Chapter 7
Force and Position Fuzzy Control: A Case Study in a Mitsubishi PA10-7CE Robot Arm

Miguel A. Llama, Wismark Z. Castañon and Ramon Garcia-Hernandez

Abstract Too many research works have focused on the problem of control of robot manipulators while executing tasks that do not involve the contact forces of the end-effector with the environment. However, many tasks require an interaction of the manipulator with the objects around it. For the correct performance of these tasks, the use of a force controller is essential. Generally, the control objective during the contact is to regulate the force and torque of the manipulator's end-effector over the environment, while simultaneously regulating the position and orientation (i.e., the pose) free coordinates of the manipulator's end-effector. Many works have been presented on this topic, in which various control strategies are presented; one of the most relevant methods is the so-called hybrid force/position control; this scheme has the advantage of being able to independently control the force in constrained directions by the environment and the pose along unconstrained directions. This work analyzes and implements the hybrid force/position control using a fuzzy logic control method, since the fuzzy control provides a solution for nonlinearities, high coupling, and variations or perturbations. The system employed is the Mitsubishi PA10-7CE robot manipulator, which is a robot of 7 degrees of freedom (DOF), but in this work, it is only used as a 6-DOF manipulator, equipped with a 6-DOF force/torque sensor in the end-effector.

7.1 Introduction

Actually, the ability to handle and manipulate physical contact between a robot and the environment that surrounds it is a demand to perform more advanced manipulation tasks. This capacity is known as the interaction of the manipulator with the physical environment in which it works.

M. A. Llama (✉) · W. Z. Castañon · R. Garcia-Hernandez
Instituto Tecnológico de La Laguna, Blvd. Revolución y Av.
Instituto Tecnológico de La Laguna S/N, Torreón, Coahuila, Mexico
e-mail: mllama@correo.itlalaguna.edu.mx

O. O. Vergara Villegas et al. (eds.), *Advanced Topics on Computer Vision, Control and Robotics in Mechatronics*,
https://doi.org/10.1007/978-3-319-77770-2_7

During the interaction, a situation called constrained movement is presented, which is a set of constraints caused by the environment during the tracking of the geometric trajectories that the robot manipulator's end-effector can follow. In this case, the force feedback becomes mandatory to achieve a robust behavior, and a safe and reliable operation, things that are very difficult to achieve with a pure movement control.

The interaction state of the manipulator with its surrounding can be described by means of estimations obtained from the presented forces in the joints of the manipulator, or more directly, from the contact force. This contact force is provided by a force/torque sensor mounted between the manipulator and the end-effector tool.

Up to date, a large number of studies have been carried out on the control of robot manipulators in interaction with the environment. Such controllers have been classified into different types according to their architecture. In the following paragraphs, the classification of the interaction controllers and their categories are described.

7.1.1 Interaction Control

The nature of the interaction between the manipulator and its environment allows classifying robotic applications in two classes. Those tasks that are involved without contact, that is, movements without restrictions in the free space, and complex robotic applications that require that the manipulator be mechanically coupled to other objects. Two categories can be distinguished within this last type of tasks. The first category is dedicated to essential tasks of force in which the end-effector is required to stabilize the physical contact with the environment and executes a specific force process. In the second category, the emphasis falls on the movement of the end-effector, which is performed on restricted surfaces (compliant motion).

7.1.1.1 Classification of Control Schemes with Restricted Movement

According to Vukobratovic et al. (2009), the type of compliance schemes can be classified as:

- Passive compliance: In which the position of the end-effector is accommodated by the contact forces themselves due to the inherent compliance in the structure of the manipulator, the servos, and the end-effector with special devices of high compliance.
- Active compliance: Where the adjustment is made through a force feedback loop in order to achieve a reaction of the robot, either through force interaction control or by generating a task with a very specific compliance.

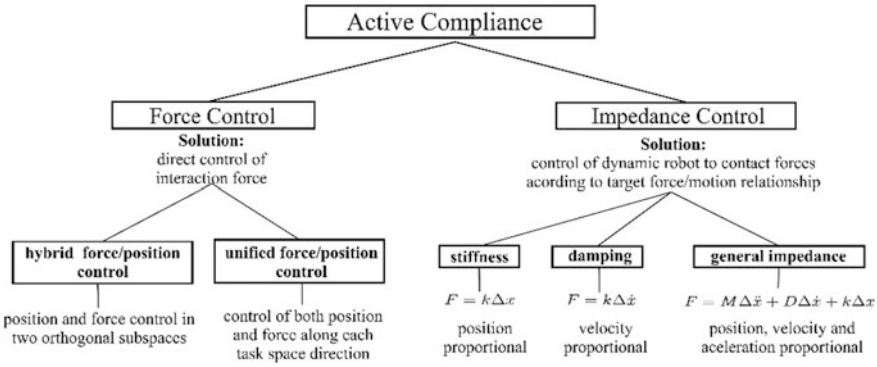


Fig. 7.1 Classification of active compliance controllers

We are interested only in the scheme of active compliance, and more specific, the hybrid control of force/position. Figure 7.1 (Vukobratovic et al. (2009) shows a control scheme that involve active compliance.

7.1.1.2 Hybrid Force/Position Control

This control methodology is based on the force and position control theory proposed by Mason (1981), depending on the mechanical and geometrical characteristics of the contact problem. This control methodology distinguishes two sets of constraints between the movement of the robot and the contact forces. The first set contains the so-called natural constraints, which arise due to the geometry of the task. The other set of constraints, called artificial constraints, is given by the characteristics associated with the execution of the specified task, i.e., the constraints are specified with respect to a framework, called a constraint framework. For instance, in a contact task where a slide is performed on a surface, it is common to adopt the Cartesian restriction framework in the way shown in Fig. 7.2 which is given in Vukobratovic et al. (2009). Assuming an ideally rigid and frictionless contact between the end-effector and the surface, it is obvious that natural constraints limit the movement of the end-effector in the direction of the z-axis, as well as rotations around the x- and y-axes.

The artificial constraints, imposed by the controller, are introduced to specify the task that will be performed by the robot with respect to the restriction framework. These restrictions divide the possible degrees of freedom (DOF) of the Cartesian movement into those that must be controlled in position and in those that must be controlled in force, in order to carry out the requested task.

In the implementation of a hybrid force/position control, it is essential to introduce two Boolean matrices S and $\overline{S}$ in the feedback loops in order to filter the forces and displacements sensed in the end-effector, which are inconsistent with the contact model of the task. The first is called the compliance selection matrix, and according

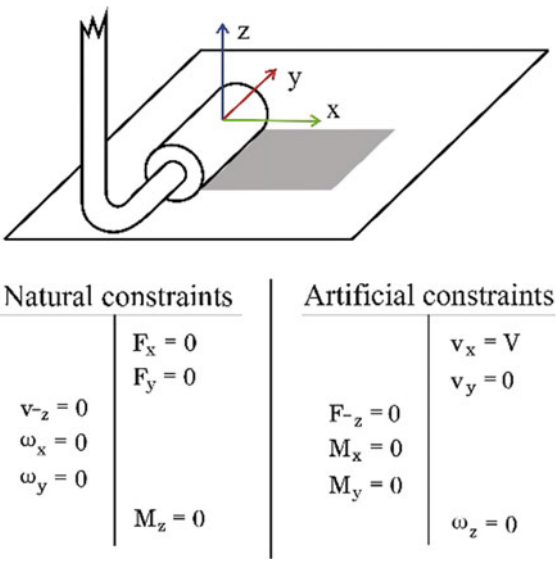


Fig. 7.2 Specification of the natural and artificial constraints for the task of sliding on a surface by means of the hybrid force/position controller

to the artificial constraints specified, the *ith* diagonal element of this matrix has the value of 1 if the *i*th DOF with respect to the frame of the task has to be controlled in force, and the value of 0 if it is controlled in position. The second matrix is the selection matrix for the DOF that is controlled in position; the *ith* diagonal element of this matrix has the value of 1 if the *ith* DOF with respect to the frame of the task has to be controlled in position, and the value of 0 if controlled in force.

To specify a hybrid contact task, the following sets of information have to be defined:

- Position and orientation of the frame of the task.
- The controlled directions in position and strength with respect to the frame of the task (selection matrix).
- The desired position and strength with respect to the frame of the task.

Once the contact task is specified, the next step is to select the appropriate control algorithm.

7.1.2 Fuzzy Logic

The concept of fuzzy logic was introduced for the first time in 1965 by Professor Zadeh (1965) as an alternative to describe sets in which there is vagueness or uncertainty and, consequently, cannot be easily defined.

Fuzzy logic or fuzzy set theory is a mathematical tool based on degrees of membership that allows modeling information which contains ambiguity, imprecision,

and uncertainty, by measuring the degree to which an event occurs, using for this a knowledge base or human reasoning.

7.1.2.1 Fuzzy Sets

A fuzzy set $A \in U$ can be represented as a set of ordered pairs of a generic element x and its membership value $\mu_A(x)$ that represents the degree to which the element x belongs to the fuzzy set A, that is,

$$A = \{(x, \mu_A(x)) | x \in U\}$$

A fuzzy set can also be expressed as a membership function $\mu_A(x)$, defined as $\mu_A(x)$: $A \in U \rightarrow [0, 1]$. This membership function assigns a membership value in A for each element x in the interval [1, 0]. This value is known as degree of membership.

7.1.2.2 Membership Functions

A membership function $\mu_A(x)$ can take different forms according to the system you want to describe. Among the most common forms are those described by impulsive, triangular, pseudo-trapezoidal, and Gaussian membership functions (Nguyen et al. 2003). The following describes the membership functions used later in this work.

Singleton membership function

A singleton membership function is shown in Fig. 7.3 and is defined by the following expression:

$$\delta(x; \bar{x}) = \begin{cases} 1, & \text{if} \quad x = \bar{x} \\ 0, & \text{if} \quad x \neq \bar{x} \end{cases}$$

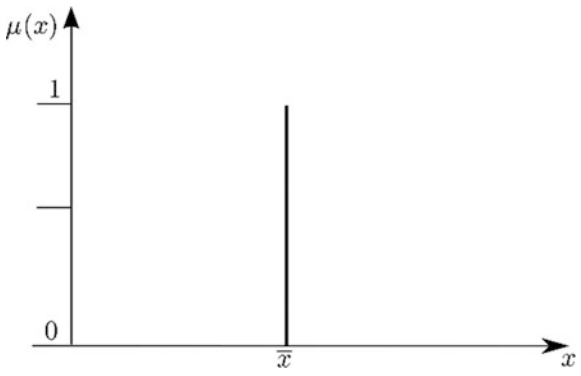


Fig. 7.3 Singleton membership function

Gaussian membership function

The Gaussian membership function G: $U \rightarrow [0, 1]$, shown in Fig. 7.4, has two parameters ρ and σ and is given by the expression

$$G(x; \rho, \sigma) = \mathrm{e}^{-\left(\frac{x-\rho}{\sigma}\right)^2}$$

Two-sided Gaussian membership function

The two-sided Gaussian membership function Υ: $U \rightarrow [0, 1]$ can be open to the left or open to the right and is created from the Gaussian function. For example, the left sigmoid membership function, which is shown in Fig. 7.5, is described by

$$\Upsilon(x; \rho, \sigma) = \begin{cases} 1, & \text{if} \quad x < \rho \\ \mathrm{e}^{-\left(\frac{x-\rho}{\sigma}\right)^2}, & \text{if} \quad x \geq \rho \end{cases}$$

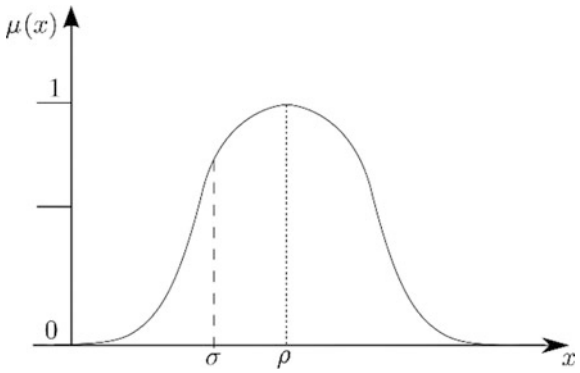


Fig. 7.4 Gaussian membership function

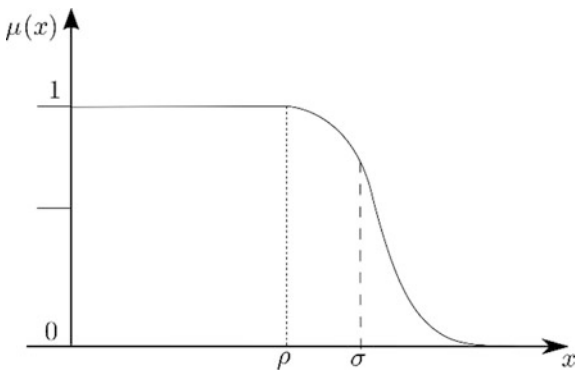


Fig. 7.5 Open to the left Gaussian membership function

7.1.3 Control Systems Based on Fuzzy Logic

Fuzzy controllers are constructed from a set of fuzzy rules based on the knowledge of the control system and the experience of operators and experts. A fuzzy rule is expressed by

$$\text{IF } x \text{ is } A \text{ THEN } y \text{ is } B$$

where A and B are fuzzy sets.

7.1.3.1 Fuzzy Controller Structure

The structure of a fuzzy controller is composed of three main modules, shown in Fig. 7.6: fuzzification module, fuzzy rule base and inference engine module, and defuzzification module.

Fuzzification

During the transition of the fuzzification stage, the n input variables x_i (i = 1, 2, 3, … n) are received, and for each of the N_i fuzzy sets $A_i^{l_i}$, a degree of membership $\mu_{A_i^{l_i}}(x_i^*)$ is assigned to the actual input values x_i^*, where l_i is used to identify each fuzzy set corresponding to each input i.

An important aspect when using fuzzy systems in real-time applications is to take into account the computing time to perform operations. Taking into account this, one method that reduces the computation time is the singleton fuzzification shown in Fig. 7.7, and it is defined by

$$\mu_{A_i^{l_i}}(x_i^*) = \begin{cases} \mu_{A_i^{l_i}}(x_i^*), & \text{if} \quad x_i = x_i^* \\ 0, & \text{if} \quad x_i \neq x_i^* \end{cases}$$

Fuzzy rule base and inference engine

The fuzzy rule base stores the knowledge and previous experience of the plant in which the fuzzy system will be based on. The fuzzy rule base is constructed by a group of fuzzy rules of the type IF–THEN denoted by $R_e^{l_1 \cdots l_n}$ and described as follows

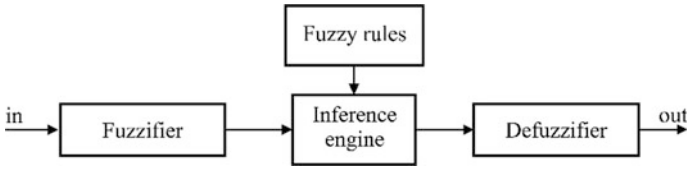


Fig. 7.6 Fuzzy controller general structure

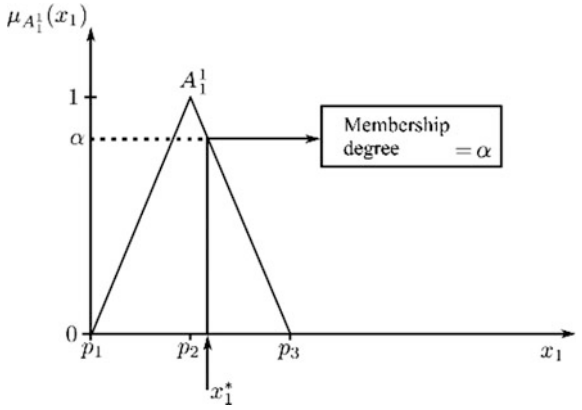


Fig. 7.7 Singleton fuzzification

Table 7.1 Fuzzy Associative Memory (FAM)

$x_2 \backslash x_1$	$l_1 = 1, A_1^1$	$l_1 = 2, A_1^2$	$l_1 = 3, A_1^3$
$l_2 = 1, A_2^1$	B^{11}	B^{21}	B^{31}
$l_2 = 2, A_2^2$	B^{12}	B^{22}	B^{32}
$l_2 = 3, A_2^3$	B^{13}	B^{23}	B^{33}

$$R_e^{l_1 \cdots l_n} : \text{ IF } x_i \text{ is } A_i^{l_i} \text{ AND} \cdots \text{AND } x_n \text{ is } A_n^{l_n} \text{ THEN } y \text{ is } B^{l_1 \cdots l_n} \tag{7.1}$$

where x_i (i = 1, 2, …, n) are the input variables, and y is the output variable. The total number of rules is expressed by $M = \prod_{i=1}^{n} N_i$.

The storage of the fuzzy rule base is done through a table called Fuzzy Associative Memory (FAM) or Lookup table, such as shown in Table 7.1, which stores a fuzzy rule base with two entries (x_1 and x_2) and three membership functions for each entry.

On the other hand, we call inference to the process in which, according to the rule base, the membership values for each output membership function are computed. Inference can be seen as the process: Given a fuzzy relationship, established in advance, between an input fuzzy set A and an output fuzzy set B, i.e., IF A THEN B, we could conclude how an output B' would look like given a new input A'; i.e., IF A' THEN B'.

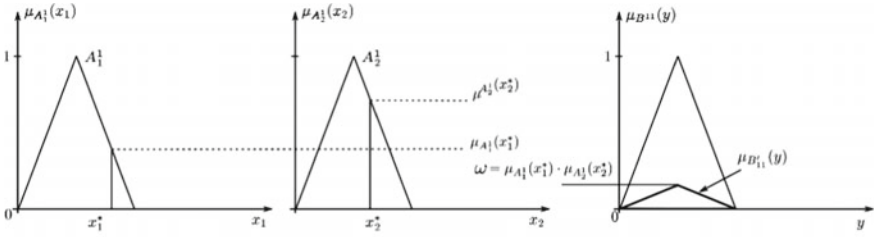
There exist several methods to perform the inference process; these different methods are known as inference engines. Among the most common inference engines are the max–min inference engine which uses the *min* operator between the antecedents of the rules, and the max-prod inference engine which uses the product

operator between the antecedents of the rules; the latest is faster and that is the reason it is the most used in real-time applications.

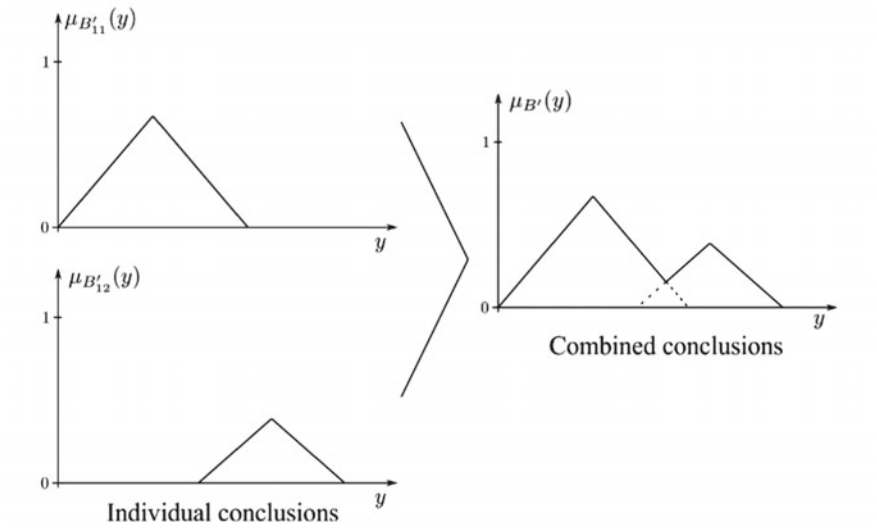
max-prod inference

Taking into account the set of rules of the form Eq. (7.1), with input membership functions $\mu_{A_i^{l_i}}(x_i)$ and output $\mu_{B^{l_1 \cdots l_n}}(y)$ for all $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T \in U \subset \mathbb{R}^n$ and $y \in V \subset \mathbb{R}$, the product inference engine is given as

$$\mu_{B'_{l_1 l_2}}(y, \boldsymbol{x}^*) = \left(\mu_{A_1^{l_1}}(x_1^*) \cdot \mu_{A_2^{l_2}}(x_2^*) \cdot \mu_{B^{l_1 l_2}}(y)\right) \tag{7.2}$$



(a) Product inference



(b) Combined conclusions

Fig. 7.8 Max-prod inference engine

Figure 7.8a describes the process of the max-product inference engine, while Fig. 7.8b describes the process of combining, by union operation, the output of several rule conclusions.

Defuzzification

In the defuzzification stage, a scalar value y^* is generated from the output $\mu_{B'}(y)$ that generates the inference engine. This value y^* is the output of the fuzzy controller that will be applied to the system to be controlled.

There are several ways to compute the output of the fuzzy controller; the most common is the center of average defuzzification which is given as

$$y^*(\boldsymbol{x}^*) = \frac{\sum_{l_1=1}^{N_1} \cdots \sum_{l_n=1}^{N_n} \bar{y}^{l_1 \cdots l_n} \omega_{l_1 \cdots l_n}(\boldsymbol{x}^*)}{\sum_{l_1=1}^{N_1} \cdots \sum_{l_n=1}^{N_n} \omega_{l_1 \cdots l_n}(\boldsymbol{x}^*)} \tag{7.3}$$

where $\bar{y}^{l_1 \cdots l_n}$ is the center of the $l_1 \ldots l_n$ output fuzzy sets, while $\omega_{l_1 \cdots l_n}$ is the height of the input membership functions, and $\boldsymbol{x}^*$ is the set of input real values.

7.2 Mitsubishi PA10-7CE Robot Arm

The Mitsubishi industrial robot manipulator PA10-7CE is one of the versions of the "Portable General-Purpose Intelligent Arm" of open architecture, developed by Mitsubishi Heavy Industries (MHI). This manipulator is composed of seven joints connected through links as shown in Fig. 7.9. The servomotors of the PA10 are three-phase brushless type and are coupled to the links by means of harmonic drives and electromagnetic brakes. In this work, the Mitsubishi is used as a 6-DOF robot arm; i.e, one of the joints is blocked, in this case the joint 3 represented by S3 in Fig. 7.9.

7.2.1 Robot Dynamics

The dynamic equation of motion for a manipulator of n DOF in interaction with the environment is expressed by Vukobratovic et al. (2009)

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} + J^T(\boldsymbol{q})\boldsymbol{f}_s \tag{7.4}$$

where $\boldsymbol{q}$ is an $n \times 1$ vector of joint displacements, $\dot{\boldsymbol{q}}$ is an $n \times 1$ vector of joint velocities, $\boldsymbol{\tau}$ is an $n \times 1$ vector of actuators applied torques, $M(\boldsymbol{q})$ is an $n \times n$ symmetric positive definite manipulator inertia matrix, $C(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is an $n \times n$ matrix of centripetal and Coriolis torques, and $\boldsymbol{g}(\boldsymbol{q})$ is an $n \times 1$ vector of gravitational torques obtained as the gradient of the potential energy $U(\boldsymbol{q})$ due to gravity, $J(\boldsymbol{q}) \in \mathbb{R}^{(d+m)\times n}$ is the geometric Jacobian matrix that relates velocities in joint space with velocities in operational space; $\boldsymbol{f}_s \in \mathbb{R}^{(d+m)}$ is the vector of forces

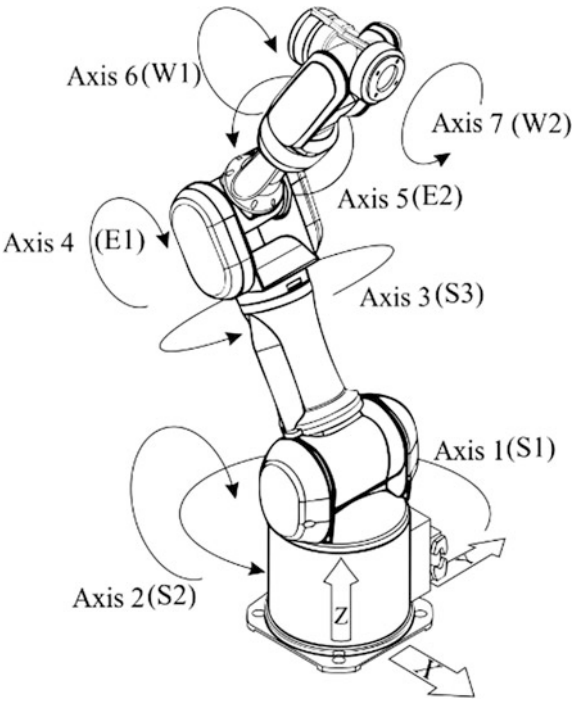


Fig. 7.9 Mechanical structure of the Mitsubishi PA10-7CE robot manipulator

and applied torques in end-effector of the manipulator, where d is the dimension of the geometric space in which the robot moves and m is the dimension of the vector space which defines the orientation. In order to avoid the resolution problem of the manipulator PA10-7CE redundancy, let the number of degrees of freedom be $n = 6$.

The matrices $M(\boldsymbol{q})$, $C(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and vector $\boldsymbol{g}(\boldsymbol{q})$ were obtained by using the software HEMERO (MATLAB–Simulink toolbox for the study of manipulators and mobile robots) (Maza and Ollero 2001) in a numerical and semi-symbolic way. Some of the instructions that this package offers, and those that were used here, are

- *inertia* (*dyn*, $\boldsymbol{q}$) to obtain the inertia matrix $M(\boldsymbol{q})$
- *coriolis* (*dyn*, $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$) to obtain the Coriolis matrix $C(\boldsymbol{q}, \dot{\boldsymbol{q}})$
- *gravity* (*dyn*, $\boldsymbol{q}$) to obtain the gravitational torques vector $\boldsymbol{g}(\boldsymbol{q})$

where *dyn* is a matrix containing the kinematic and dynamic parameters of the manipulator. The results obtained were compared with the results in Salinas (2011), and it was verified that the previously obtained model is correct. For this reason, the dynamic model obtained is not included in this work but can be seen in Salinas (2011).

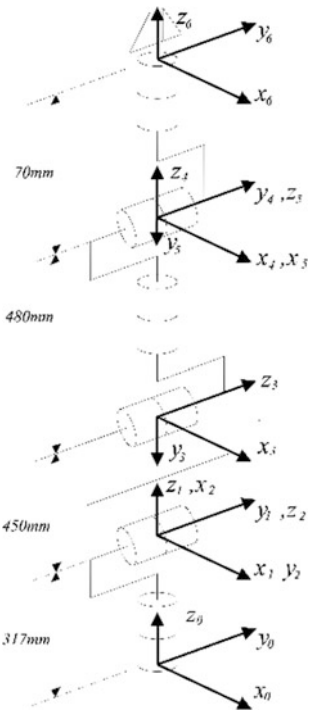


Fig. 7.10 Kinematic scheme for D-H parameters of 6-DOF reduced PA10-7CE robot manipulator

Table 7.2 D-H parameters of 6-DOF reduced PA10-7CE robot manipulator

Link	a_{i-1}[m]	α_{i-1}[rad]	d_i[m]	θ_i[rad]
1	0	0	0.317	q_1
2	0	$-\frac{\pi}{2}$	0	$q_2 - \frac{\pi}{2}$
3	0.450	0	0	$q_2 + \frac{\pi}{2}$
4	0	$\frac{\pi}{2}$	0.480	q_4
5	0	$-\frac{\pi}{2}$	0	q_5
6	0	$\frac{\pi}{2}$	0.070	q_6

7.2.2 Robot Kinematics

To obtain the manipulator's kinematics, the Denavit–Hartenberg convention described in Craig (2006) was used. The Denavit–Hartenberg parameters for the Mitsubishi PA10 robot were obtained with the frames assigned in Fig. 7.10 and are shown in Table 7.2.

7.2.2.1 Forward Kinematics

The position direct kinematic model of a robot manipulator is the relation that allows determining the vector $\boldsymbol{x} \in \mathbb{R}^{d+m}$ of operational coordinates according to its articular configuration $\boldsymbol{q}$:

$$\boldsymbol{x} = \boldsymbol{h}(\boldsymbol{q}) \tag{7.5}$$

where $\boldsymbol{h}$ is a vector function. Equation (7.5) is known as the direct kinematic equation of the robot manipulator (Sciavicco and Siciliano 1996). The components of the function $\boldsymbol{h}$ are determined implicitly by the product of matrices of homogeneous transformation of the manipulator.

$$^{0}_{n}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T \ldots {}^{n-1}_{n}T \tag{7.6}$$

In general, the homogenous transformation matrix ${}^{a}_{b}T \in SE(3) \subset \mathbb{R}^{4\times4}$ that describes the relative position of the frame $\sum_b$ with respect to the frame $\sum_a$ is given by

$${}^{a}_{b}T = \begin{bmatrix} {}^{a}_{b}R & {}^{a}_{b}p \\ 0^{T} & 1 \end{bmatrix} \in \mathrm{SE}(3) \subset \mathbb{R}^{4\times4} \tag{7.7}$$

where ${}^{a}_{b}\boldsymbol{p} \in \mathbb{R}^{3}$ describes the position and $R \in SO(3) \subset \mathbb{R}^{3\times3}$ is a matrix that describes the orientation.

The homogenous transformation matrix ${}^{0}_{n}T$ (with $n = 6$) for the Mitsubishi PA10-7CE robot was carried out using the HEMERO tool. The function used for that tool was

$$\mathrm{fkine}(dh, \boldsymbol{q})$$

where dh is a matrix with the Denavit–Hartenberg parameters with the following format

$$\begin{bmatrix} \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{i-1} & \alpha_{i-1} & d_i & \theta_i & \sigma_i \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} \in \mathbb{R}^{n\times5}$$

where

- $a_{i-1}, \alpha_{i-1}, d_i, \theta_i$ are the Denavit–Hartenberg parameters according to Craig (2006).
- σ_i indicates the type of joint (it takes the value of 0 for a rotational joint and 1 if it is a prismatic one).

The elements of the homogeneous transformation matrix ${}^{0}_{n}T$ are shown in Castañon (2017).

Taking the time deriving of Eq. (7.5), we obtain

$$\dot{\boldsymbol{x}} = J_A(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{7.8}$$

where $J_A(\boldsymbol{q}) \in \mathbb{R}^{(d+m)\times n}$ is the analytic Jacobian matrix of the robot. This matrix can be found in Salinas (2011). The geometric Jacobian is obtained using HEMERO tool using the following instruction

$$\text{jacob0}(dh, \boldsymbol{q})$$

The elements of this matrix can be found in Castañon (2017).

7.2.2.2 Inverse Kinematics

The position inverse kinematic model is the inverse function $\boldsymbol{h}^{-1}$ that if it exists for a given robot, it allows obtaining the necessary configuration to locate its end-effector in a given position $\boldsymbol{x}$:

$$\boldsymbol{q} = \boldsymbol{h}^{-1}(\boldsymbol{x}) \tag{7.9}$$

The expressions of the $\boldsymbol{h}^{-1}$ function of the position inverse kinematic model were calculated with the help of SYMORO+ robotics software (Khalil et al. 2014), and the results are shown in Castañon (2017).

Finally, from Eq. (7.8) the expression that characterizes the velocity inverse kinematic model is given by

$$\dot{\boldsymbol{q}} = J_A(q)^{-1}\dot{\boldsymbol{x}} \tag{7.10}$$

Fig. 7.11 ATI Delta force/torque sensor

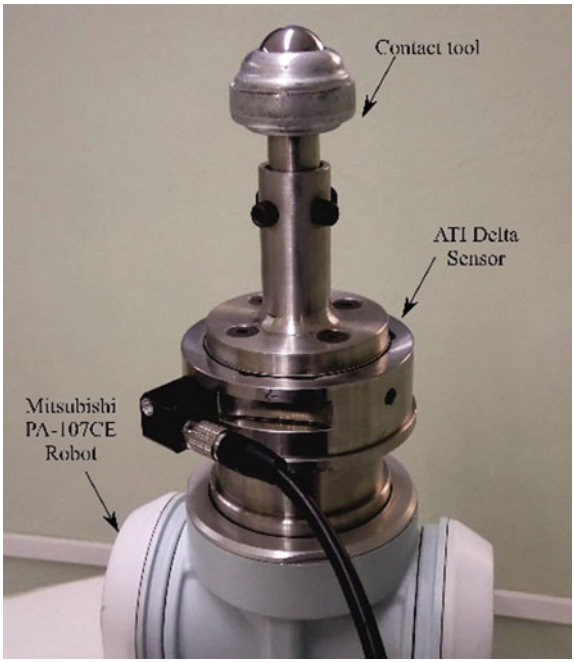


Fig. 7.12 Coupling between PA10-7CE robot, ATI Delta sensor, and contact tool

7.2.3 *Force/Torque Sensor*

The hybrid force/position controller requires the feedback of the forces and torques present in the robot's end-effector or in the contact tool used; to achieve this, the robot was fitted with a Delta model ATI force sensor shown in Fig. 7.11. This is a sensor of 6 degrees of freedom; this means that it is able to acquire the forces and torques in each of the Cartesian axes (F_x, F_y, F_z, T_x, T_y, T_z).

The main characteristics of the ATI Delta sensor are shown in Castañon (2017). For more technical information, consult (ATI Industrial Automation 2018a, b).

The ATI sensor was paired with a NI PCI-6220 DAQ card fitted in the control computer. Once the voltages signals read by the DAQ are in the MATLAB/Simulink environment, these are converted to force/torque values. This conversion is given by the expression

$$\boldsymbol{f}_s = M_T \boldsymbol{v}_c + \boldsymbol{c}_o \tag{7.11}$$

where M_T is a 6×6 transformation matrix provided in the calibration file of the sensor, $\boldsymbol{v}_c \in \mathbb{R}^6$ is the vector containing the voltages sent by each gauge on the sensor, and $\boldsymbol{c}_o \in \mathbb{R}^6$ is a compensation vector that is also provided in the calibration file of the sensor found in Castañon (2017).

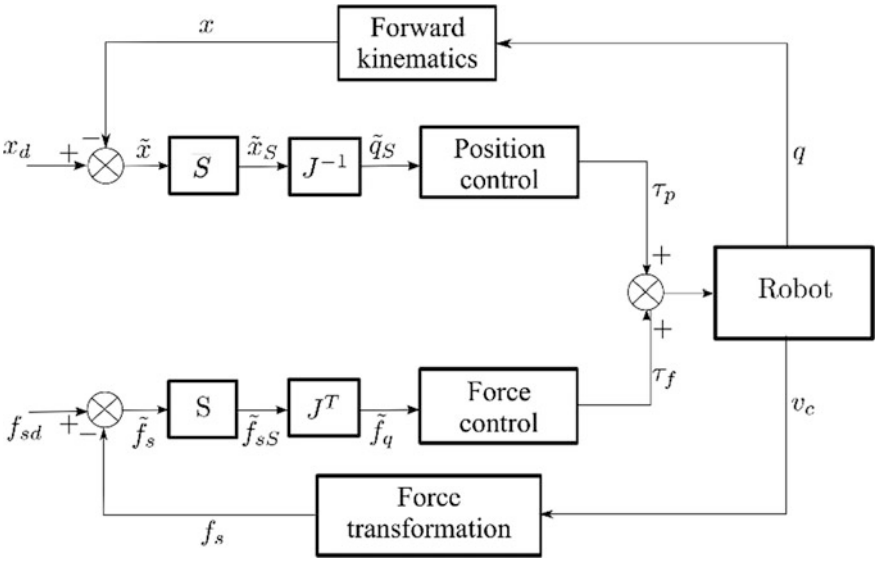


Fig. 7.13 Hybrid controller original structure

The sensor ATI was mounted between the last link of the robot and a special contact tool designed in Salinas (2011) to reduce, as much as possible, the friction between this tool and the contact surface; this is illustrated in Fig. 7.12.

7.3 Hybrid Force/Position Control with Fixed Gains

In the literature, there is a wide collection of works about different and very varied algorithms of hybrid force/position control; however, one of the most important approaches is the one proposed by Craig and Raibert (1979), which is shown in Fig. 7.13.

It contains two control loops in parallel with independent control and feedback laws for each one. The first loop is the position control which makes use of the information acquired by the position sensors in each robot joint. The second loop is the force control. This loop uses the information collected by the force sensor mounted on the end-effector. The matrix S is used to select which link will be controlled in either position or force.

In the direction controlled in force, the position errors are set to zero when multiplied by the orthogonal complement of the selection matrix (position selection matrix) defined as $\overline{S} = I - S$. This means that the part of the position control loop does not interfere with the force control loop; however, this is not the real case because there is still some coupling between both control loops.

A PD-type position control law is used with gain matrices $k_p \in \mathbb{R}^{(d+m)\times(d+m)}$ and $k_v \in \mathbb{R}^{(d+m)\times(d+m)}$, while the force control law consists of a proportional and integral action PI with their respective matrices' gains $k_{pf} \in \mathbb{R}^{(d+m)\times(d+m)}$ and $k_{if} \in \mathbb{R}^{(d+m)\times(d+m)}$, as well as a part of the feedback of the desired force in the force loop, then the control law can be written in operational space as

$$\boldsymbol{\tau_x} = \boldsymbol{\tau_{fx}} + \boldsymbol{\tau_{px}} \tag{7.12}$$

where the control torque of the position loop is given by

$$\boldsymbol{\tau_{px}} = k_p \overline{S}\tilde{\boldsymbol{x}} + k_v \overline{S}\dot{\tilde{\boldsymbol{x}}} \tag{7.13}$$

and for the force loop as

$$\boldsymbol{\tau_{fx}} = k_{pf}\overline{S}\widetilde{\boldsymbol{f}}_s + k_{if} S \int_0^t \widetilde{\boldsymbol{f}}_s \mathrm{d}t + \boldsymbol{f_{sd}} \tag{7.14}$$

where $\boldsymbol{\tau_x} \in \mathbb{R}^6$ is the vector of control torques; k_p, k_v, k_{pf} and k_{if} are the 6×6 control gain diagonal matrices; $\tilde{\boldsymbol{x}}$ is the vector resulting from the difference between the desired operational posture vector $\boldsymbol{x_d} \in \mathbb{R}^{d+m}$ and the actual operational posture vector $\boldsymbol{x} \in \mathbb{R}^{d+m}$; $\dot{\tilde{\boldsymbol{x}}} \in \mathbb{R}^{d+m}$ is the vector of speed errors in operational space; and $\tilde{\boldsymbol{f}}_s \in \mathbb{R}^{d+m}$ is the vector obtained by the difference between the desired contact forces vector $\boldsymbol{f_{sd}} \in \mathbb{R}^{d+m}$ and the instant force vector $\boldsymbol{f_{sd}} \in \mathbb{R}^{d+m}$.

A problem that arises in this formulation is a dynamic instability in the force control part due to high gain effects of the feedback from the force sensor signal that is caused when a high rigidity is present in the environment, unmodeled dynamics effects caused by the arm and the elasticity of the sensor. To solve this problem, the dynamic model of the manipulator was introduced into the control law. In Shin and Lee (1985), a hybrid control of force/position is formulated in which the dynamic model of the robot is used in the control law; the expression is given by

$$\boldsymbol{\tau_x} = M_x(\boldsymbol{x})\ddot{\boldsymbol{x}}^* + C_x(\boldsymbol{x}, \dot{\boldsymbol{x}}) + \boldsymbol{g_x}(\boldsymbol{x}) + S\boldsymbol{f}^* \tag{7.15}$$

where $\ddot{\boldsymbol{x}}^*$ is the equivalent acceleration control,

$$\ddot{\boldsymbol{x}}^* = \ddot{x}_d + k_v\dot{\tilde{\boldsymbol{x}}} + k_p\tilde{\boldsymbol{x}} \tag{7.16}$$

and $\boldsymbol{f}^* \in \mathbb{R}^6$ is the vector generated by the control law selected for the part of the force loop.

To avoid rebounding and minimize overshoots during the transition, an active damping term is added in the force control part (Khatib 1987).

$$\boldsymbol{\tau}_{\boldsymbol{fx}} = S\boldsymbol{f}^* - M_x(\boldsymbol{x})Sk_{vf}\dot{\boldsymbol{x}} \tag{7.17}$$

where the term k_{vf} is a diagonal matrix with Cartesian damping gains. In Bona and Indri (1992), it is proposed to modify the control law for position as

$$\boldsymbol{\tau}_{\boldsymbol{px}} = M_x(\boldsymbol{x})\overline{S}\left[\ddot{\boldsymbol{x}}^* - M_x^{-1}(\boldsymbol{x})(S\boldsymbol{f}^* - \boldsymbol{f}_s)\right] + C_x(\boldsymbol{x},\dot{\boldsymbol{x}}) + \boldsymbol{g}_{\boldsymbol{x}}(\boldsymbol{x}) \tag{7.18}$$

being $M_x^{-1}(\boldsymbol{x})(S\boldsymbol{f}^* - \boldsymbol{f}_s)$ an added term to compensate the coupling between the force and position control loops, as well as the disturbances in the position controller due to the reaction force.

So far, the control laws have been handled in operational space; however in Zhang and Paul (1985), a transformation of the Cartesian space to joint space is proposed by transforming the selection matrices S and $\overline{S}$ given in Cartesian space to joint space as

$$S_q = J^{-1}SJ \tag{7.19}$$

and

$$\overline{S}_q = J^{-1}\overline{S}J \tag{7.20}$$

where J is the geometric Jacobian. With these transformations, the equivalent control law in joint space is obtained as

$$\boldsymbol{\tau} = \boldsymbol{\tau}_f + \boldsymbol{\tau}_p \tag{7.21}$$

being

$$\boldsymbol{\tau}_f = S_q\boldsymbol{f}_c^* - M(\boldsymbol{q})S_qK_{vf}\dot{\boldsymbol{q}} \tag{7.22}$$

$$\boldsymbol{\tau}_p = M(\boldsymbol{q})\overline{S}_q\left[\ddot{\boldsymbol{q}}^* - M(\boldsymbol{q})^{-1}(S_q\boldsymbol{f}_c^* - J^T\boldsymbol{f}_s)\right] + C(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) \tag{7.23}$$

where

$$\ddot{\boldsymbol{q}}^* = \ddot{\boldsymbol{q}}_d + K_v\dot{\tilde{\boldsymbol{q}}} + K_p\tilde{\boldsymbol{q}} \tag{7.24}$$

and

$$\boldsymbol{f}_c^* = K_{pf}J^T\tilde{\boldsymbol{f}}_s + K_{if}J^T\int_0^t \tilde{\boldsymbol{f}}_s \mathrm{d}t \tag{7.25}$$

$M(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$, $C(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ and $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$ are the dynamic joint components of the manipulator; $\tilde{\boldsymbol{q}} \in \mathbb{R}^n$ is the vector of differences between the vector of desired

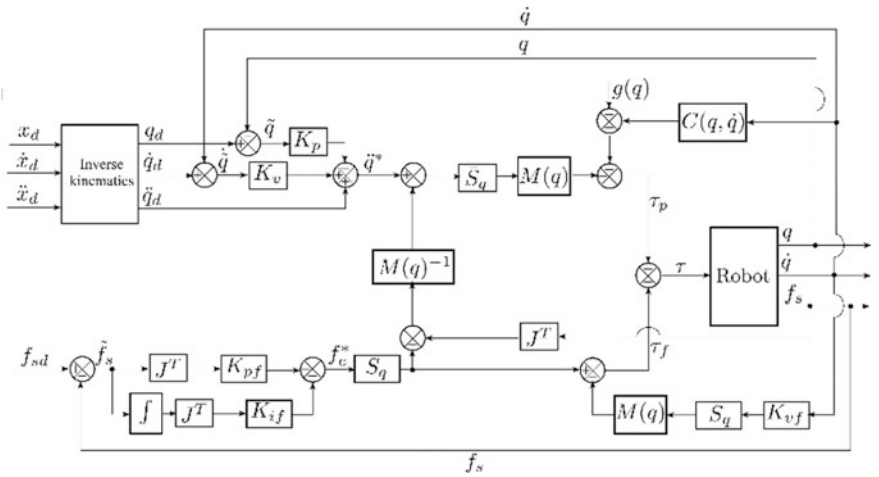


Fig. 7.14 Block diagram for hybrid controller with fixed gains in joint space

joint positions $\boldsymbol{q_d} \in \mathbb{R}^n$ and the vector of instantaneous joint positions $\boldsymbol{q} \in \mathbb{R}^n$; $K_v \in \mathbb{R}^{n\times n}$, $K_p \in \mathbb{R}^{n\times n}$ are gain diagonal matrices for the position control loop, and $K_{pf} \in \mathbb{R}^{n\times n}$, $K_{if} \in \mathbb{R}^{n\times n}$, and $K_{vf} \in \mathbb{R}^{n\times n}$ are gain diagonal matrices for the loop of force control in joint space.

The block diagram of the hybrid controller with fixed gains in joint space is shown in Fig. 7.14.

7.4 Hybrid Force/Position Control with Fuzzy Gains

Based on experimental results with the control law expressed by Eqs. (7.21)–(7.25) performed in the Mitsubishi PA10 robot arm, it was observed that the control performance changes depending on the rigidity of the contact environment; hence, to improve the performance from one surface to another, it was necessary to retune the control gains. A similar approach was proposed by Shih-Tin and Ang-Kiong (1998) but in a hierarchical way by tuning the scaling factor of the fuzzy logic controller.

Our experimental results showed that K_{pf} was the most sensible gain compared to K_{vf} and K_{if} gains. For this reason, we proposed that only the gain K_{pf} be supervised in a fuzzy manner and that K_{vf} and K_{if} were configured with constant values.

The proposed fuzzy control design is based on the control laws of Eqs. (7.21)–(7.25) with the difference that a supervisory fuzzy system is used to tune in the control gain $\widehat{K}_{pf}$ in the force control loop. Since now the gain is given by the function $\widehat{K}_{pf}(x)$, the Eq. (7.25) in the force control loop is given by

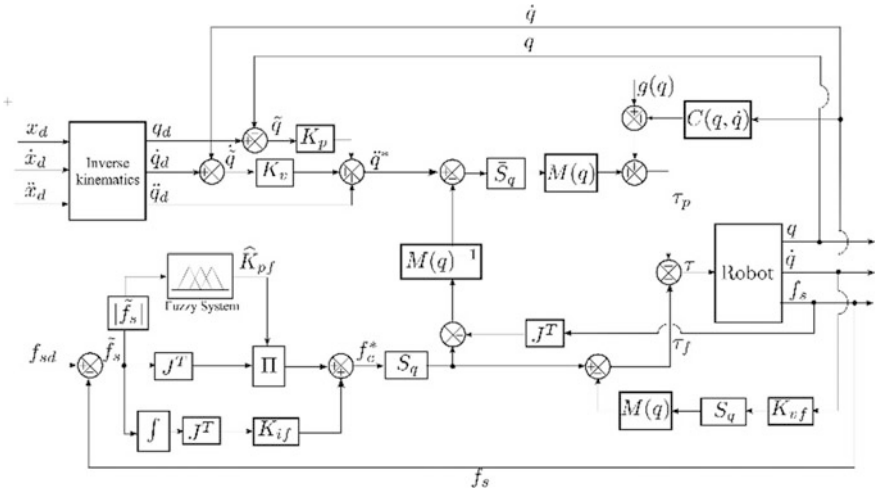


Fig. 7.15 Block diagram for hybrid controller with fuzzy gains in joint space

$$\boldsymbol{f}_{\boldsymbol{c}}^{*} = \widehat{K}_{pf}(x)J^{T}\tilde{\boldsymbol{f}}_{\boldsymbol{s}} + K_{if}J^{T}\int_{0}^{t}\tilde{\boldsymbol{f}}_{\boldsymbol{s}}dt \tag{7.26}$$

A fuzzy system $\widehat{K}_{pf}(x)$ like that represented in (7.3) with one input $x_1 = \tilde{f}_{sz}$ and one output $\bar{y}_1^{l_1}$ is designed. We define N_1 fuzzy sets $A_1^{l_1}(l_1 = 1, 2, \ldots, N_1)$ for input x_1, each of them described by a Gaussian membership function $\mu_{A_1^{l_1}}(x_1)$. For the output, singleton membership functions are selected.

The fuzzy system can be built from the set of N_1 fuzzy IF–THEN rules of the form

$$\text{IF } x_1 \text{ is } A_1^{l_1} \text{ THEN } \widehat{K}_{pf}(x) \text{ is } \bar{y}_1^{l1}$$

If singleton fuzzification, product inference engine, and center average defuzzification are used, the system can be described by

$$\widehat{K}_{pf}(x) = \frac{\sum_{l_1=1}^{N_1}\bar{y}_1^{l_1}\left(\mu_{A_1^{l_1}}(x_1)\right)}{\sum_{l_1=1}^{N_1}\left(\mu_{A_1^{l_1}}(x_1)\right)} \tag{7.27}$$

The block diagram of the hybrid controller with fuzzy gains in joint space is shown in Fig. 7.15.

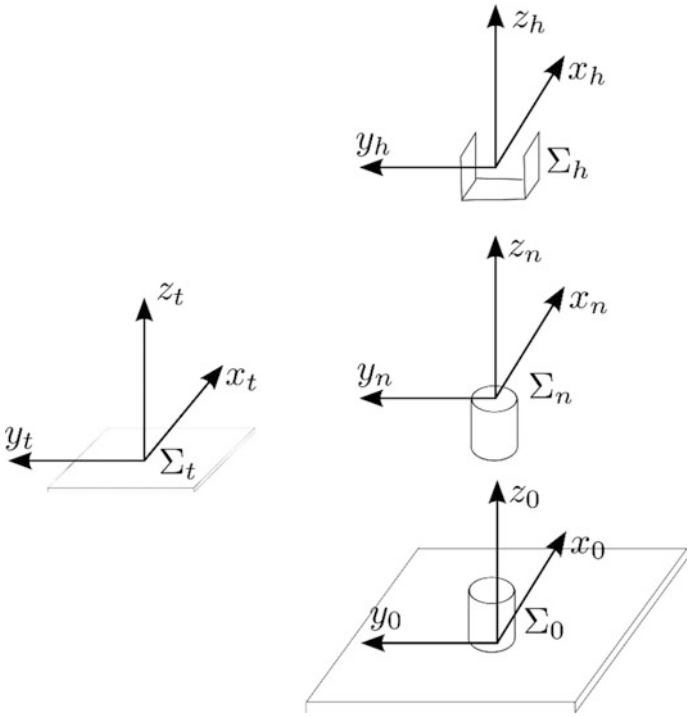


Fig. 7.16 Situation of the task frame referred to the frame of the base of the robot

7.5 Experimental Results

The controller was programmed in the MATLAB/Simulink environment. Also, the QUARC-BLOCKSET-PA10-Real-Time Control Software for Windows was used, which provides a simple way to make a communication between the internal control system of the PA10-7CE robot arm and the Simulink environment, thus obtaining easily the signals of the joint positions and joint speeds of the robot and send the control signals in torque mode back to the manipulator. In addition, the voltage signals provided by the Delta ATI force sensor could be also read and used within Simulink environment. The sampling time used for the whole system was 5 ms.

The task to be performed is to apply a desired force f_{szd} only on the z-axis of the frame of the task Σ_t (frame associated with where the contact is made) and which coincides in orientation with the base frame of the robot Σ_0 (frame of reference of the robot), with the frame of the last link Σ_n and with the frame of the end-effector Σ_h (see Fig. 7.16), while a tracking task of a straight line is carried out on the x-axis of the frame of the task.

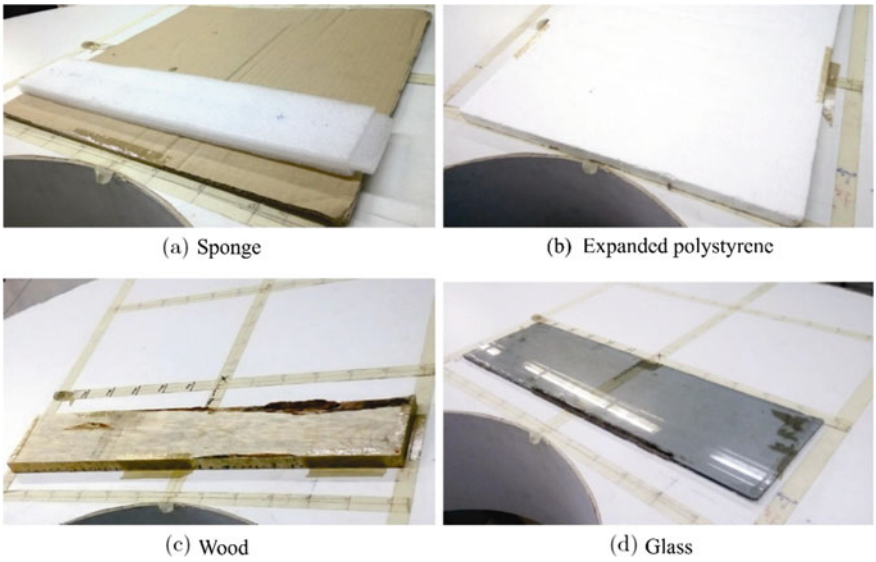


Fig. 7.17 Materials used as contact surface

Different materials were selected, with different degrees of stiffness, to place them as contact surfaces. The selected materials with a large stiffness coefficient K_s, which have greater resistance to deformation, were a wood board with a thickness of ¾ in., and a glass with a thickness of ¼ in. Materials with a low K_s stiffness coefficient, which tend to deform more easily, were a block of expanded polystyrene and a sponge. The selected materials are shown in Fig. 7.17.

7.5.1 Hybrid Force/Position Control with Fuzzy Gains

The controller was implemented by using Eqs. (7.21)–(7.24) and (7.26). The selection matrices for position S and for force $\overline{S}$ were selected as Eqs. (7.28) and (7.29), respectively. On the other hand, the values of the diagonal gain matrix for the position control loop are given in Table 7.3 and the fixed gains for the force control loop are shown in Table 7.4. K_{pf} was selected as the variable gain, and a fuzzy logic tuner was implemented for tuning such a gain.

$$S = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{7.28}$$

Table 7.3 Gains of the position control loop τ_p in the hybrid position/force controller

Joint	K_{pp} [1/s^2]	K_{vp} [1/s]
1	1250	100
2	1750	15
3	2750	12.5
4	500	10
5	500	200
6	2500	9000

Table 7.4 Fixed parameters of the force control part τ_f in the hybrid force/position controller with fuzzy gains

	K_{vf} [1/rad]	K_{if}
Value	0.1	3

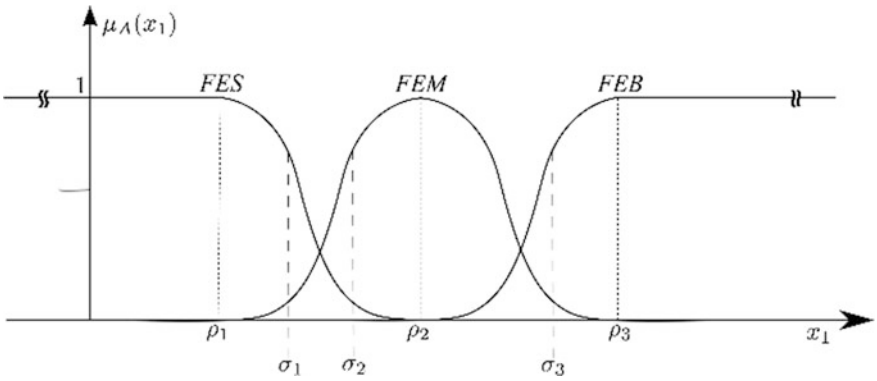


Fig. 7.18 Input membership functions

$$\overline{S} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.29}$$

To approximate the gain through the fuzzy system $\widehat{K}_{pf}(x)$, it receives an input $x_1 = \left|\tilde{f}_{sz}\right|$ with a universe of discourse partitioned into $N_1 = 3$ fuzzy sets: $A_1^1 =$ FES (Force Error Small), $A_1^2 =$ FEM (Force Error Medium), $A_1^3 =$ FEB (Force Error Big). To build the fuzzy system, we propose to use an open to the left Gaussian

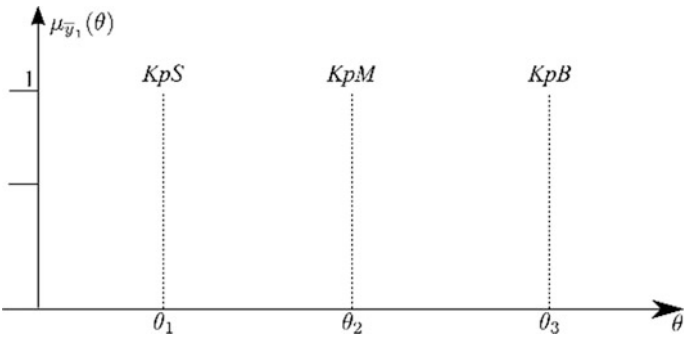


Fig. 7.19 Output membership functions

function, a Gaussian function, and an open to the right Gaussian function as shown in Fig. 7.18.

The partitions of the universe of discourse, using the notation $\rho_{A_1} = \{\rho_1, \rho_2, \rho_3\}$, were selected as

$$\rho_{|\tilde{f}_{sz}|} = \{0.5, 2, 4\}[\mathbf{N}]$$

and the standard deviations as

$$\sigma_{|\tilde{f}_{sz}|} = \{-6, 0.5, 3\}[\mathbf{N}]$$

As already mentioned, the fuzzy system consists of singleton functions for the output variable. The universe of discourse of the output is also partitioned into 3 impulsive functions: *KpS* (Small K_{pf} Gain), *KpM* (Medium K_{pf} Gain), and *KpB* (Big K_{pf} Gain); this is shown in Fig. 7.19, where each parameter θ corresponds to the position of the impulse functions. Taking the notation $\theta_{\bar{y}_1} = \{\theta_1, \theta_2, \theta_3\}$, the partitions of the universe of discourse for the output variable were selected like

$$\theta_{K_{pf}} = \{0.1, 0.15, 0.2\}$$

Fuzzy rules are selected as:

$$\text{IF } |\tilde{f}_{sz}| \text{ is FES THEN } \widehat{K}_{pf}(x) \text{ is } KpB$$

$$\text{IF } |\tilde{f}_{sz}| \text{ is FEM THEN } \widehat{K}_{pf}(x) \text{ is } KpM$$

$$\text{IF } |\tilde{f}_{sz}| \text{ is FEB THEN } \widehat{K}_{pf}(x) \text{ is } KpS$$

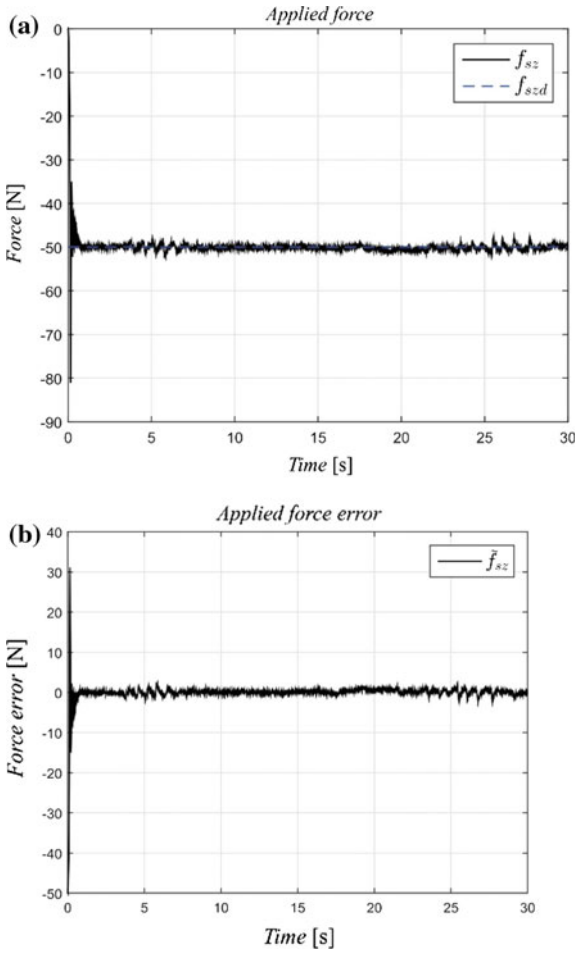


Fig. 7.20 **a** Applied force and **b** force error with the fuzzy gain hybrid controller for a desired force reference $f_{szd} = -50$ N on sponge

7.5.2 Force Regulation and Position Tracking on a Sponge

The experiments carried out on the Mitsubishi PA10 robot arm with the hybrid controller with fuzzy gains were made on different materials (shown in Fig. 7.17) and a desired force reference $f_{szd} = -50$ N (force applied down on the z-axis). Figure 7.20 shows the response f_{sz} to the applied force reference f_{szd} and the force error $\tilde{f}_{sz}$ on the z-axis applied on a sponge surface like the one in Fig. 7.17a.

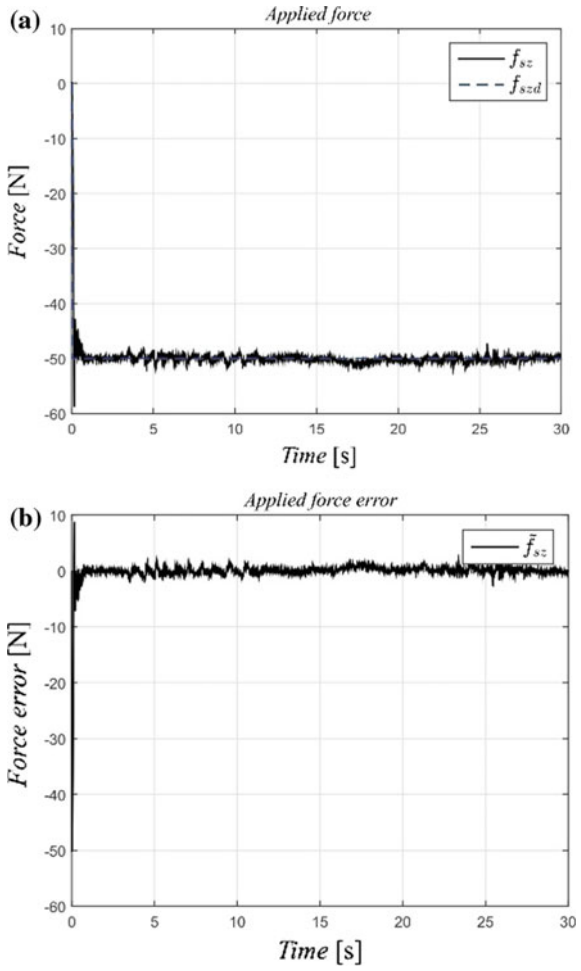


Fig. 7.21 **a** Applied force and **b** force error with the fuzzy gain hybrid controller for a desired force reference $f_{szd} = -50$ N on expanded polystyrene

7.5.3 Force Regulation and Position Tracking on Expanded Polystyrene

The results of the hybrid controller with fuzzy gains applied to the PA10 robot manipulator in interaction with an expanded polystyrene surface are shown in Fig. 7.21. This figure shows the response f_{sz} to the applied force reference $f_{szd} = -50$ N and the force error $\tilde{f}_{sz}$ on the z-axis applied on an expanded polystyrene surface like the one in Fig. 7.17b.

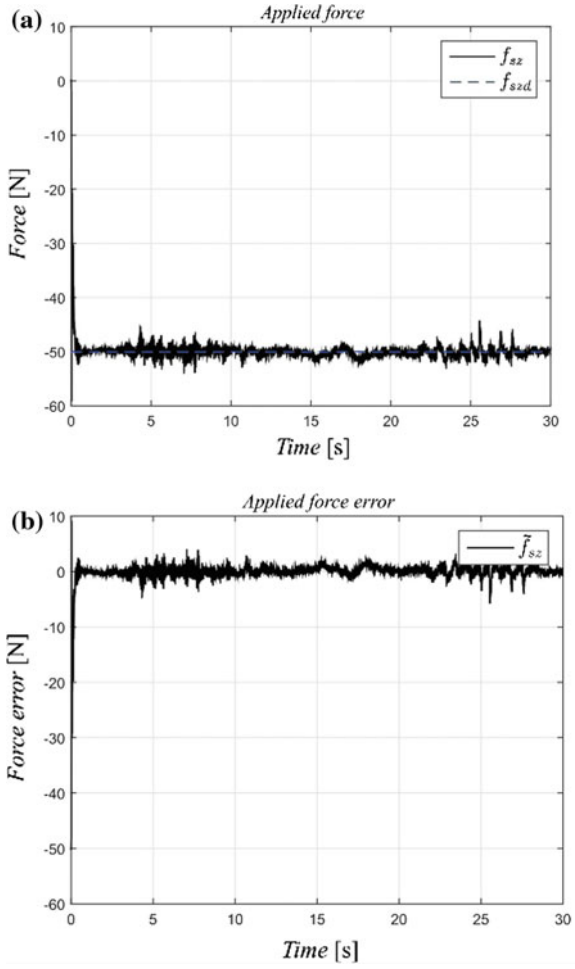


Fig. 7.22 **a** Applied force and **b** force error with the fuzzy gain hybrid controller for a desired force reference $f_{szd} = -50$ N on a wood board

7.5.4 Force Regulation and Position Tracking on a Wood Board

This section shows the results of the hybrid controller with fuzzy gains applied to the PA10 robot manipulator in interaction with a wood board surface like the one in Fig. 7.17c. Figure 7.22 shows the response f_{sz} to the applied force reference $f_{szd} = -50$ N and the force error $\tilde{f}_{sz}$ on the z-axis.

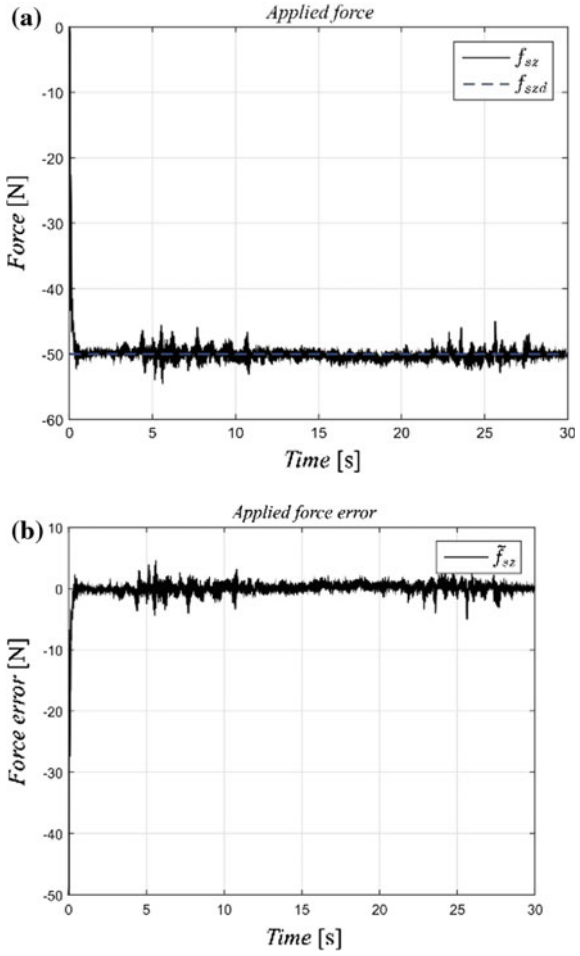


Fig. 7.23 **a** Applied force and **b** force error with the fuzzy gain hybrid controller for a desired force reference $f_{szd} = -50$ N on glass

7.5.5 Force Regulation and Position Tracking on Glass

The following figures show the results of the hybrid controller with fuzzy gains applied to the PA10 robot manipulator in interaction with a glass surface like the one in Fig. 7.17d. Figure 7.23 shows the response f_{sz} to the applied force reference $f_{szd} = -50$ N and the force error $\tilde{f}_{sz}$ on the z-axis.

The position and orientation errors are all very small, and they are reported in Castañon (2017).

7.6 Conclusions

The proposed hybrid force/position controller with fuzzy gains has the great advantage over its corresponding fixed gain controller that it does not require to retune the gains to exert a desired force in different types of materials with good performance. Conversely, the hybrid force/position controller with fixed gains requires the retuning of its gains for each material; in other words, for the fixed gain controller, the best gains obtained for soft materials cannot be used in hard materials because the system becomes unstable and very violent vibrations occur. This problem is not present in the proposed fuzzy version.

References

ATI Industrial Automation, I. (2018a). ATI F/T Catalogs and Manuals. Recuperado el 16 de enero de 2018, a partir de http://www.ati-ia.com/products/ft/ft_literature.aspx.

ATI Industrial Automation, I. (2018b). ATI Industrial Automation: Multi-Axis Force / Torque Sensors. Recuperado el 16 de enero de 2018, a partir de http://www.ati-ia.com/products/ft/sensors.aspx.

Bona, B., & Indri, M. (1992). Exact decoupling of the force-position control using the operational space formulation. In Proceedings of IEEE International Conference on Robotics and Automation, Nice, France, May.

Castañon, W. Z. (2017). Control difuso de fuerza para el robot manipulador Mitsubishi PA10-7CE, Master dissertation, Instituto Tecnológico de la Laguna, Torreón, Coahuila, México.

Craig, J. J. (2006). Robotica. Upper Saddle River: Prentice Hall.

Craig, J. J., & Raibert, M. H. (1979). A systematic method of hybrid position/force control of a manipulator. In Proceedings of the IEEE Computer Software and Applications Conference, Chicago, IL, USA.

Khalil, W., Vijayalingam, A., Khomutenko B., Mukhanov I., Lemoine P., & Ecorchard, G. (2014). OpenSYMORO: An open-source software package for Symbolic Modelling of Robots. IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Besancon, France. pp. 1206–1211.

Khatib, O. (1987). 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.

Mason, M. T. (1981). Compliance and force control for computer controlled manipulators. *IEEE Transactions on Systems, Man and Cybernetics, 11*(6), 418–432.

Maza, J. I., & Ollero, A. (2001). HEMERO: Herramienta MATLAB/Simulink para el estudio de manipuladores y robots moviles, Marcombo-Boixareu.

Nguyen, H., Prasad, R., & Walker, C. (2003). *A first course in fuzzy and neural control*. USA: Chapman & Hall/CRC.

Salinas, A. (2011). *Análisis e implementación de esquemas de control de interacción activa para robots manipuladores: Aplicación al robot Mitsubishi PA10, Master dissertation, Instituto Tecnológico de la Laguna, Torreón*. Diciembre: Coah.

Sciavicco, L., & Siciliano B., (1996). Modelling and control of robot manipulators. Berlin: Springer.

Shih-Tin, L., & Ang-Kiong, H. (1998, August). Hierarchical fuzzy force control for industrial robots. *IEEE Transactions on Industrial Electronics, 45*(4).

Shin, K. G., & Lee, C. P. (1985). Compliant control of robotic manipulators with resolved acceleration. In Proceedings of 24th IEEE Conference on Decision and Control, Ft. Lauderdale, FL, USA, December.

Vukobratovic, M., Surdilovic, D., Ekalo, Y., & Katic, D. (2009). Dynamics and robust control of robot-environment interaction. Singapore: World Scientific.

Zadeh, L. A. (1965). Fuzzy sets, *Information and Control, 8,* 338–353.

Zhang, H., & Paul, R. (1985). Hybrid control of robot manipulator. In Proceedings of the IEEE International Conference on Robotics and Automation.