A Position-based Approach for Force/ Moment Control of an Industrial Manipulator

Manuel Esteves de Mendonça
manuel.mendonca@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

December 2019

Abstract

Force control strategies can provide an effective framework to deal with tasks that involve robotic-environment interaction. In this thesis, a position-based approach to force and moment control robotic manipulators is proposed while considering non-minimal 6-DOF interactions. Such approach allows a complete use of the available sensor measurements by operating the control action in a full-dimensional space without resorting to selection matrices. The force and moment control actions are designed to prevail the motion control loop, therefore ensuring limited deviations from the prescribed force trajectory. Position/ orientation and force/ moment must be specified along each direction of the task frame. A strategy to overcome the hurdle related to the non-contact to contact transition with the environment is considered, assuming a simplified compliant model of the environment and a PI controller law for both controllers' action. It relies on a hardware-software architecture for which the manipulator is remotely controlled while having no access to the lower layers of software running on the industrial controller. This architecture contains a ABB® IRB140 manipulator endowed by an IRC5 industrial controller and a JR3® 6-DOF force-moment sensor. The communication between the IRC5 and the external computer is achieved by a remote control application between Simulink Real-Time™ and RAPID through the RS-232 protocol with a sampling rate of $33Hz$. To validate and prove the effectiveness of the postulated approach, several experiments of representative applications were performed utilizing an analytical trajectory planner.

Keywords: Force Control, Standard Industrial Controller, Position-based Force/ Moment Control, Trajectory Planning

1. Introduction

Industrial robots are designed to meet the requirements for the widest set of potential applications, which is difficult to achieve. Classes regarding payload capacity, number of robot axes and workspace volume have emerged for application categories such as assembly, palletizing, painting, welding, machining and general handing tasks. Versatility enables robots to work in both rigid and flexible automation [1]. Robot-based working cells are more flexible and allow the production of different products at the same time, since they can be easily adapted.

Focusing in machining applications, robots could acquire all functionalities of CNC-machines and represent a reasonable alternative. Displaying good accuracy, they provide larger workspace and more flexibility. The same robot can realize diverse manufacturing processes, making it universal, while CNC-machines can only execute one or a group of similar operations.

Robots in machining applications often use force and torque sensors that allow online estimation of the deflections in the tool locations with respect to the desired ones. The force and torque sensors are usually integrated into a robot's wrist, and the robot controller is able to do an appropriate modification of the prescribed robot motion upon the information provided by measuring the interaction force. They need to be more than position controlled, that will only be sufficient to perform tracking tasks where there is no interaction with the surrounding environment. Force control is up the essence to take in consideration the physical interaction between the robot tool and the working objects or surfaces, since forces and moments will arise from contact. One of the most commonly used approaches in force control is Hybrid force/ motion control, which allows a robot manipulator to follow a position trajectory and simultaneously adjust the forces applied to the environment based on measurements from sensors, handling them as two separate sub problems.

2. State of the Art

During the interaction, the environment imposes constraints on the end-effector motion, denoted kinematic constraints. The contact with a stiff surface is generally referred to as *constrained motion* [1]. Motion can be constrained by the environment along both the translational and the rotational degrees-of-freedom which corresponds to a six-DOF interaction task [2]. Khatib [3] was the first to address the control of end-effector motion and contact forces with a general six-DOF controller, considering that both forces and moments may arise during the task execution when the end-effector tends to violate the constraints.

A suitable description of the end-effector orientation should be adopted to describe and perform a six-DOF interaction task. The usual minimal representation of orientation is given by a set of three Euler angles. According to [2], this formulation fails in the occurrence of representation singularities and can lead to an inconsistency between the moment applied during the task execution and the corresponding displacement in terms of Euler angles, they are not always aligned to the same axis. The latter is due to the fact that a set of three Euler angles does not represent a vector in the Cartesian space. To overcome the drawbacks of the previous formulation, the author concluded that the unit quaternion, a non-minimal representation characterised by four parameters, is the most suitable way to represent the end-effector orientation. It has a physical meaning and mitigates the effects of representation singularities.

The two main approaches to control the interaction of the manipulator and its environment are Hybrid force/ position control and Impedance Control. Hybrid control decomposes the task space into force and position controlled directions [4]. On the other hand, Impedance control does not regulate motion or force directly, but instead regulates the ratio of force to motion [5].

Hybrid control enables the tracking of position and force references but requires an accurate model of the environment. The controller structure depends directly on the geometrical or analytical environment model. In addition, the presence of modelling errors leads to unwanted movements along the force controlled directions and unwanted forces along position controlled directions. Hybrid control designs neglect the impedance behaviour of the robot in response to these imperfections and Impedance control only provides a partial answer, since contact forces cannot be directly imposed and may grow in an uncontrolled manner due to modelling errors of the environment impedance [6]. Trying to cope with uncertainties in the environment geometry and regarding the task space decoupling of Hybrid control but in a full-dimensional space, Chiaverini et al. [7] presented the Parallel force/ position control. The control action is obtained as the sum of the two parallel control actions, force and position control.

2.1. Impedance and Admittance Control

In admittance control or Position-based impedance control, the motion control is separated from the Impedance control and each problem is taken separately. Resorting to an external FTS, the author in [8] developed a controller of this kind, requiring joint position feedback as well as force sensing. The design is ideal for applications using commercial manipulators featuring a motion controller, which he used on an ABB® IRB 140 with an ABB® IRC5 controller. Thus, the position and orientation control is achieved by the industrial manipulator controller, guaranteeing a rejection of the disturbances, and the gains of the impedance control laws (1) and (2) can be set to ensure a satisfactory behaviour during the interaction with the environment. The impedance is given by the two second-order dynamic equations:

$$\boldsymbol{F}_e = \boldsymbol{M}_d \Delta\ddot{\boldsymbol{x}} + \boldsymbol{B}_d \Delta\dot{\boldsymbol{x}} + \boldsymbol{K}_d \Delta\boldsymbol{x} \tag{1}$$

$$\boldsymbol{\mu}_e^d = \mathbf{M}_o \Delta\dot{\boldsymbol{\omega}}_{cd}^d + \boldsymbol{B}_o \Delta\boldsymbol{\omega}_{cd}^d + \boldsymbol{K}_o' \boldsymbol{\epsilon}_{cd}^d \tag{2}$$

with

$$\boldsymbol{K}_o' = 2\mathbf{E}^T(\eta_{cd}, \boldsymbol{\epsilon}_{cd})\boldsymbol{K}_o, \tag{3}$$

$$\boldsymbol{E}(\eta, \boldsymbol{\epsilon}) = \eta\boldsymbol{I} - \boldsymbol{S}(\boldsymbol{\epsilon}) \tag{4}$$

In the above equations, $\boldsymbol{F_e}$ and $\boldsymbol{\mu}_e^d$ represent, respectively, the 3×1 vector of generalized contact forces and moments applied at the end-effector, $\boldsymbol{M}$, $\boldsymbol{B}$ and $\boldsymbol{K}$ are the 3×3 diagonal impedance parameters for both the translational (subscript d) and rotational parts (subscript o), the 3×1 vector $\Delta\boldsymbol{x}$ is the positional error between the compliant and desired frames ($\Delta\boldsymbol{x} = \boldsymbol{x}_c - \boldsymbol{x}_d$) and $\boldsymbol{\epsilon}_{cd}^d$ is the vector part of the unit quaternion describing the orientation displacement (or the mutual orientation) between the compliant and desired frames with respect to the desired frame. Angular velocities $\boldsymbol{\omega}_{cd}^d$ are computed by integration of the quaternion propagation equations and the variation $\Delta\boldsymbol{\omega}_{cd}^d$ is the error between the angular velocities of the compliant and desired frames relative to the desired frame ($\Delta\boldsymbol{\omega}_{cd}^d = \boldsymbol{\omega}_c^d - \boldsymbol{\omega}_d^d$). The matrix $\boldsymbol{S}(\cdot)$ is the so-called skew-symmetric operator.

2.2. Hybrid Position/ Force Control

In a commercial robotic system it is suitable to implement implicit or position-based force control by closing a force-sensing loop around the position controller. The practical reason why the methods based on explicit force control can not be suitably applied in commercial robotic system lies in the fact that commercial robots are designed as "positioning devices".

De Schutter et al. [9] presented a method for compliant motion control based on the theory of Mason [10] in hybrid force/position. The author implemented it with a force-control loop around the position control and since the position controller provides a basis for realization of force control, this concept was referred as external force control or position-based (implicit) force control. The output is an equivalent position in force-controlled directions that is used as the reference input to the position controller. Force control block in this scheme has a twofold role: firstly, to compensate for the effects of the environment (contact process), and secondly, to track the desired force. Commonly, a PI force controller is applied.

2.3. Parallel Force/ Position Control

To offer some robustness with respect to the uncertainties in the environment, Chiaverini et al. in [7] proposed the Parallel force/ position control. Firstly, the controller was only designed for force-position control then Natale et al. in [11] formulated it for moment and orientation. The controller combines a PD position control loop with a PI force control loop where the control action is obtained as the sum of the two parallel control loops. In the resulting Parallel control the force tracking is dominant to accommodate contact forces (planned and unplanned) in any situation, while the position control loop allows the compliance (deviation from the nominal position) to attain the desired forces at the expense of a position error. The prevalence from force control prevents the undesirable effects described for the case of hybrid control.

In [2] the author formulated a Parallel Control variant for a six-DOF interaction task, where an inner motion loop should be designed and the references to be tracked should be suitably computed by an outer force loop. The subscript r denotes the reference frame to be tracked, and its position $\boldsymbol{p}_r$ is computed through the technique of the parallel composition defined as,

$$\boldsymbol{p}_r = \boldsymbol{p}_c + \boldsymbol{p}_d \tag{5}$$

$$\dot{\boldsymbol{p}}_r = \dot{\boldsymbol{p}}_c + \dot{\boldsymbol{p}}_d \tag{6}$$

$$\ddot{\boldsymbol{p}}_r = \ddot{\boldsymbol{p}}_c + \ddot{\boldsymbol{p}}_d \tag{7}$$

being $\boldsymbol{p}_c$ in (5) the solution to the differential equation expressing the force control law

$$\boldsymbol{K}_{Pf}\ddot{\boldsymbol{p}}_c + \boldsymbol{K}_{If}\dot{\boldsymbol{p}}_c = \Delta\boldsymbol{f} \tag{8}$$

with $\Delta\boldsymbol{f} = \boldsymbol{f}_d - \boldsymbol{f}$. In regard the rotational part, the desired orientation trajectory is specified as a relative orientation between the desired frame and the compliant frame, in the sense that the quaternion $\mathcal{Q}_{dc}$ is a rotation about an axis aligned to an unconstrained direction and defined in the compliant frame. Therefore, the parallel composition for the rotational part is defined as,

$$\mathcal{Q}_r = \mathcal{Q}_c * \mathcal{Q}_{dc} \tag{9}$$

$${}^{c}\boldsymbol{\omega}_r = {}^{c}\boldsymbol{\omega}_c + {}^{c}\boldsymbol{\omega}_{dc} \tag{10}$$

$${}^{c}\dot{\boldsymbol{\omega}}_r = {}^{c}\dot{\boldsymbol{\omega}}_c + {}^{c}\dot{\boldsymbol{\omega}}_{dc} \tag{11}$$

where $\mathcal{Q}_c$, $\boldsymbol{\omega}_c$ and $\dot{\boldsymbol{\omega}}_c$ characterize the rotational motion of the compliant frame. These quantities can be computed since the rotational motion of the compliant frame has been computed according to the differential equation expressing the moment controller

$$\boldsymbol{K}_{Po}{}^{c}\dot{\boldsymbol{\omega}}_c + \boldsymbol{K}_{Io}{}^{c}\boldsymbol{\omega}_c = \Delta^{c}\boldsymbol{\mu} \tag{12}$$

with $\Delta^{c}\boldsymbol{\mu} = {}^{c}\boldsymbol{\mu}_d - {}^{c}\boldsymbol{\mu}$.

3. Methods and Implementation

Within the approach here taken, the IRC5 industrial controller is responsible for the position control and the force-moment control strategy will accommodate the user defined trajectory with the data collected from the sensor. Position, orientation and force/ moment must be specified along each direction of the task frame and the control strategy rests on the combination of the following two modules:

- Trajectory Planning - responsible for the analytical trajectory between initial and target points, it generates a time sequence of values for the end-effector motion and orientation so the manipulator follows a geometrically specified path in space;
- Position-based Force/ Moment Control - responsible for regulation to a desired constant force and moment, and tracking of a time-varying desired pose trajectory considering a non-minimal representation of the operational space (position and quaternion orientation) when there is contact between the robotic system and the environment

3.1. Trajectory Planning

Moving the manipulator is done by calling a `Move` function whose arguments are target destination, travelling speed, accuracy of motion and the type of interpolation. Meaning, the IRC5 controller features algorithms that interpolate paths between two points, and in what concerns path interpolation and latency `MoveAbsJ` is the fastest way to move the manipulator [8]. By leaving the trajectory planning for the IRC5 there would be no control or access to the interpolated points between the starting and final position. If the end-effector motion has to follow a prescribed trajectory of position and/or orientation, this must be expressed analytically. It is then necessary to refer to motion primitives defining the geometric features of the path and time primitives defining the timing law on the path itself.

For a rectilinear path, consider the linear segment connecting point $\boldsymbol{p}_i$ to point $\boldsymbol{p}_f$, the parametric representation of this path is

$$\boldsymbol{p}(s) = \boldsymbol{p}_i + \frac{s}{\|\boldsymbol{p}_f - \boldsymbol{p}_i\|}(\boldsymbol{p}_f - \boldsymbol{p}_i). \tag{13}$$

For a circular path, consider the frame $O'-x'y'z'$, where O' coincides with the centre of the circle, axis x' is oriented along the direction of the vector $\boldsymbol{p}_i - \boldsymbol{c}$, axis z' is the characteristic circle plane vector (user defined) and axis y' is chosen so as to complete a right-handed frame. When expressed in the base reference frame, the parametric representation of the circle as function of the arc length is

$$\boldsymbol{p}(s) = \boldsymbol{c} + \boldsymbol{R}\boldsymbol{p}'(s), \tag{14}$$

where $\boldsymbol{p}'(s)$ is the parametric representation in $O' - x'y'z'$ frame (15), $\boldsymbol{c}$ is centre of the circle expressed in the base reference frame, $\boldsymbol{R}$ is the rotation matrix of frame $O' - x'y'z'$ with respect to base reference frame and ρ is the radius of the circle.

$$\boldsymbol{p}'(s) = \begin{bmatrix} \rho\cos(s/\rho) \\ \rho\sin(s/\rho) \\ 0 \end{bmatrix}, \tag{15}$$

End-effector orientation was achieved resorting to angle and axis description. Let $\boldsymbol{R}_i$ and $\boldsymbol{R}_f$ denote respectively the rotation matrices of the initial frame $O_i - x_i y_i z_i$ and the final frame $O_f - x_f y_f z_f$, both with respect to the base frame. The rotation matrix between the two frames can be computed by $\boldsymbol{R}^i_f = \boldsymbol{R}_i^T \boldsymbol{R}_f$ and can be expressed as the rotation matrix about a fixed axis in space; the unit vector $\boldsymbol{r}^i$ of the axis and the angle of rotation ϑ_f can be computed by using

$$\vartheta_f = \cos^{-1}\left(\frac{r_{11} + r_{22} + r_{33} - 1}{2}\right) \tag{16}$$

$$\boldsymbol{r}^i = \frac{1}{2\sin\vartheta_f}\begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} \tag{17}$$

for $\sin\vartheta_f \neq 0$. The matrix $\boldsymbol{R}^i(t)$ can be interpreted as a matrix $\boldsymbol{R}^i(\vartheta(t), \boldsymbol{r}^i)$ and it is then sufficient to assign a timing law to ϑ with $\vartheta(0) = 0$ and $\vartheta(t_f) = \vartheta_f$.

The timing law for the s coordinate is defined using a third-order polynomial to ensure no discontinuity both in position and velocity.

$$s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{18}$$

$$\dot{s}(t) = a_1 + 2a_2 t + 3a_3 t^2 \tag{19}$$

Equations (18) and (19) describe the cubic polynomial timing law. Since four coefficients are available, it is possible to impose initial and target positions as well as velocity constraints for the same points, usually set to zero.

$$\begin{cases} s(t_i) = 0 \\ s(t_f) = s_f \\ \dot{s}(t_i) = \dot{s}(t_f) = 0 \end{cases} \Leftrightarrow \begin{cases} a_0 = \frac{3t_f t_i^2 - t_i^3}{(t_f - t_i)^3} s_f \\ a_1 = -\frac{6t_f t_i}{(t_f - t_i)^3} s_f \\ a_2 = \frac{3(t_f - t_i)}{(t_f - t_i)^3} s_f \\ a_3 = -\frac{2}{(t_f - t_i)^3} s_f \end{cases} \tag{20}$$

Coefficients a_0, a_1, a_2, a_3 are obtained solving the boundary equations defined in (20). The same logic is applied for each motion primitive stated previously, since the boundary constraints only differ in the value of $s(t_f)$.

3.2. Position-based Force/ Moment Control

The integration of a force-moment control strategy with the closed-ended IRC5 industrial controller represents the major effort of this thesis, since this industrial controller was not designed to have a force control mode. Therefore, a position-based force/ moment controller is implemented. A generic block diagram of the controller is represented in Figure 1, where the force/ moment control generates position and orientation corrections to the planned trajectory then sends them as joint coordinates to the IRC5 position controller resorting to inverse kinematics.

In Figure 1, the subscript d denotes the desired values, the subscript c denotes the deviation resulting from force-moment controller action and r denotes the reference frame to be tracked and its position $\boldsymbol{p}_r$ and orientation $\boldsymbol{Q}_r = \{\eta_r, \boldsymbol{\epsilon}_r\}$ are computed through the parallel composition as it will be detailed on the following controller subsections.

The typical inner motion control on a position-based force control is compacted inside the position control block, since the proposed controller performs inverse kinematics outside that loop, as already mentioned, motion control is performed on the joint space. The force control part of the controller is responsible for accommodating manipulator's motion concerning the first three joint variables (q_1, q_2, q_3), while the moment-orientation control part is responsible for the last three joint variables (q_4, q_5, q_6). Thereby, force and moment-orientation control actions will act, respectively, on

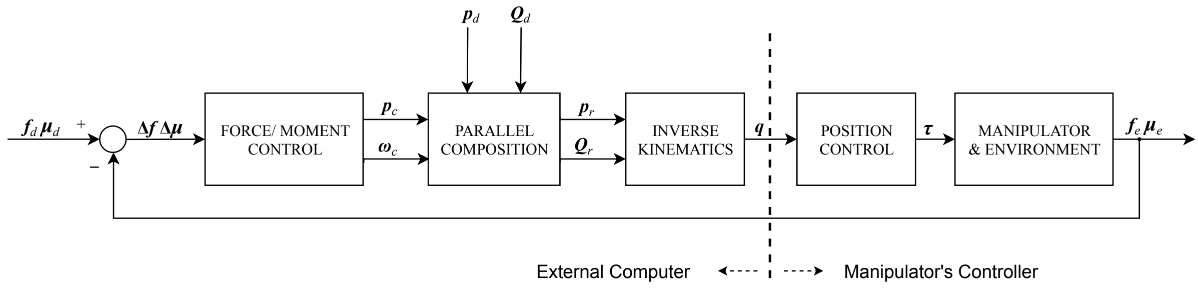


Figure 1: Generic block scheme of position-based force/ moment control

the translational (position) and rotational part of the desired trajectory.

To ensure that the introduction of the control action will not affect free motion movement a finite state-machine is implemented. Force-moment state-machine is responsible for designating how the manipulator is controlled (position or force controlled) depending on force/ moment error values. In a broad sense, it will define when the force-moment controller takes action in addition to plan the trajectory to execute a given task.

Both force and moment control laws will be defined in a classical and simple perspective, as a PI controller in sense of achieving null stationary error (force and torque) and avoiding the differential component (D) due to noisy readings from the force/ moment sensor. The stability of the controller is guaranteed if every pole lies on the left half-plane of $s-$plane and if the controller bandwidth is smaller than the robot motion controller bandwidth, which is a necessary condition for cascade control, the inner loop must be faster than the outer loop. Notwithstanding, open-loop stability of the coupled system is guaranteed given the robotic system is stable. Force and moment control actions can be designed on the basis of a simplified model of the environment (see Figure 2). Assuming $\boldsymbol{k}_s >> \boldsymbol{k}_e$ and $\boldsymbol{c}_s = \boldsymbol{c}_e \simeq 0$, since they can be neglected for cases where small velocities are used. Therefore, contact is modelled by a linear string.

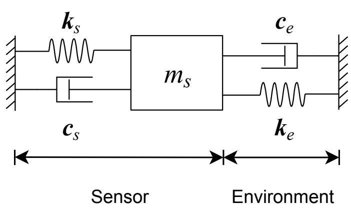


Figure 2: Simplified sensor and environment model

A. Force Control

Explaining the position-based controller through 2 separated controllers (each control action individually), for the force control part the outer loop containing the force controller is the kernel of the control strategy. It has a twofold role: adjusts the manipulator's motion given the force sensed by the sensor and tracks the desired force so it can be imposed to the environment surface. In detail, the force control developed is depicted in Figure 3.

As one can see, the input of the force controller is the difference between the desired and actual contact force ($\Delta \boldsymbol{f} = \boldsymbol{f}_d - \boldsymbol{f}_e$) and a PI controller is considered so there would be no force error. The force control law is given by

$$\boldsymbol{u}_f(t) = \boldsymbol{K}_{P_f} \Delta \boldsymbol{f}(t) + \boldsymbol{K}_{I_f} \int_0^t \Delta \boldsymbol{f}(\tau) d\tau, \qquad (21)$$

where $\boldsymbol{u}_f$ is the displacement due to the error between the desired and sensed force, K_{P_f} and K_{I_f} are, respectively, the proportional and integral gains from the controller.

Position accommodation is defined as the parallel composition

$$\boldsymbol{p}_r = \boldsymbol{p}_d + \boldsymbol{p}_c \qquad (22)$$

where $\boldsymbol{p}_r$ is the position of the reference frame to be tracked, $\boldsymbol{p}_d$ is the desired position and $\boldsymbol{p}_c$ is the displacement due to force contact computed by the force controller with respect to reference base frame. Since the controller acts on the force imposed by the environment on the manipulator, a premultiplication with the rotation matrix from direct kinematics for end-effector actual orientation ($\boldsymbol{R}_e$) transforms the resultant displacement from the sensor frame to the manipulator base reference frame. When there is contact with the environment $\boldsymbol{p}_r$ represents the position of the compliant frame with respect to the base reference frame.

B. Moment-Orientation Control

The strategy presented above for force control can be conceptually pursued also for moment and orientation control, as shown in Figure 3. The outer loop containing the moment controller and the parallel composition is the core of the control strategy, since the output of moment control block is a angular velocity and the trajectory planned will be a quaternion orientation.

The input of the moment controller is the difference between the desired and contact moment ($\Delta\boldsymbol{\mu} = \boldsymbol{\mu}_d - \boldsymbol{\mu}_e$) and a PI controller is considered since it is undesirable to have stationary error. The moment control law is given by

$$\boldsymbol{u}_m(t) = \boldsymbol{K}_{P_m}\Delta\boldsymbol{\mu}(t) + \boldsymbol{K}_{I_m}\int_0^t \Delta\boldsymbol{\mu}(\tau)d\tau, \quad (23)$$

where $\boldsymbol{u}_m$ is the angular velocity due to the error between the desired and sensed moment, K_{P_m} and K_{I_m} are, respectively, the proportional and integral gains from the controller.

Orientation accommodation is defined as the parallel composition

$$\boldsymbol{\omega}_r = \boldsymbol{\omega}_c + \boldsymbol{\omega}_O \quad (24)$$

where $\boldsymbol{\omega}_r$ is the angular velocity of the reference frame to be tracked, $\boldsymbol{\omega}_O$ is the angular velocity necessary to rotate the reference frame r to the desired frame and $\boldsymbol{\omega}_c$ is the angular velocity resulting from the contact moment computed by the moment controller with respect to reference base frame. Since the controller acts on the moment imposed by the environment on the manipulator, a premultiplication with the rotation matrix from direct kinematics for end-effector actual orientation ($\boldsymbol{R}_e$) transforms the resultant angular velocity from the end-effector's frame to the manipulator base reference frame.

Since the inverse kinematics has an orientation as input, a quaternion is computed from the resultant angular velocity due to contact moment ($\mathcal{Q}_c$). However, for the planned orientation trajectory to be considered the orientation difference between reference and desired must be computed. This difference is achieved through a quaternion error ($\boldsymbol{e}_O$) and then expressed as an angular velocity with respect to the base frame (ω_O) for the parallel composition (24). The same line of thought to compute a angular velocity from a quaternion error can be seen on CLIK (closed-loop inverse kinematics) [12].

The equations that rule the orientation control block on the above diagram are given by

$$\boldsymbol{\omega}_O = \boldsymbol{K}_O\boldsymbol{e}_O \quad (25)$$

$$\boldsymbol{e}_O = \eta_r\boldsymbol{\epsilon}_d - \eta_d\boldsymbol{\epsilon}_r - S(\boldsymbol{\epsilon}_d)\boldsymbol{\epsilon}_r, \quad (26)$$

where $\boldsymbol{\omega}_O$ is the angular velocity error between reference and desired frames with respect to the base reference frame, K_O is a suitable positive definite matrix gain and $\boldsymbol{e}_O$ is the quaternion error between reference and desired frames.

It is worth pointing out that the computation of the quaternion from the angular velocity due to the contact moment gives rise to an integral action. The quaternion $\mathcal{Q}_r$ is computed from $\boldsymbol{\omega}_r$ by integrating the quaternion propagation equations with initial conditions $\mathcal{Q}_r(t = 0) = \mathcal{Q}_u$, being $\mathcal{Q}_u$ the quaternion describing the orientation of the undeformed frame with respect to the base frame.

C. Force-Moment State-Machine

The Force-moment state-machine provides the inputs to the controller depending on the force/ moment error. Assuming that they are exerted by manipulator on the environment surface with reference to the end-effector frame. The goals are to avoid the unnecessary activation of the force-moment control action when the manipulator is in free motion, and the force/ moment peak value, which rises when the manipulator contacts with environment.

It was defined three thresholds for the force/ moment error ($\Delta\boldsymbol{h} > 0$, $\Delta\boldsymbol{h} = 0$ and $\Delta\boldsymbol{h} < 0$) that define in which task phase the manipulator is and how it is going to be controlled. Accordingly, the state-machine runs as follows:

1. It starts in the "Approach" state and it remains there while $\Delta\boldsymbol{h} > 0$, meaning there is no contact with the environment. During this state, the manipulator is only position controlled following a given trajectory. When the error decreases below zero ($\Delta\boldsymbol{h} < 0$), the state-machine transits to the "Force Stabilization" state and the manipulator becomes force controlled.

2. While in "Force Stabilization" state, a stabilization on the desired force/ moment value is made by holding end-effector pose right before the contact occurred. Controller action takes place after t_1 seconds to let the sensor readings normalize and consequently avoid misleading position and/ or orientation accommodations. Hence, as long as the error is not null the manipulator will accommodate is pose until the desired force/ moment is not obtained.

3. When $\Delta\boldsymbol{h} = 0$, the state-machine transits to the "Task" state. When in this state, the robotic system is going to wait t_2 seconds then it starts the task itself from the pose where the error was null.

4. Until the manipulator reaches the desired position and/ or orientation the task is not complete. When the conditions are met, the state-machine transits to the "Stop" state and the robotic system stops.

This state-machine structure gives use to the force-moment controller developed and increases system flexibility. Its methodology can be adapted for more complex environments and tasks, since there is no transitions to previous states, and it can be implemented differently given the case as long as sensor signals and transition conditions are consistent.

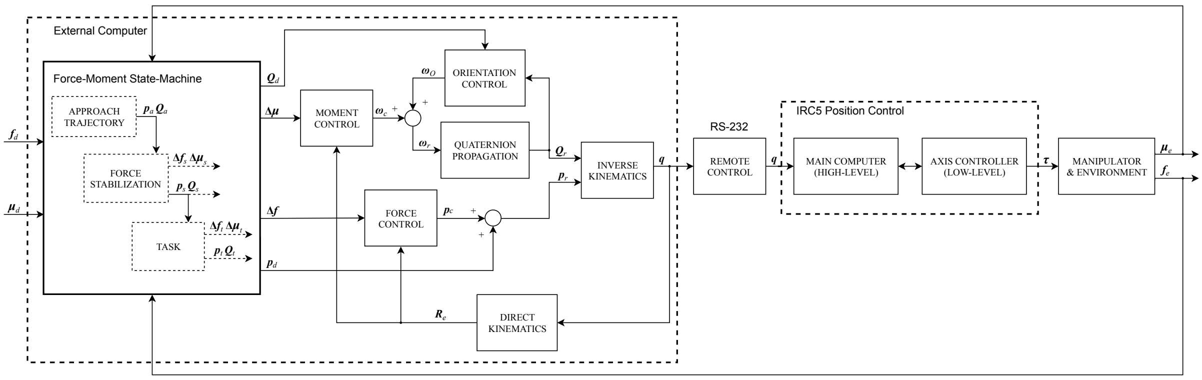


Figure 3: Hardware-software architecture

3.3. Hardware-Software Architecture

The hardware-software architecture used to integrate the force-moment control strategy above presented is made connecting a force/ moment sensor directly to an external computer via a PCIe Ethernet adapter and remote controlling ABB IRB140 throughout an adapted version of the remote control algorithm developed by [8].

While defining the hardware-software architecture, the objective was to implement the maximum number of components necessary to the task realization directly in the external computer. In that way, the motion controller precision present in IRC5 could be exploited and the basis of the remote control application developed by [8] could be kept unaltered. The strategy is split between the two computers as depicted in Figure 3. Note that the communication between the IRC5 controller and external computer shown is performed via RS-232 at a rate of $33Hz$, the best that can be achieved for the IRC5 used.

As one can see, Force-moment state-machine provides the controller inputs in the first three states, even when the manipulator is approaching the material, which at programming level, corresponds to null input values on the force and moment error variables. This way, the controller will act as a block with unitary transfer function and the manipulator will only be position controlled.

By using a trajectory planner on the external computer, the trajectory can be discretized in intermediate points. Smaller sub-paths are created and the analytical trajectory between initial and target points is specified. Therefore, the ability of the manipulator's controller to define the trajectory between the given points is taken. This way, the manipulator's interpolation by the `Move` function will be done on intermediate points in the path. This implementation was not only designed to avoid controller's interpolation in assigning the intermediate points but also to enable the action from the force-moment controller implemented on a smaller distance. If the trajectory was only defined by two points and interpolated by the `Move` function between them, the data received by the FTS could only be used in those two points. Since there is no access to IRC5 main computer trajectory planner, the information of the sensor could not be used to accommodate the manipulator's motion between points. Trajectory discretization will allow the force-moment controller to act in each trajectory point given the data received by the FTS.

Sampling rate has a direct influence on the trajectory discretization, the sub-paths are smaller the higher the sampling rate. As the sampling frequency of the communication between the external computer and the IRC5 is fixed at $33Hz$, the algorithm will generate an intermediate point at each $0.03s$.

The proposed controller performs inverse kinematics in the external computer, different from typical position-based force control, since the RAPID function `MoveAbsJ` has the joint coordinates as input, which was chosen for instilling a better performance to the controller compared to the others functions available.

Since the manipulator is being remotely controlled it will only move when a new joint position is sent by the external computer. Which means that there is no need for the IRC5 to send its pose to the external computer, communication can be unidirectional.

4. Results

The tests were thought trying to resemblance real industrial tasks. In the following the controller is implemented separately in order to evaluate both strategies action, since they can be applied independently.

4.1. Force Control Experiment

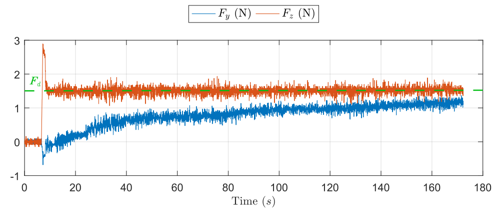
The goal of this experiment is to evaluate the effectiveness of the controller to change the task execution velocity due to excessive process forces in the opposite direction of the movement while imposing pressure against the surface.

This experiment runs with transition conditions on Force-moment state-machine regarding force error along z_e-axis since y direction is both position and force controlled. The manipulator is performing a rectilinear trajectory on the $y-$axis while imposing a pressure of $1.5N$ on the surface (positive direction of z_e-axis). Which means that the robot will change its position in order to apply a constant force on a surface perpendicularly to the movement and adapting the task execution speed given the F_y values, even if the surface is not known. Machining tasks like deburring and polishing with not regular surfaces or cutting processes are encompassed in this test. Forces along the x-axis and moments in all directions are not relevant for this application and will not be considered. Therefore, the force controller gains used are,

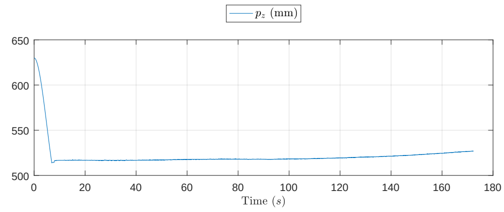
$$\boldsymbol{K}_{P_f} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{K}_{I_f} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 5 \end{bmatrix} \tag{27}$$

The main results of this experience are depicted in Figure 4 and some snapshots taken at meaningfully moments are shown in Figure 5.

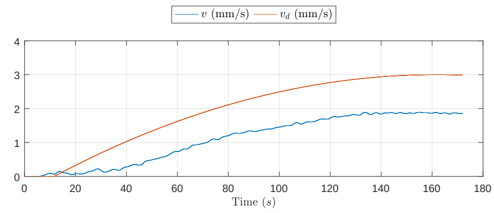
From Figure 4, it is clear that the manipulator accommodates its position along z axis to impose a desire force of $1.5N$ on the surface (in the negative direction of z axis) while task execution velocity is changed due to the values of F_y. There is no contact with the environment within the interval $t \in [0, 6.93]$ while the manipulator executes an approaching downwards trajectory with a parabolic shaped velocity with an average value of $15mm/s$ and is not force controlled. Immediately after the surface is felt (see snapshot of Figure 5(a)), the manipulator stops its movement and holds the pose where the error value decreased below zero to avoid damaging the tool and let the sensor readings normalize. The force peak (occurs at $t = 7.17s$) does not affect the TCP position due to the waiting time on the "Force Stabilization" state in which force values are ignored by the control strategy. The manipulator stays in that state until $t = 10.5s$ stabilizing the position so the desired force value can be imposed. However, due to the system $0.03s$ sampling rate, the robotic system can not achieve a position that nullifies the force error. The condition value was adapted to a interval where the error is considered zero. But only after the error values are consecutively within the interval (with the help of a counter) the condition is considered true.



(a) Force along y_e and z_e axis

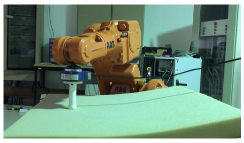


(b) TCP position with respect to base frame

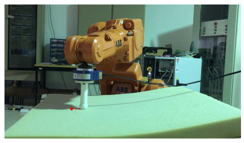


(c) Velocity profile

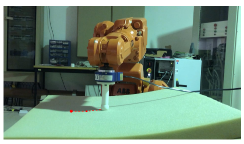
Figure 4: Main results of force control experiment



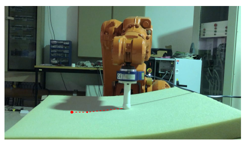
(a) $t \approx 8s$



(b) $t \approx 79s$



(c) $t \approx 130s$



(d) $t \approx 172s$

Figure 5: Snapshots of Pressure and SpeedChange experiment

At $t = 11.4s$ the manipulator starts the task itself and performs a rectilinear trajectory on the xy plane with a velocity profile defined by v_d in Figure 4(c) from the position where force error was considered null. The profile in blue represents the task execution velocity, that was estimated using a $\alpha - \beta$ filter as in the work [13] and then passed through a low pass filter. From the same plot it can be observed the difference between desired and task execution velocity profiles, since y is both position and force controlled. The speed reduction demonstrates that the controller works for directions both position and force controlled. However, while the force increases the velocity decreases(see Figure 4(a)). That is due to the way the trajectory planner is computing the next trajectory point which does not match with the purpose of speed changing in a robotic task. It is intend to change the task velocity so it can decrease the force to a desired value.

It is worth to mentioned that the material location and surface contour are not known. Despite the manipulator starts task execution on the position held from the "Force Stabilization" state (where force error was considered zero) it follows the surface contour. That is, force regulation to a desired value prevails over position control.

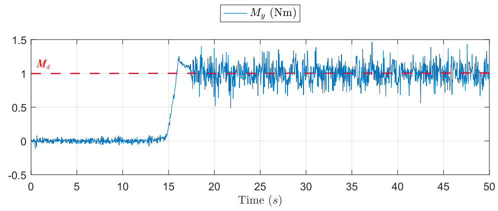
4.2. Moment Control Experiment

This experiment will only portray the first two states of the Force-moment state-machine, since assembly task are more complex than machining tasks. Generally have algorithms running above the control action to decide how to react upon the a given torque and that is not the scope of the present work. Therefore, it tests how a moment influence the end-effector orientation and runs with transition conditions on FSM regarding moment error along y_e-axis. The moment and orientation controller gains used are,

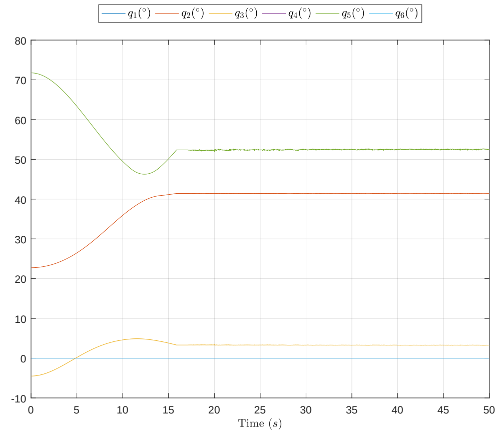
$$\boldsymbol{K}_{P_m} = \begin{bmatrix} 0.3 & 0 & 0 \\ 0 & 0.3 & 0 \\ 0 & 0 & 0.3 \end{bmatrix}, \boldsymbol{K}_{I_m} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \quad \boldsymbol{K}_O = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix} \tag{28}$$

The main results of this experience are depicted in Figure 6.

Figure 6(a) depicts the manipulator maintaining the desired imposed moment it also shows bigger fluctuations around that value comparing to the experiments on Force control. That is due to the low sampling rate used, the robotic system can not react in time. In Force control that effect can be overcome by lowering the execution task speed. However, for the controller gains chosen the effects of those fluctuations are almost unnoticed on joint coordinate q_5 (see Figure 6(b)) which is where the effect would be shown. As for the force control experiments, Force-moment state-machine avoids the moment peak sensed when contacting the object.



(a) Moment along y_e axis



(b) Manipulator joint coordinates

Figure 6: Main results of moment control experiment

5. Conclusions

This work focused on developing a force/ moment control strategy with the aim of giving a industrial manipulator endowed by standard industrial controller the ability to 'feel' its surroundings. However, it was aware from the very beginning of the limitations in the hardware-software architecture regarding the communication bandwidth of $33Hz$ and that it would affect the conditions to perform a task with such a controller. Velocities, controller gains and consequently response time needed to consider that drawback. This also affected the characteristics of the environment for which the controller was tested. It was used a compliant material.

The control strategy was experimentally validated using the hardware-software architecture proposed, which differs from those found in other academic works essentially because it has a moment control action implemented without any access to the low-level axis controller of the IRC5. Most of moment control strategies proposed in the literature are implemented using formulations that can not be used in close-ended industrial manipulators.

This controller can be designed considering a simplified model of the environment for both position and orientation while providing some sort of robustness to uncertainty. Other strategies demand *a priori* knowledge of the contact surface characteristics and the dynamics inherent in order to tune properly the controllers gains.

The results of the experiments show how the control strategy presented can increase the robotic system flexibility by being capable of perform machining and assembly tasks where the force-moment action should prevail motion control. They also enhanced force-moment state-machine importance on the task design. Using the force-moment controller with different goals in its states, overcomes the drawbacks of the controller itself on performing a task as the only component on the control strategy. Besides bypassing the concern with impact control it also enables the force/ moment controller's action only when there is contact with the environment. Allowing the robotic system to perform free motion movements when there is no contact and avoiding force peaks that appear on contact.

Although the controller is designed to operate in a full-dimensional space without using selection matrices it is the user that defines which directions are going to be force and/ or position controlled. It is here presented that for hybrid situations, where force and position controlled directions are orthogonal the architecture developed can be directly applied on a robotic application. However, for a situation where a direction is both force and position controlled the trajectory planner running on the force-moment state-machine, therefore apart from the controller, must be adapted.

5.1. Future Work

Future work should address the controller ability to have a direction both position and force controlled. The next desired pose on the trajectory should be defined accordingly to the force sensed so it can be maintained at desirable values. For example a reactive trajectory planner or a controller acting on the trajectory timing law could be exploited.

With higher sampling rate the system response time would be faster and the robotic system would be capable of performing machining tasks at real velocities. Force-moment state-machine for instance would be capable of achieving the pose where the force error is null or would narrower the interval.

References

[1] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. Springer, 1st edition, 2008.

[2] Ciro Natale. *Interaction Control of Robot Manipulators: Six degrees-of-freedom tasks*, volume 3. Springer-Verlag Berlin Heidelberg, 2003.

[3] O. Khatib. A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

[4] M. Raibert and J. Craig. Hybrid Position/ Force Control of Manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 102:126–133, 1981.

[5] N. Hogan. Impedance Control: An Approach to Manipulation: Part I-III. *Journal of Dynamic Systems, Measurement, and Control*, 107, 1985.

[6] J. De Schutter, H. Bruyninckx, W. H. Zhu, and M. W. Spong. Force control: A bird's eye view. In K. P. Valavanis and B. Siciliano, editors, *Control Problems in Robotics and Automation*, volume 230, pages 1–17. Springer, Berlin, Heidelberg, 1998.

[7] S. Chiaverini and L. Sciavicco. The Parallel Approach to Force/ Position Control of Robotic Manipulators. *IEEE Transactions on Robotics and Automation*, 9(4):361–373, 1993.

[8] V. Sampaio. Towards Interaction Control of an Industrial Robotic Manipulator. Master's thesis, Instituto Superior Técnico, October 2017.

[9] J. De Schutter and H. Van Brussel. Compliant Robot Motion ii. a Control Approach based on External Control Loops. *The International Journal of Robotics Research*, 7(4):18–33, 1988.

[10] Matthew T. Mason. Compliance and Force Control for Computer Controlled Manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(6):418–432, 1981.

[11] C. Natale, B. Siciliano, and L. Villani. Control of Moment and Orientation for a Robot Manipulator in Contact with a Compliant Environment. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation*, volume 2, pages 1755–1760. IEEE, 1998.

[12] S. Chiaverini and B. Siciliano. The Unit Quaternion: A Useful Tool for Inverse Kinematics of Robot Manipulators. *Systems Analysis Modelling Simulation*, 35(1):45–60, 1999.

[13] R. Lucas. Towards Safe Human-Robot Interaction in Industrial Environments. Master's thesis, Instituto Superior Técnico, May 2018.