## 7. Acknowledgments

This work has been partially funded by the European Commission's Sixth Framework Programme under grant no. 011838 as part of the Integrated Project SMERobot [15]. The authors would like to thank Comau Robotica, and in particular Luca Lachello, Gerio GianPaolo, Maela Bertone, Valerio Perna, Michela Macciocu, for their support regarding the experimental test-bed of sensor-less manual guidance.

## 8. References

- [1] G. Pritschow, Y. Altintas, F. Jovane, Y. Ko-ren, M. Mitsuishi, S. Takata, H. Van Brussel, M. Weck, K. Yamazaki, 2001, Open Controller Architecture - Past, Present and Future, Annals of the Cirp, vol. 50/2/2001.
- [2] The Orocos Project. <http://www.orocos.org>, 2003
- [3] D. Dallefrate, E. Carpanzano, L. Molinari Tosatti, F. Jovane, Feed rate optimization technique for high-speed CNC machining with parallel manipulators, The 3rd Chemnitz parallel Kinematics Seminar and 2002 Parallel Kinematic Machines International Conference; April 23-25, 2002, Chemnitz, Germany.
- [4] G. Pritschow, G. Rogers, G. Bauer, M. Kremer Open Controller Enabled by an Advanced Real-Time Network. CIRP 2nd International Conference on Reconfigurable Manufacturing, Ann Arbor, MI, USA, August 20-21 2003.
- [5] L. Dozio, P. Mantegazza, Linux Real Time Application Interface (RTAI) in low cost high performance motion control, Motion Control 2003, a conference of ANIPLA, Associazione Nazionale Italiana per l'Automazione (National Italian Association for Automation), Milano, Italy, March 27-28 2003.
- [6] M. Malosio, M. Finardi, S. Negri, L. Molinari Tosatti, F. Jatta, A Modular Architecture for High-Level Robot Programming and Control in a PC-Based Environment IEEE International Conference on Robotics and Automation (ICRA), New Orleans, 2004.
- [7] D. Dallefrate, D. Colombo, L. Molinari Tosatti, Development of robot controllers based on PC hardware and open source software, Seventh Real-Time Linux Workshop - University for Science and Technology of Lille (USTL) - Lille, France, 2005
- [8] Biggs, G; MacDonald, B.: *A Survey of Robot Programming Systems*, Proceedings of the Australasian Conference on Robotics and Automation, Brisbane, Australia, 2003.

- [9] Ciblak N. and H. Lipkin: New Properties of the Remote Center of Compliance, Proceedings of the 1997 IEEE International Conference on Robotics and Automation
- [10] Nathan Delson, Harry West, Robot Programming by Human Demonstration: Adaptation and Inconsistency In Constrained Motion", Proceedings of the 1996 IEEE International Conference on Robotics and Automation Minneapolis, Minnesota - April 1996
- [11] Sangcheol Lee, "Development of a New Variable Remote Center Compliance (VRCC) With Modified Elastomer Shear Pad (ESP) for Robot Assembly", IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, VOL. 2, NO. 2, APRIL 2005
- [12] D. Colombo, F. Jatta, L. Molinari Tosatti, *Reconfigurable control strategies applied to a reconfigurable PKM*, 3rd CIRP Conference on Reconfigurable Manufacturing, Ann Arbor, USA, May 10-12, 2005
- [13] H. Bruyninckx and J. De Schutter, "Specification of force controlled actions in the task frame formalism – a synthesis," IEEE Trans. on Robotics and Automation, vol. 12, no. 4, pp. 581-589
- [14] Sciavicco L., Siciliano B. "A Survey on Robot Control Technology", Proc. of the 4th Jordanian IEEE Conference, Amman, Jordan, pp. 395-400
- [15] [www.smerobot.org](http://www.smerobot.org)
- [16] K. S. Eom, I.H. Suh, W. K. Chung, S-R. Oh - "Disturbance Observer Based Force Control of Robot Manipulator without Force Sensor" - proc. of the IEEE International conference on Robotics & Automation – Leuven, Belgium – May 1998

## 9. Curriculum vitae

Davide Colombo took the Master degree in electrical engineering, major in industrial automation in 2004 from Politecnico di Milano . Since 2004 he is a researcher in the Institute of Industrial Technologies and Automation (ITIA) of the National Research Council (CNR). His research interests include PC based control, compliance control, modelling and estimation of mechanical systems.

Dario Dallefrate received the Master degree in Mechanical Engineering in 2000 from Politecnico di Milano, Italy. Since 2001 he works in the Institute of Industrial Technologies and Automation (ITIA) of the National Research Council (CNR) His main research interests are: PC based control, machine tools and robots path planning, parallel kinematics machines and simulation.

Lorenzo Molinari Tosatti received the 'Laurea' degree in Electrical Engineering and the Doctorate degree in Applied Mechanics and Robotics in 1992 and 1996, respectively, both from Politecnico di Milano, Italy. Since 1994 he taught several courses in mechanics and robotics at Politecnico di Milano and trained several Master's students. In November 1996 he joined the Institute of Industrial Technologies and Automation (ITIA) of the Italian National Research Council (CNR) working on the analysis of modular, distributed and reconfigurable controllers for flexible automation. Since November 1996 he is Italian Eureka-Factory Resource Team Leader. In 1996 he has been visiting Researcher at the Engineering Research Centre for Reconfigurable Machining Systems, University of Michigan. Since 1998 he is Director of the Research Group on Industrial Robotics and Automation Systems (IRAS) at ITIA-CNR.