

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



## Predictive Force Control of Robot Manipulators in Nonrigid Environments

L.F. Baptista, J.M.C. Sousa and J.M.G. Sa da Costa

### 1. Introduction

The application of robot manipulators in industry is in general related to tasks such as manipulation or painting that requires only position control of the arm. Nonetheless, there are other robotic tasks like pushing, polishing and grinding that require interaction between the manipulator and a contact surface or environment. This fact leads to the desire of controlling the interaction between the robot and the environment. Although a lot of different control schemes has been proposed in the literature, as surveyed by (Zeng & Hemami, 1997 ; De Schutter et al., 1997), the major force control approaches can be classified as hybrid control (Raibert & Craig, 1981) or impedance control (Hogan, 1985). The hybrid control separates a robotic force task into two subspaces: a force controlled subspace and a position controlled subspace. Two independent controllers are then designed for each subspace. In contrast, impedance control does not attempt to control force explicitly but rather to control the relationship between force and position of the end-effector in contact with the environment. Furthermore, when the environment is rigid with known characteristics it is possible to plan a virtual trajectory, such that a desired force profile is obtained (Singh & Popa, 1995). However, the same does not hold in the presence of nonrigid environments, which disables a reliable application of the classical impedance controller. This problem has motivated the development and design of more sophisticated force control methodologies which usually take into consideration the dynamics of the environment. In (Love & Book, 1995) it is shown that the performance of an impedance controlled manipulator increases when the desired impedance includes some modeling of the environment. Another possible solution to tackle this problem is to use a model-based control scheme like predictive control, which incorporates the manipulator and environment models in a force optimization-based strategy (Wada et al., 1993). Recently, a force control strategy for robotic manipulators in the presence of nonrigid environments combining impedance control and a model predictive control (MPC) algorithm in a force control scheme has been proposed (Baptista et al., 2000b). In this force control methodology, the predictive

controller generate the position and velocity references in the constrained direction, to obtain a desired force profile acting on the environment. The main advantage of this control strategy is to provide an easy inclusion of the environment model in the controller design and thus to improve the global performance of the control system.

Usually, impedance and environmental models are linear, mainly because the solution of an unconstrained optimization procedure can be analytically obtained with moderate computational burden. However, a nonrigid environment has in general a nonlinear behavior, and a nonlinear model for the contact surface must be considered. Therefore, in this paper the linear spring/damper parallel combination, often used as a model of the environment, is replaced by a nonlinear one, where the damping effect depends on the penetration depth (Marhefka & Orin, 1996). Unfortunately, when a nonlinear model of the environment is used, the resulting optimization problem to be solved in the MPC scheme is nonconvex. This problem can be solved using discrete search techniques, such as the branch-and-bound algorithm (Sousa et al., 1997). This discretization, however, introduces a tradeoff between the number of discrete actions and the performance. Moreover, the discrete approximation can introduce oscillations around non-varying references, usually known as the chattering effect, and slow step responses depending on the selected set of discrete solutions. These effects are highly undesirable, especially in force control applications. A possible solution to this problem is a fuzzy scaling machine, which is proposed in this paper. Fuzzy logic has been used in several applications in robotics. In the specific field of robot force control, some relevant references, such as (Liu, 1995 ; Corbet et al., 1996 ; Lin & Huang, 1997), can be mentioned. However, these papers use fuzzy logic in the classic low level form, while in this paper fuzzy logic is applied in a higher level. Here, the fuzzy scaling machine alleviates the effects due to the discretization of the nonconvex optimization problem to be solved in the model predictive algorithm, which derives the virtual reference for the impedance controller considering a nonlinear environment. The fuzzy scaling machine proposed in this paper uses an adaptive set of discrete alternatives, based on the fulfillment of fuzzy criteria applied to force control. This approach has been used in predictive control (Sousa & Setnes, 1999), and is generalized here for model predictive algorithms. The adaptation is performed by a scaling factor multiplied by a set of alternatives. By using this approach, the number of alternatives is kept low, while performance is increased. Hence, the problems introduced by the discretization of the control actions are highly diminished.

For the purpose of performance analysis, the proposed predictive force control strategy with fuzzy scaling is compared with the impedance controller with force tracking by simulation with a two-degree-of-freedom (2-DOF) manipulator, considering a nonlinear model of the environment. The robustness of the predictive control scheme is tested considering unmodeled friction and Corio-

lis effects, as well as geometric and stiffness uncertainties on the contact surface.

The implementation and validation of advanced control algorithms, like the one presented above, require a flexible structure in terms of hardware and software. However, one of the major difficulties in testing advanced force/position control algorithms relies in the lack of available commercial open robot controllers. In fact, industrial robots are equipped with digital controllers having fixed control laws, generally of PID type with no possibility of modifying the control algorithms to improve their performance. Generally, robot controllers are programmed with specific languages with fixed programmed commands having internally defined path planners, trajectory interpolators and filters, among other functions. Moreover, in general those controllers only deal with position and velocity control, which is insufficient when it is necessary to obtain an accurate force/position tracking performance (Baptista et al., 2001b). Considering these difficulties, in the last years several open control architectures for robotic applications have been proposed. Generally, these solutions rely on digital signal processor techniques (Mandal & Payandeh, 1995 ; Jaritz & Spong, 1996) or in expensive VME hardware running under the VxWorks operating system (Kieffer & Yu, 1999). This fact has motivated the development of an open PC-based software kernel for management, supervision and control. The real-time software tool for the experimentation of the algorithms proposed in this paper was developed considering requirements such as low cost, high flexibility and possibility of incorporating new hardware devices and software tools (Baptista, 2000a).

This article is organized as follows. Section 2 summarizes the manipulator and the environment dynamic models. The impedance controller with force tracking is presented in section 3. Section 4 presents the model predictive algorithm with fuzzy scaling applied to force control. The simulation results for a 2-DOF robot manipulator are discussed in section 5. The experimental setup and the force control algorithms implemented in real-time are presented in section 6. The experimental results with a 2-DOF planar robot manipulator are presented in section 7. Finally, some conclusions are drawn in section 8.

## 2. Manipulator and environment modeling

Consider an  $n$ -link rigid-link manipulator constrained by contact with the environment, as shown in fig.1. The complete dynamic model is described by (Siciliano & Villani, 2000)

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + d(\dot{q}) = \tau - \tau_e \quad (1)$$

where  $q, \dot{q}, \ddot{q} \in R^{n \times 1}$  correspond to the joint, position, velocity and acceleration vectors, respectively,  $M(q) \in R^{n \times n}$  is the symmetric positive definite inertia matrix,  $C(q, \dot{q}) \in R^{n \times n}$  is the centripetal and Coriolis matrix,  $g(q) \in R^{n \times 1}$  contains the gravitational terms and  $d(\dot{q}) \in R^{n \times 1}$  accounts for the frictional terms. The vector  $\tau \in R^{n \times 1}$  is the joint input torque vector and  $\tau_e \in R^{n \times 1}$  denote the generalized vector of joint torques exerted by the environment on the end-effector. From (1) it is possible to derive the robot dynamic model in the Cartesian space:

$$M_x(x)\ddot{x} + C_x(x, \dot{x})\dot{x} + g_x(x) + d_x(\dot{x}) = f - f_e \quad (2)$$

where  $x$  is the  $n$ -dimensional vector of the position and orientation of the manipulator's end-effector,  $f = J^{-T}(q)\tau \in R^{n \times 1}$  is the robot's driving force,  $f_e \in R^{n \times 1}$  is the contact force vector and  $J$  represents the Jacobian matrix.

The interaction force vector  $f_e = [f_n \quad f_t]^T$  is composed by the normal contact force  $f_n$  and the tangential contact forces  $f_t$  caused by friction contact between the end-effector and the surface. An accurate modeling of the contact between the manipulator and the environment is usually difficult to obtain due to the complexity of the robot's end-effector interaction with the environment. In this paper, the normal contact force  $f_n$  is modeled as a nonlinear spring-damper mechanical system according (Marhefka & Orin, 1996):

$$f_n = k_e \delta x + \rho_e (\delta x) \dot{x} \quad (3)$$

where the terms  $k_e$  and  $\rho_e$  are the environment stiffness and damping coefficients, respectively,  $\delta x = x - x_e$  is the penetration depth, where  $x_e$  stands for the distance between the surface and the base Cartesian frame. Notice that the damping effect depends non-linearly on the penetration depth  $\delta x$ . The tangential contact force vector  $f_t$  due to surface friction is assumed to be given as proposed by (Yao & Tomizuka, 1995):

$$f_t = \mu |f_n| \text{sgn}(\dot{x}_p) \quad (4)$$

where  $\dot{x}_p$  is the unconstrained or sliding velocity and  $\mu$  is the dry friction coefficient between the end-effector and the contact surface.

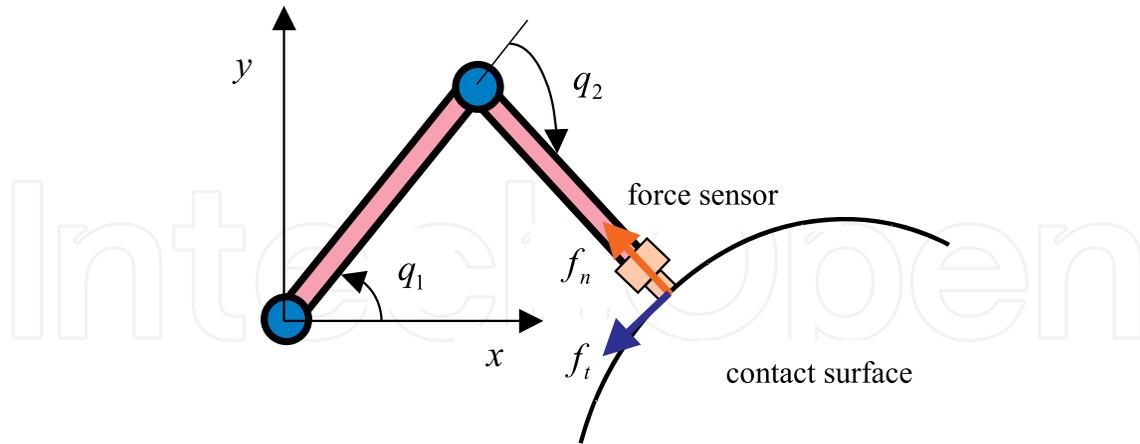


Figure 1. Robot manipulator applying a desired force on the environment. (Reprinted from Baptista, L.; Sousa, J. & Sá da Costa, J. (2001a) with kind permission of Springer Science and Business Media).

### 3. Impedance control

The impedance controller proposed by (Hogan, 1985) aims at controlling the dynamic relation between the manipulator and the environment. The force exerted by the manipulator on the environment depends on the end-effector position and the correspondent impedance. The impedance of the robot is divided in the following terms: one that is physically intrinsic to the manipulator and the other that is given to the robot by the controller. The impedance control goal is to oblige the manipulator to follow the reference or target impedance. As shown by (Volpe & Khosla, 1995) a good impedance relation is achieved with a linear model of second order. The complete form of a second order type impedance control model, which is valid for free or constrained motion, is given by:

$$M_d \ddot{x} - B_d (\dot{x}_d - \dot{x}) - K_d (x_d - x) = -f_e \tag{5}$$

where  $\dot{x}_d, x_d$  are the desired velocity and position defined in the Cartesian space, respectively, and  $\dot{x}, x$  are the end-effector velocity and position in Cartesian space, respectively. The matrices  $M_d, B_d, K_d$  are the desired inertia, damping and stiffness for the manipulator. The reference or target end-effector acceleration  $u \equiv \ddot{x}$  is then given by:

$$u = M_d^{-1} (B_d \dot{e} + K_d e - f_e) \tag{6}$$

where  $\dot{e} = \dot{x}_d - \dot{x}$ ,  $e = x_d - x$  are the velocity and position errors, respectively. Thus,  $u$  can be used as the command signal to an inner position control loop in order to drive the robot accordingly to the desired trajectory.

### 3.1 Virtual trajectory for force tracking

The major drawback of the impedance control scheme presented above is related to its poor force tracking capability, especially in the presence of nonrigid environments (Baptista et al., 2000b). However, from the conventional impedance control scheme it is possible to obtain a force control scheme in a steady-state contact condition with the surface. Considering the impedance control scheme (6) in the constrained direction, the following holds:

$$u_f = m_d^{-1}(b_d(\dot{x}_v - \dot{x}) + k_d(x_v - x) - f_n) \quad (7)$$

where  $x_v$ ,  $\dot{x}_v$  and  $u_f$  are the virtual position, velocity and target acceleration, respectively, while  $m_d, b_d, k_d$  are appropriate elements of  $M_d, B_d, K_d$  matrices defined in (5) in the constrained direction. The contact force  $f_n$  during steady-state contact with the surface is given by:

$$f_n = k_d(x_v - x) \quad (8)$$

Considering for simplicity the environment modeled by a linear spring with stiffness  $k_e$  the contact force is given by:

$$f_n = k_e \delta x \quad (9)$$

This leads to the following steady-state position and contact force (Singh & Popa, 1995):

$$x_{ss} = \frac{k_d x_v + k_e x_e}{k_d + k_e} \quad (10)$$

$$f_{n_{ss}} = \frac{k_d k_e}{k_d + k_e} (x_v - x_e) \quad (11)$$

It is possible to apply a desired force  $f_d$  on the system while simultaneously achieving the desired impedance by estimating the desired virtual position  $x_v$  as:

$$x_v = x_e + f_d \left( \frac{k_e + k_d}{k_e k_d} \right) \tag{12}$$

Moreover, when the environment stiffness is unknown, it is also possible to obtain the virtual position from  $f_d, f_n$  and  $\delta x$  (Jung & Hsia, 1995). In this case, by substitution of  $k_e$  in (12) the following virtual position  $x_v$  is obtained:

$$x_v = \begin{cases} x_e + \frac{f_d}{k_d} & \text{if } f_n = 0 \\ x_e + f_d \left( \frac{k_d \delta x + f_n}{k_d f_n} \right) & \text{if } f_n \neq 0 \end{cases} \tag{13}$$

which is valid for contact and non-contact condition. This approach enables the classical impedance controller, given by (6), with force tracking capability without explicit knowledge of the environment stiffness. Notice that  $\dot{x}_v$  is usually assumed to be zero due to the noise always present in the force sensor measurements.

### 3.2 Impedance control with force tracking

The impedance control with force tracking (ICFT) block diagram is presented in fig.2.

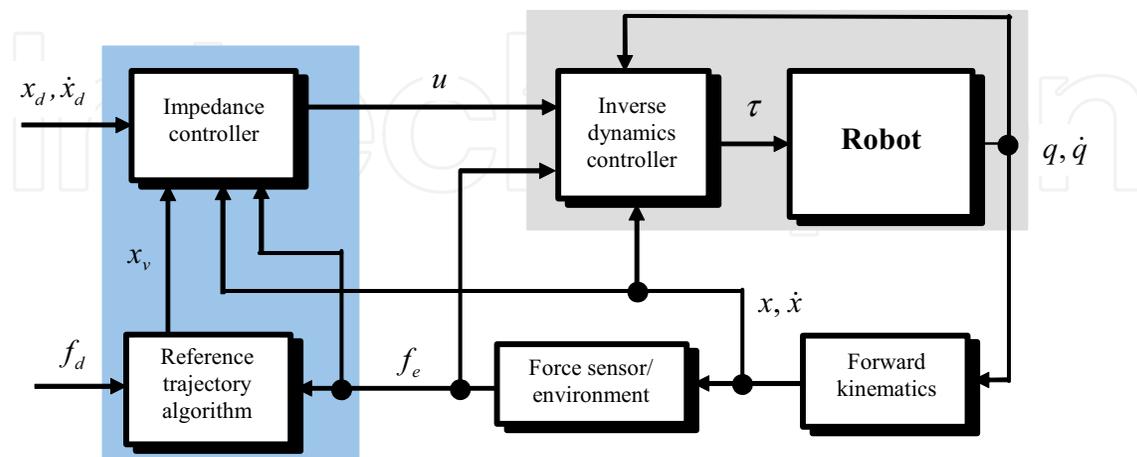


Figure 2. Impedance control with force tracking (ICFT) block diagram.

In this scheme, the virtual position  $x_v$  given by (13), is computed in the *Reference trajectory algorithm* block, while the target acceleration vector

$u = [u_f \quad u_p]^T$  with  $u_p$  is obtained from (6) and  $u_f$  from (7), is computed in the *Impedance controller* block. Moreover, the unconstrained target acceleration vector  $u_p$  is further compensated by a proportional-derivative (PD) controller, which is given by:

$$u_{pc} = u_p + K_P e + K_D \dot{e} \quad (14)$$

where  $K_P$  and  $K_D$  are proportional and derivative gain matrices, respectively. The target acceleration vector  $u_c = [u_f \quad u_{pc}]^T$  is then used as the driving signal to the inverse dynamics controller, in order to track the desired force profile. Since robot controllers are usually implemented in the joint space, it is useful to obtain the correspondent target joint acceleration  $u_q$  for the inverse dynamics controller.

Then, using the appropriate kinematics transformations,  $u_q$  is given by:

$$u_q = J^{-1}(u_c - \dot{J}(q)\dot{q}) \quad (15)$$

Then, applying an inverse dynamics controller in the inner control loop, the joint torques are given by:

$$\tau = \hat{M}(q)u_q + \hat{g}(q) + J(q)^T f_e \quad (16)$$

where  $\hat{M}(q)$ ,  $\hat{g}(q)$  are estimates of  $M(q)$ ,  $g(q)$  in the robot dynamic model (1). Notice that Coriolis and friction effects are neglected. The impedance controller with force tracking (ICFT) presented above is a good control approach for rigid environments since the end-effector velocity in the constrained direction is close to zero, which leads to a virtual position with an acceptable precision. However, for nonrigid environments the constrained velocity can hardly be zero, which limits the accuracy of the control system to track the desired force profile (Baptista et al, 2001a). To overcome the drawbacks of the scheme presented above, this paper proposes an alternative force control methodology based on a model predictive algorithm (MPA) which is presented in the next section.

#### 4. Model predictive algorithms applied to force control

Predictive algorithms consist of a broad range of methods, which are used to predict a desired variable in an optimal way. The most common predictive algorithms are model predictive controllers (Maciejowski, 2002), which have one common feature; the controller is based on the prediction of the future system behavior by using a process model. In a more general way, predictive algorithms are based on the following basic concepts:

1. Use of a (nonlinear) model to predict the process outputs at future time periods over a prediction horizon;
2. Computation of a sequence of future inputs using the model of the system by minimizing a certain objective function;
3. Receding horizon principle; at each sampling period the optimization process is repeated with new measurements, and only the first input obtained is applied to the system.

In this paper, an MPA is used to predict the target position  $x_v$  to the impedance control law in (7), such that a desired force profile is obtained. In general, a predictive algorithm minimizes a cost function over a specified prediction horizon  $H_p$ . In order to reduce model-plant mismatch and disturbances in an effective way, the predictive algorithm is combined with an internal model control (IMC) structure (Economou et al., 1986) which increases the force tracking performance. A filter is included in the feedback loop of the predictive structure to reduce the noise present in the force sensor data. This filter stabilizes the loop by decreasing the gain, increasing the robustness of the force control loop. The sequence of future target positions  $x_v(k) \dots x_v(k + H_p - 1)$  over a specified prediction horizon, produced by the MPA, results in a new target acceleration by the impedance control law (6), which determines the force to apply on the surface. Predictive algorithms need a prediction model to compute the optimal input. In this paper, the model must predict the contact force  $f_m$  based on the measured position  $x$  and velocity  $\dot{x}$ . This model must consider the dynamics of the environment given by (3). In order to minimize the number of calculations during the nonlinear optimization procedure, only the virtual trajectory is computed in an optimal way, and thus  $\dot{x}_v$  is assumed to be zero. Therefore, the nonlinear prediction model in the constrained direction is given by:

$$m_d u_f + b_d \dot{x} - k_d (x_v - x) = -f_m \quad (17)$$

Note that a discrete version of this model is required, predicting the future values  $f_m(k+i)$  based on the measured position  $x(k)$  and the measured velocity

$\dot{x}(k)$  at time instant  $k$ . The predictive scheme is combined with an internal model control scheme, and the model-plant mismatch is given by

$$e_m(k) = f_n(k) - f_m(k) \quad (18)$$

The desired force profile  $f_d$  is compensated by the filtered modeling error  $e_{mf}$ , as shown in fig.4, resulting in the modified force reference  $f_{dc}$  defined as:

$$f_{dc}(k) = f_d(k) - e_{mf}(k) \quad (19)$$

The cost function considered for the force control scheme is then given by:

$$J(x_v) = \sum_{i=1}^{H_p} (f_{dc}(k+i) - f_m(k+i))^2 \quad (20)$$

The process inputs and outputs, as well as state variables, can be subjected to constraints, which must be incorporated in the optimization problem.

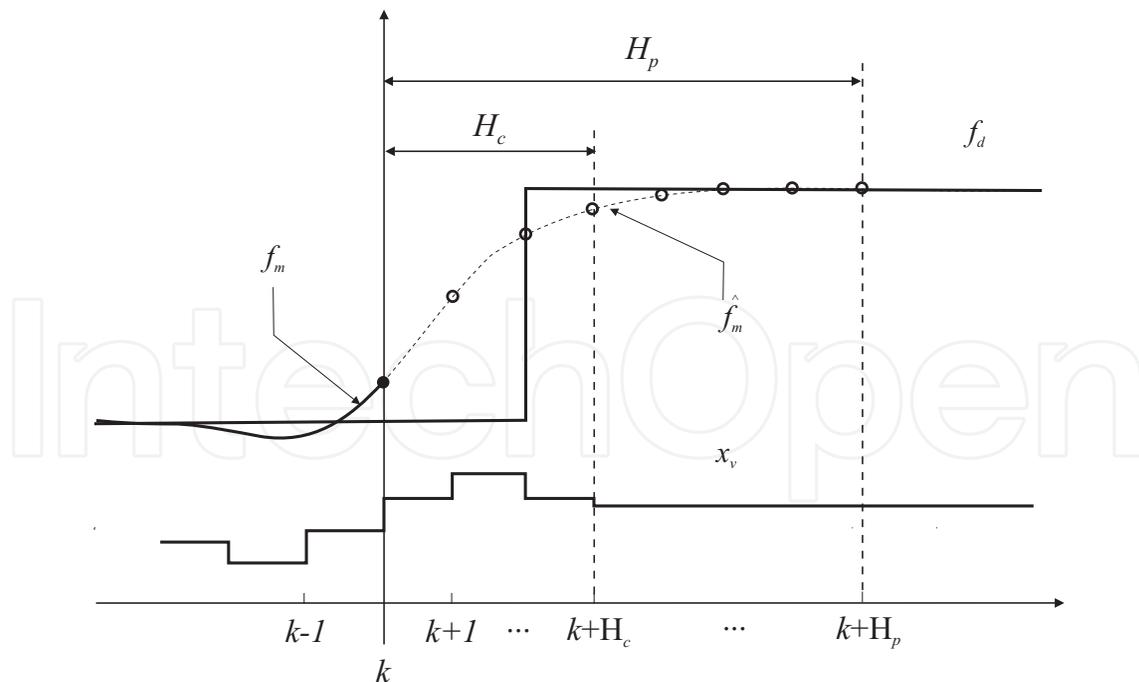


Figure 3. Basic principle of MPA applied to robot force control.

The performance of the MPA depends largely on the accuracy of the process model. The model must be able to accurately predict the future process out-

puts, and at the same time be computationally attractive to meet real-time demands. When both nonlinear models and constraints are present, the optimization problem is nonconvex. Efficient optimization methods for nonconvex optimization problems can be used when the solution space is discretized, and techniques such as Branch-and-Bound - B&B (Sousa et al., 1997) can be applied. The B&B method can be used in a recursive way, demanding less computation effort than other methods, and is used in this paper to solve the nonconvex optimization problem. Figure 3 presents the basic principle of a predictive strategy applied to robot force control.

#### 4.1 Branch-and Bound Optimization

Branch-and-Bound algorithms solve optimization problems by partitioning the solution space. In this paper, B&B is used for the optimization problem that must be solved at each time instant  $k$  in the model predictive algorithm. A B&B algorithm can be characterized by two rules: *Branching rule* - defines how to divide a problem into sub-problems; and *Bounding rule* - establishes lower and upper bounds in the optimal solution of a sub-problem, allowing for the elimination of sub-problems that do not contain an optimal solution.

The model predicts the future outputs of the system, which are the forces  $f_m(k+1), \dots, f_m(k+H_p)$  and can be given by (3) when the stiffness coefficient is considered to be constant. Let  $M$  be the possible discrete inputs of the system, which are denoted as  $w_j$ . Thus, at each step the desired positions  $x_v(k+i-1) \in \Omega$ , are given by  $\Omega = \{\omega_j | j=1, 2, \dots, M\}$ .

In the considered predictive scheme, the problem to be solved is represented by the objective function (20) minimizing the predicted force error. This optimization problem is successively decomposed by the branching rule into smaller sub-problems. At time instant  $k+i$  the cumulative cost of a certain path followed so far, and leading to the output  $f_m(k+i)$  is given by

$$J^{(i)} = \sum_{l=1}^i (f_{dc}(k+l) - f_m(k+l))^2 \quad (21)$$

where  $i = 1, \dots, H_p$ , denotes the level corresponding to the time step  $k+i$ . A particular branch  $j$  at level  $i$  is created when the cumulative cost  $J^{(i)}(u)$  plus a *lower bound* on the cost from the level  $i$  to the terminal level  $H_p$  for the branch  $j$ , denoted  $J_{L_j}$ , is lower than an *upper bound* of the total cost, denoted  $J_U$ :

$$J^{(i)} + J_{L_j} < J_U \quad (22)$$

Let the total number of branches verifying this rule at level  $i$  be given by  $N$ . In order to increase the efficiency of the B&B method, it is required that this number should be as low as possible, i.e.  $N \ll M$ .

The major advantages of the B&B algorithm applied to MPA over other non-convex optimization methods are the following: the global discrete minimum containing the optimal solution is always found, guaranteeing good performance; and the B&B method implicitly deals with constraints. In fact, the presence of constraints improve the efficiency of bounding, restricting the search space by eliminating non-feasible sub-problems.

The most serious drawbacks of B&B are the exponential increase of the computational time with the prediction horizon and the number of alternatives, and the discretization of the possible inputs, which are the position references  $x_v$  in this paper. A solution to these problems is proposed in the next section.

## 4.2 Fuzzy scaling machine

Fuzzy predictive filters, as proposed in (Sousa & Setnes, 1999), select discrete control actions by using an adaptive set of control alternatives multiplied by a gain factor. This approach diminishes the problems introduced by the discretization of control actions in MPA. The predictive rules consider an error in order to infer a scaling factor, or gain,  $\gamma(k) \in [0,1]$  for the discrete incremental inputs. For the robotic application considered in this paper this error is given by  $e_m$ , as defined in (18). The gain  $\gamma(k)$  goes to the zero value when the system tends to a steady-state situation, i.e., the force error and the change in this error are both small. On the other hand, the gain increases when the force error or the change in this error is high. When the gain  $\gamma(k)$  is small, the possible inputs are made close to each other, diminishing to a great extent, or even nullifying, oscillations of the output. When the system needs to change rapidly the gain is increased and the interval of variation of the inputs is much larger, allowing for a fast response of the system. The fuzzy scaling machine reduces thus the main problem introduced by the discretization of the inputs, i.e. a possible limit cycle due to the discrete inputs, maintaining also the number of necessary input alternatives low, which increases significantly the speed of the optimization algorithm. The design of the fuzzy scaling machine consists of three parts: the choice of the discrete inputs, the construction of the fuzzy rules for the gain filter, and the application of the B&B optimization. The first two parts are explained in the following.

Let the virtual position  $x_v(k-1) \in X$ , which was described in (17), represent the input reference at time instant  $k-1$ , where  $X = [X^-, X^+]$  is the domain of this reference position. Upper and lower bounds must be defined for the possible changes in this reference signal at time  $k$ , which are respectively  $x_k^+$  and

$$x_k^- : x_k^+ = X^+ - x_v(k-1), x_k^- = X^- - x_v(k-1).$$

These values are then defined as the maximum changes allowed for the virtual reference when it is increased or decreased, respectively. The adaptive set of incremental input alternatives can now be defined as

$$\Omega_k^* = \{0, \lambda_l x_k^+, \lambda_l x_k^- \mid l = 1, \dots, N\} \tag{23}$$

The distribution  $\lambda_l$  must be chosen taking into account that  $0 \leq \lambda_l \leq 1$ . In this way, the choice of  $\lambda_l$  sets the maximum change allowed at each time instant by scaling the maximum variations  $x_k^+$  and  $x_k^-$ . The parameter  $l$  is important to define the number of possible inputs. From (23) it follows that the cardinality of  $\Omega_k^*$ , i.e., the number of discrete alternatives, is given by  $M = 2l + 1$ .

The fuzzy scaling machine applies a scaling factor,  $\gamma(k) \in [0, 1]$  to the adaptive set of inputs  $\Omega_k^*$  in order to obtain the scaled inputs  $\Omega_k$  of the optimization routine, the B&B in this case:

$$\Omega_k = \gamma(k) \cdot \Omega_k^* \tag{24}$$

The scaling factor  $\gamma(k)$  must be chosen based on the predicted error between the reference and the system's output, which is defined as

$$e(k + H_p) = f_{dc}(k + H_p) - f_n(k + H_p), \tag{25}$$

where  $f_{dc}(k + H_p)$  is the reference to be followed at time  $H_p$ , as in (19). Added to the error, the change in the error gives usually important indications on the evolution of the system behavior. This information can also be considered in the derivation of  $\gamma(k)$ . The change in error is given by

$$\Delta e(k) = e(k) - e(k-1). \tag{26}$$

The fuzzy rules to be constructed have as antecedents the predicted error and the change in the error, and as consequent a value for the scaling factor. Simple heuristic rules can be constructed noticing the following. The system is close to a steady-state situation when the error and the change in the error are both small. In this situation, the discrete virtual references must be scaled down, allowing smaller changes in the reference  $x_v$ , which yield smaller variations in the impedance controller, and  $\gamma(k)$  should tend to zero. On the other hand, when the predicted error or the change in error are high, larger discrete refer-

ences must be considered, and  $\gamma(k)$  should tend to its maximum value, i.e. 1. The trapezoidal and triangular membership functions  $\mu_e(e(k+H_p))$  and  $\mu_{\Delta e}(\Delta e(k))$  define the two following fuzzy criteria: “small predicted error” and “small change in error”, respectively. The two criteria are aggregated using a fuzzy intersection; the minimum operator (Klir, 1995). In this way, the membership degree of these criteria using the min operator is given by:

$$\mu_\gamma(e(k+H_p), \Delta e(k)) = \min(\mu_e, \mu_{\Delta e}), \quad (27)$$

The scaling factor  $\gamma(k)$  must be the fuzzy complement of a certain membership degree  $\mu_\gamma$ :

$$\gamma(k) = \bar{\mu}_\gamma = 1 - \mu_\gamma. \quad (28)$$

Summarizing, the set of inputs  $\Omega_k^*$  at time instant  $k$ , which are virtual references in this paper, is defined in (23). These inputs are within the available input space at time  $k$ . Further, the inputs are scaled by the factor  $\gamma(k) \in [0,1]$  to create a set of adaptive alternatives  $\Omega_k$ , which are passed on to the optimization routine. At a certain time  $k$ , the value of  $\gamma(k)$  is determined by simple fuzzy criteria, regarding the predicted error of the system. Note that the proposed fuzzy scaling machine has only the following design parameters:  $\lambda_\gamma$ , and the membership functions  $\mu_e$  and  $\mu_{\Delta e}$ . Moreover, the tuning of these parameters is not a hard task, allowing the use of some heuristics to derive them. Possible constraints on the input signal, which is the virtual trajectory in this paper, are implemented by selecting properly the parameters  $\lambda_\gamma$ .

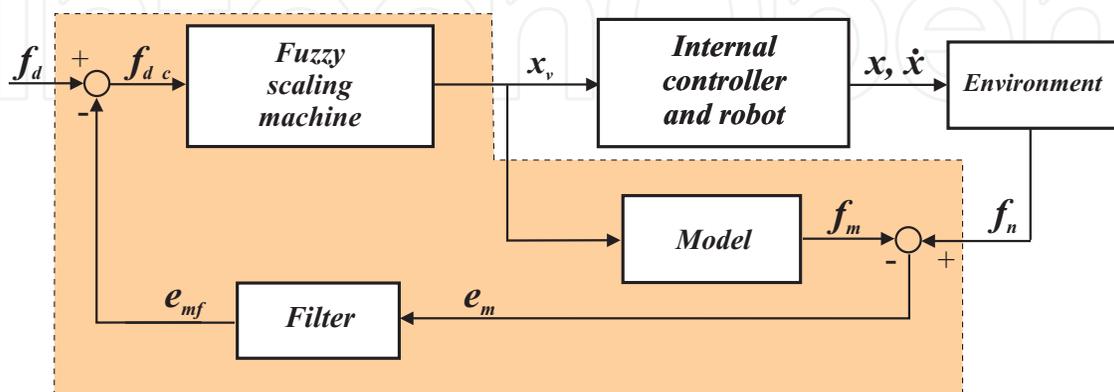


Figure 4. Block diagram of proposed predictive force control algorithm with fuzzy scaling machine.

(Reprinted from Baptista, L.; Sousa, J. & Sá da Costa, J. (2001a) with kind permission of Springer Science and Business Media).

Figure 4 depicts the proposed predictive force control algorithm with fuzzy scaling. The block *Fuzzy scaling machine* contains the model predictive algorithm, the B&B optimization and the fuzzy scaling strategy. The block *Internal controller and robot* implement the impedance and the inverse dynamics control algorithms. The robot dynamic model equations are also computed in this block. The block *Environment* contains the nonlinear model of the environment. In order to cope with disturbances and model-plant mismatches, an internal model controller is included in the control scheme. The block *Filter* belongs to the IMC structure (Baptista et al., 2001a).

## 5. Simulation results

The force control scheme introduced in this paper is applied to a robot through computer simulation for an end-effector force/position task in the presence of robot model uncertainties and inaccuracy in the environment location and the correspondent stiffness characteristics. The robot model represents the links 2 and 3 of the PUMA 560 robot. In all the simulations, a constant time step of 1 ms is used. The overall force control scheme including the dynamic model of the PUMA robot is simulated in the Matlab/Simulink environment. A nonrigid friction contact surface is placed in the vertical plane of the robot workspace where it is assumed that the end-effector always maintain contact with the surface during the complete task execution.

In order to analyze the force control scheme robustness to environment modeling uncertainties, a non rigid time-varying stiffness profile  $k_e(t)$  is considered, given by:

$$k_e(t) = \begin{cases} 1000 + \sin(\pi t / 2) & 0 < t < 2 \\ 1000e^{(-0.25(t-2))} & 2 \leq t < 3 \end{cases} \quad (29)$$

The damping coefficient and the coefficient of dry friction are settled to  $\rho_e=45$  Ns/m<sup>2</sup> and  $\mu=0.2$ , respectively. Notice that the stiffness coefficient is considered to be constant ( $k_e=1000$  N/m) in the environment model used for predict the contact force  $f_m$ . The matrices in the impedance model (6) are defined as  $M_d = \text{diag}[2.5 \ 2.5]$  and  $K_d = \text{diag}[250 \ 2500]$  to obtain an accurate force tracking in the  $x$ -axis direction and an accurate position tracking performance in the  $y$ -axis direction.

The matrix  $B_d$  is computed to obtain a critically damped system behavior. The control scheme was tested considering a smooth step force profile of 10 N and a desired position trajectory from  $p_1 = [0.5 \ -0.2]$  m to  $p_2 = [0.5 \ 0.6]$  m.

Uncertainties in the location of the contact surface given by the final real position of  $p_{2r}=[0.512 \ 0.6]$  m are considered in the simulations, as shown in fig.5.

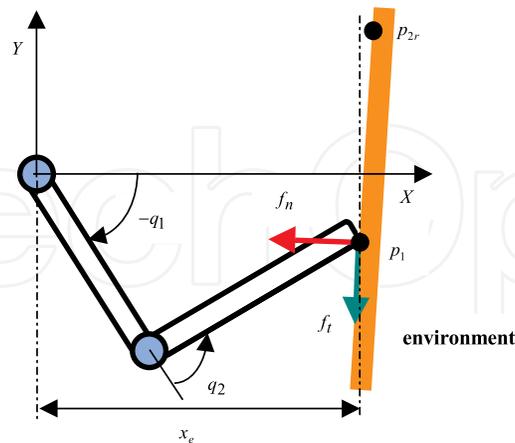


Figure 5. 2-DOF planar robot in contact with the environment.

(Reprinted from Baptista, L.; Sousa, J. & Sá da Costa, J. (2001a) with kind permission of Springer Science and Business Media).

The parameters of the predictive controller are  $H_p = H_c = 2$  and the fuzzy scaling machine is applied only during the constant path of the reference force trajectory. This means that during the reference force transition periods, the fuzzy scaling inference is switched off. The discrete alternatives  $\Delta x_v$  for the fuzzy scaling machine are given by:

$$\Omega_k^* = [-0.050 \quad 0 \quad 0.050] \quad (30)$$

In the inner loop controller (16), only the elements of the inertia matrix and the gravitational terms with parameters 20% smaller than their exact values are considered. The Coriolis and friction terms were neglected in the implementation of the algorithm but considered in the simulation of the robot dynamic model. The proportional and derivative gains in (14) are settled to  $K_P = \text{diag}[5000 \ 5000]$  and  $K_D = \text{diag}[500 \ 500]$ .

Simulations using the impedance controller with force tracking (ICFT) and the control algorithm proposed in this paper are compared. The conventional impedance controller uses the reference trajectory algorithm presented in (13) considering the environment modeled as a linear spring with  $k_e=1000$  N/m. The simulation results obtained with the ICFT are presented in fig.6, which exhibit poor force tracking performance with relatively large force tracking errors. However, the ICFT follows the desired position trajectory with high accuracy; in fact, it is not possible to distinguish the reference from the actual  $y$ -axis position in fig.6.

The force control scheme uses the model predictive algorithm to compute the virtual trajectory  $x_v$ , the fuzzy scaling machine and the nonlinear environment model, which furnish the normal force described by (3). The force and position

results from the application of this controller are presented in fig.7. Comparing this figure with fig.6, it becomes clear that the proposed force controller presents a remarkable performance improvement in terms of force tracking capability. In fact, it is not possible to distinguish the reference force from the actual contact force. In terms of position control, similar performance is achieved. The results for both controllers can be compared in Table 1, where the error norm  $\| \cdot \|$  for position and force errors, as well as the absolute maximum values for these errors are presented. The table shows that the force control performance is clearly superior for the MPA with fuzzy scaling machine.

Force control algorithms	$\ e_p\ $ [m]	Max( $e_p$ ) [mm]	$\ e_f\ $ [N]	Max( $e_f$ ) [N]
Impedance control with force tracking	0.041	0.836	60.426	4.420
MPA with fuzzy scaling machine	0.041	0.801	0.8021	0.064

Table 1. Euclidian norm of position, force errors and absolute maximum errors.

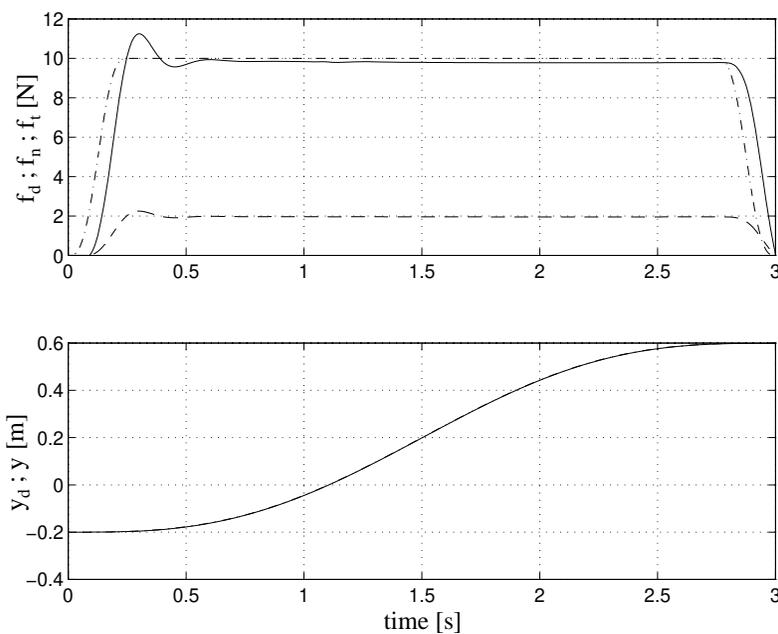


Figure 6. Impedance control with force tracking: desired force (dashdot), normal force (solid) and friction force (dashed) – top view; desired y-axis trajectory (dashdot) and actual position trajectory (solid) – bottom view.

(Reprinted from Baptista, L.; Sousa, J. & Sá da Costa, J. (2001a) with kind permission of Springer Science and Business Media).

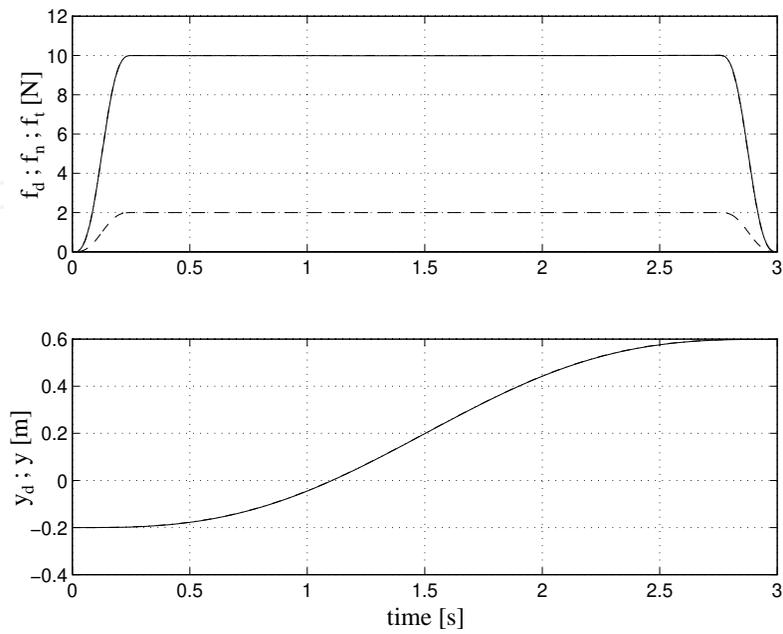


Figure 7. MPA with fuzzy scaling: desired force (dashdot), normal force (solid) and friction force (dashed) – top view; desired  $y$ -axis trajectory (dashdot) and actual position trajectory (solid) – bottom view.

(Reprinted from Baptista, L.; Sousa, J. & Sá da Costa, J. (2001a) with kind permission of Springer Science and Business Media).

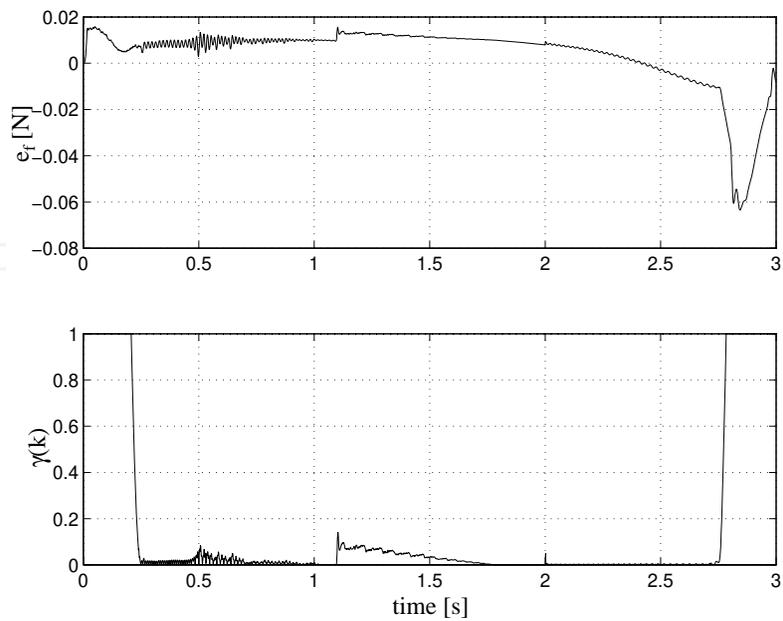


Figure 8. MPA with fuzzy scaling: contact force errors (top view) and fuzzy scaling factor  $\gamma(k)$  (bottom view).

(Reprinted from Baptista, L.; Sousa, J. & Sá da Costa, J. (2001a) with kind permission of Springer Science and Business Media).

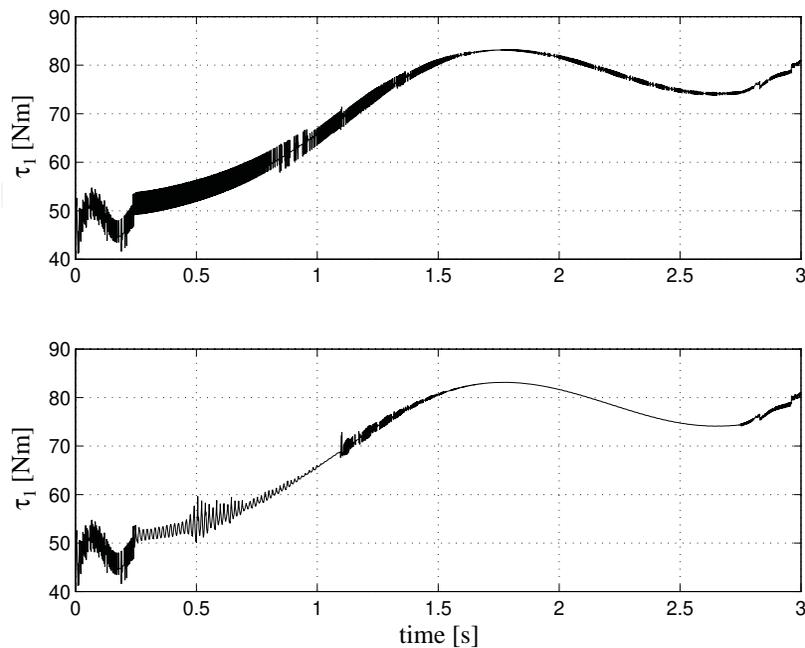


Figure 9. MPA with fuzzy scaling: joint torque  $\tau_1$  without (top view) and with fuzzy scaling machine (bottom view).

(Reprinted from Baptista, L.; Sousa, J. & Sá da Costa, J. (2001a) with kind permission of Springer Science and Business Media).

In order to study in more detail the proposed force approach, fig. 8 presents the contact force error and the fuzzy scaling factor  $\gamma(k)$  for the same trajectory. The factor  $\gamma(k)$  exhibits a fast convergence to values around zero during the constant reference force path, which reduces the chattering present on the target trajectory and in the joint torques.

The joint torque  $\tau_1$  using the predictive approach with and without the fuzzy scaling machine is shown in fig. 9. The strategy without fuzzy scaling produce undesirable oscillations on the virtual trajectory  $x_v$ , which has the same effect on the joint torques  $\tau$ . This effect is significantly reduced by the fuzzy scaling machine, as shown in fig.9 for the joint torque  $\tau_1$ .

## 6. Experimental setup

In order to validate the proposed force control scheme presented in previous sections, real-time experiments were carried out with a 2-DOF planar manipulator and a nonrigid mechanical environment built at the Robotics Laboratory of Technical University of Lisbon/Instituto Superior Técnico (IST). A low cost open PC-based control architecture using a commercial servo-axis interface board was developed to control the robotic setup in an effective and reliable way.

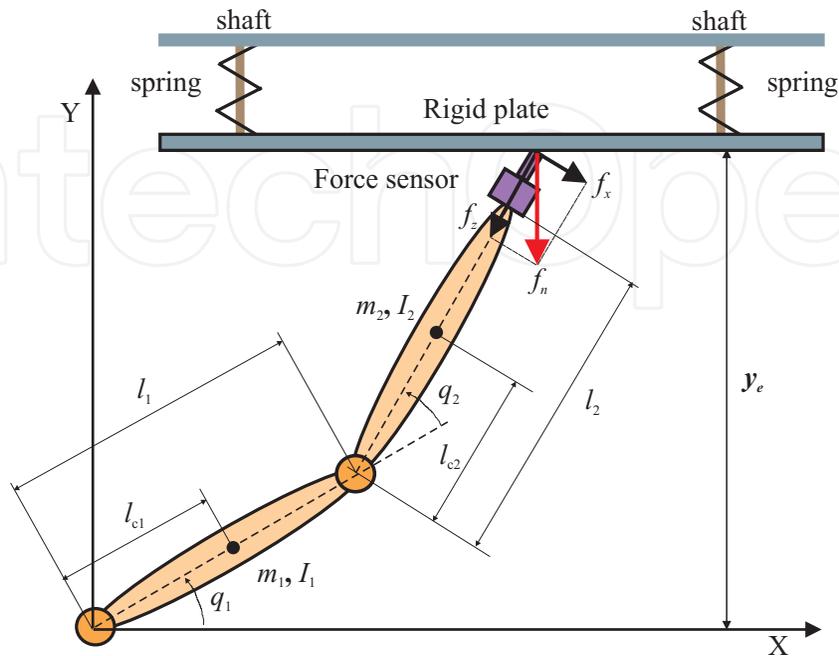


Figure 10. Top view schematic representation of the robotic setup. (Note: The arrows represent the  $f_x$ ,  $f_z$  components and indicate the negative directions of the contact forces measured by the force sensor.).

(Reprinted from Baptista, L.; Sousa, J. & Sá da Costa, J. (2001b) with permission of Elsevier).

The planar robot has two revolute joints driven by HARMONIC DRIVE actuators (HDSH-14). Each actuator has an electric d.c. motor, a harmonic drive, an encoder and a tachogenerator. The robot links parameters are given in Table 2 where  $l_{ci}$  is the distance to the mass center of the link  $i$ ,  $I_{zzi}$  is the inertia moment related to the  $z$ -axis and  $I_{mi}$  is the actuator's inertia moment related to the output shaft. The contact surface used for force control experiments is based on a steel plate with two springs in parallel guided by shafts with ball bearings (Baptista, 2000a). The top view of the planar robot and the nonrigid mechanical contact surface are shown in fig.10 and a picture of the robot and mechanical environment is depicted in fig.11.

Link	$l_i$	$m_i$	$l_{ci}$	$I_{zzi}$	$I_{mi}$
$I$	[m]	[Kg]	[m]	[Kg.m <sup>2</sup> ]	[Kg.m <sup>2</sup> ]
1	0.320	5.00	0.163	0.120	0.081
2	0.330	1.47	0.137	0.018	0.021

Table 2. Parameters of the planar 2-DOF robot. (Note: link length  $l_2$  includes the force sensor length and the end-effector device length.)



Figure 11. Top view of the 2-DOF robot and the nonrigid mechanical environment. (Reprinted from Baptista, L.; Sousa, J. & Sá da Costa, J. (2001b) with permission of Elsevier).

### 6.1 Control hardware and software tools

The control hardware is based on a PC Pentium 200 MHz, a servo-axis I/O card, linear power amplifiers and a 6-axis force/torque sensor. The power amplifiers are configured to operate as current amplifiers. In this functioning mode, the input control signal is a voltage in the range  $\pm 10$  V with current ratings in the interval  $[-2; 2]$  A. The signals are processed through a low cost ISA-bus servo I/O card from ServoToGo, Inc. The contact forces are measured by a JR<sup>3</sup> 6-axis force/torque sensor mounted at the end-effector of the arm. The force and torque signals are computed by a ISA-bus DSP card with a frequency rate of 8 KHz. Note that in order to reduce the surface friction during motion, a ball bearing is mounted at the end-effector contact device. The simplified control hardware architecture is shown in fig.12.

The real-time software tool used to implement the force control experiments was developed in the C++ language for the MS-DOS operating system. Real-time performance is guaranteed by reprogramming the Programmable Interval Timer (PIT) of the computer system (Baptista, 2000a). The periodic interrupt generated by the servo-axis I/O board PIT is configured to the Interrupt Request 5 (IRQ 5), which activates the Interrupt Service Routine 13 (ISR 13). This ISR is reserved for the control algorithm implementation at a sampling frequency  $F_s=1$  KHz. The software uses also the ISR 8 which is activated by the computer's IRQ 0.

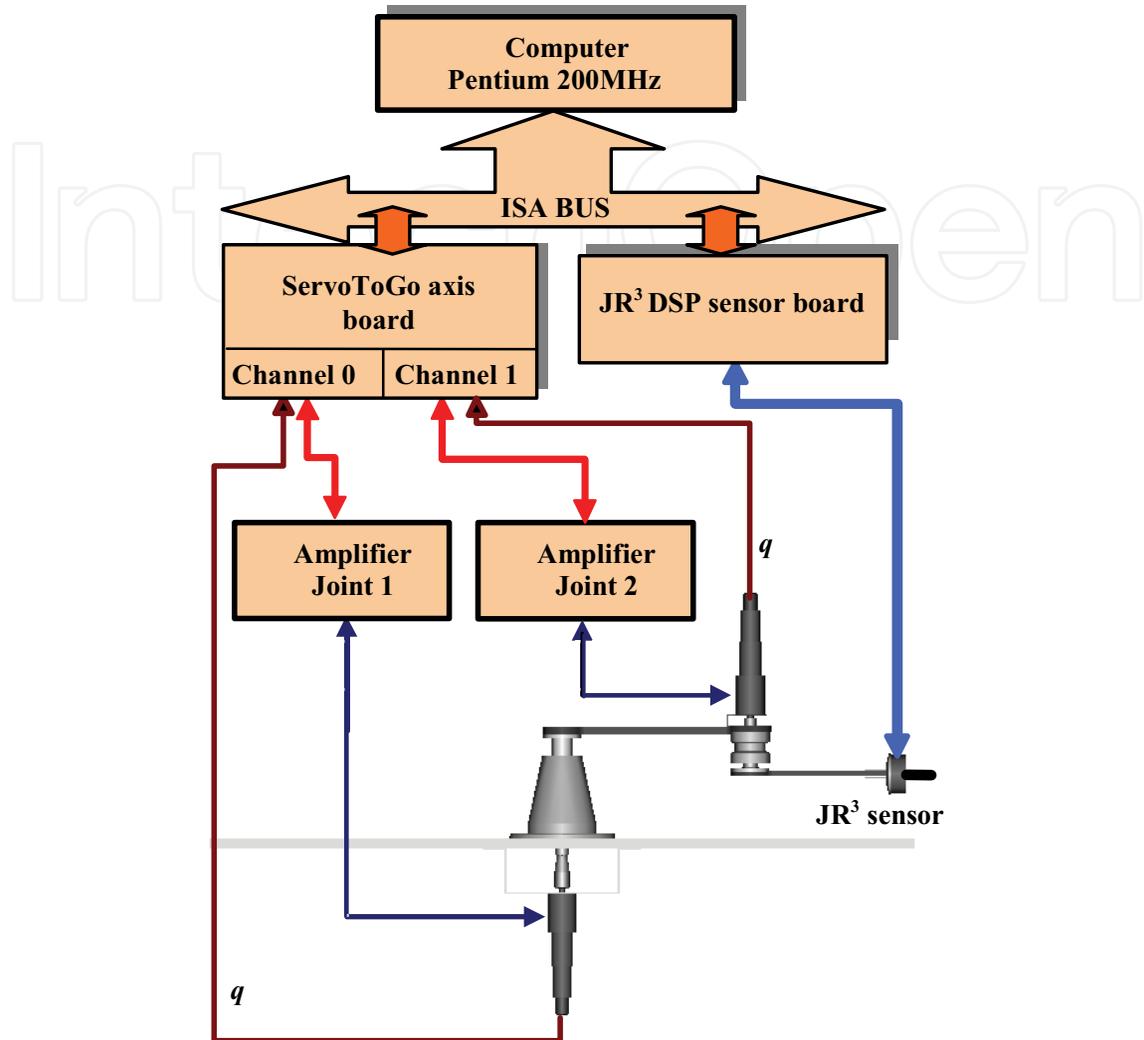


Figure 12. Simplified hardware control architecture of the 2-DOF robot.

This ISR is redefined to execute the safety routine at a frequency that is the double of the sampling frequency. In order to avoid the accumulation of interrupt requests, the ISR 13 increments a flag at the beginning of the execution of the selected control routine. This flag is decremented at the end of the execution of the procedure and is checked by the safety routine, which aborts the real-time procedure if its content is superior to the unit value. In this situation, the failure is reported to the operator by the interface program.

## 6.2 Real-time force control algorithms

For real-time implementation of the force control algorithms presented in previous sections, some simplifications were considered due to the higher number of calculations that must be performed in real-time. Let's first consider the

classical model-based impedance controller, derived from (2) and (5), represented by the following equation:

$$\tau = J^T \left( M_x(u + M_d^{-1}(B_d\dot{e} + K_d e - f_e)) + C_x(x, \dot{x})\dot{x} + g_x(x) + f_e \right) \quad (31)$$

It is possible to obtain an important simplification considering that  $M_d = M_x$ . Moreover, it is also possible to consider "quasi-static" conditions, which means that velocities  $\dot{x} \approx 0$  and  $\dot{q} \approx 0$ . This is a reasonable approximation since force control tasks are generally executed at low speeds. Thus, the following simplified impedance control law is obtained:

$$\tau = J(q)^T (B_d\dot{e} + K_d e) + g(q) \quad (32)$$

This control scheme is equivalent to a proportional-derivative cartesian position control law with gravity compensation. It is possible to demonstrate that for a constant reference  $x_d$  and without contact with the environment, asymptotic stability of  $x_d$  is obtained (de Witt et al, 1997). This control law can be used to apply a desired contact force on the environment considering the virtual references given by the target vector  $x_t = [x_v \quad x_d]^T$ . The application of this vector in the control law (32), allows a force tracking capability without explicit knowledge of the environment stiffness coefficient. In order to improve the force tracking accuracy, an integral control action can be included in (32), which leads to:

$$\tau = J(q)^T \left( B_d\dot{e} + K_d e + K_i \int e_f dt \right) + g(q) \quad (33)$$

where  $e = x_t - x$  and  $\dot{e} = \dot{x}_t - \dot{x}$  are redefined as virtual position and velocity errors and  $e_f = f_d - f_e$  are the force errors. Taking the previous assumptions in consideration, the impedance control algorithm with force tracking implemented in real-time is given by (33) where the gravitational terms  $g(q)$  are null since the robot is of planar type. The main steps of the controller implementation for each sampling period  $T_s$  are (Baptista et al., 2001b):

- Step 1: calculation of cartesian positions and velocities,  $x(k)$  and  $\dot{x}(k)$ , respectively.
- Step 2: calculation of the normal contact force  $f_n(k)$  from JR<sup>3</sup> sensor data.
- Step 3: calculation of  $x_v(k)$  in the constrained direction ( $y$ -axis):

$$x_v(k) = \begin{cases} x_v(k-1) - \frac{\delta f_d(k)}{k_d} & \text{if } f_n(k) \geq 0 \\ x_v(k-1) - \frac{\delta f_d(k) + \kappa_f e_f(k)}{\hat{k}_e} & \text{if } f_n(k) < 0 \end{cases} \quad (34)$$

where

$$\delta f_d(k) = f_d(k) - f_d(k-1) \quad (35)$$

$$e_f(k) = f_d(k) - f_n(k) \quad (36)$$

- Step 4: definition of the virtual vector

$$x_v(k) = \begin{cases} x_d(k) & \square \text{ } x\text{-axis direction} \\ x_v(k) & \perp \text{ } y\text{-axis direction} \end{cases} \quad (37)$$

- Step 5: calculation of the control actions in the Cartesian space:

$$f(k) = -B_d \dot{x}(k) + K_d e(k) + K_i (T_s I_p(k)) \quad (38)$$

where

$$I_p(k) = \frac{e(k) + e(k-1)}{2} \quad (39)$$

- Step 6: calculation of the joint control actions

$$\tau(k) = J(q(k))^T f(k) \quad (40)$$

Notice that in (35) the minus signals follow the convention of signals used by the force/torque sensor. In this algorithm, an estimate of the environment stiffness  $\hat{k}_e$  is used to compute  $x_v(k)$ . The term  $\kappa_f e_f(k)$  is used to compensate geometric and stiffness uncertainties and other non-modeled terms of the environment.

For purposes of performance analysis, the classical hybrid position/force control algorithm was also implemented in real-time. This algorithm has been proposed by (Raibert & Craig, 1981) and uses a selection matrix  $S = \text{diag}(s_j)$  with  $j = 1, \dots, m$ , to separate the position and force subspaces, by the attribution of "1" to the force subspace and "0" to the position subspace. The matrix  $S$  is used to represent the force controlled direction and the matrix  $I-S$  to represent the remaining position controlled directions, where  $I$  is the iden-

tity matrix. In this algorithm, the control forces are given by:

$$f(k) = Sf_f(k) + [1 - S]f_p(k) \tag{41}$$

The real-time implementation of the control laws for unconstrained space and constrained space, are respectively given by:

$$f(k) = \begin{cases} f_d(k) + K_{pf}e(k) + T_s K_{if} I_{I_f}(k) - K_{Df} \dot{x} & \text{Force subspace} \\ K_p e(k) + T_s K_I I_p(k) - K_D \dot{x} & \text{Position subspace} \end{cases} \tag{42}$$

where  $e_f$  is defined in (37) and  $I_p$  follows the definition in (39) for the unconstrained direction. The coefficients  $K_p$ ,  $K_I$  and  $K_D$  are the parameters of the PID position controller and  $K_{pf}$ ,  $K_{if}$  and  $K_{Df}$  are the parameters of the PID force controller. Notice that velocity damping is included in the control law for the constrained direction (Mandal & Payandeh, 1995).

Unfortunately, the control scheme presented in fig.4 is not possible to implement directly in real-time due to the high sampling rate used in the force control loop ( $F_s=1$  KHz). Thus, one possible solution is to use more powerful hardware and software tools to implement the control scheme in a straightforward way. Due to computer hardware limitations, an alternative strategy was used to implement the proposed force control algorithm in real-time. The optimized virtual trajectory  $x_v$  for the impedance controller is computed off-line in Matlab/Simulink using a realist simulator of the experimental robotic setup. The optimized virtual trajectory calculated off-line with an additional term for compensation of uncertainties and modeling errors is then furnished as a reference signal ( $x_{vopt}$ ) to the impedance controller described in Section 3. The virtual reference used in the simplified version of the MPA with fuzzy scaling implemented in real-time is then given by:

$$x_v(k) = x_{vopt}(k) + \kappa_f \frac{e_f(k)}{\hat{k}_e} \tag{43}$$

where  $\kappa_f$  and  $\hat{k}_e$  are terms as defined in (34).

## 7. Experimental results

In this section, experimental results obtained with the predictive force algorithm with fuzzy scaling proposed in this paper are presented. These results are compared with the classical force control algorithms, namely the classical impedance controller and the hybrid position/force controller. Note that the predictive control algorithm uses a nonlinear model of the nonrigid environment where the stiffness coefficient is varying with the penetration depth  $\delta x$ .

It is necessary to obtain an estimate of the environment stiffness in the constrained direction  $\hat{k}_e$ , which is used in (34) and (43). This estimate is obtained experimentally by measuring the linear surface displacement when a force profile is applied by the end-effector. The collected pairs of  $f_n - \delta x$  data are then used to estimate the global stiffness coefficient  $\hat{k}_e = 1636$  N/m, using the least-squares method. The obtained results presented in fig.13, exhibit a large deformation of the surface when a 10 N smooth step force profile is applied on the surface. From the plot is possible to observe that the environment has a nonlinear behavior of hysteresis type (Baptista et al., 2001b).

The control algorithms are implemented in the cartesian space with a sampling period of  $T_s = 1$  ms. The implemented force/position task is a linear trajectory with 100 mm length between  $x_i = 229.5$  mm and  $x_f = 329.5$  mm along  $x$ -axis, with simultaneous application of a smooth 10 N step force profile along the  $y$ -axis, during 8 seconds. It is assumed that the arm is initially in contact with the environment. Moreover, geometric uncertainty in the contact surface of 5 mm at the end of the trajectory is intentionally imposed to the environment. The filtered force data components ( $f_x, f_z$ ) are obtained from the JR<sup>3</sup> DSP board considering a first order low-pass digital filter with 500 Hz cut-off frequency in order to attenuate the force sensor noise. The predictive force control algorithm, which derives the virtual position to the impedance controller, use the parameters  $H_c = H_p = 2$ . As refereed in Section 4, the fuzzy scaling machine is applied only during the constant path of the reference force profile.

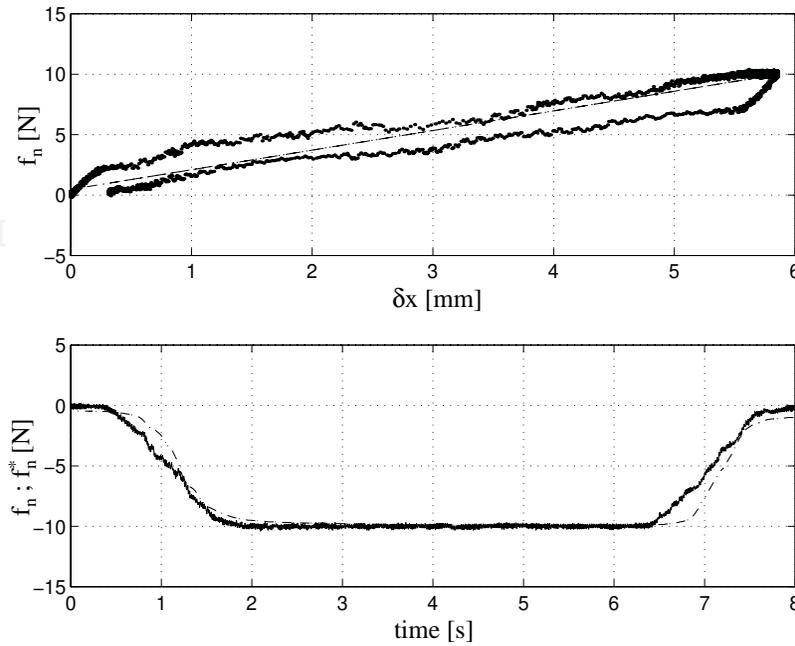


Figure 13. Estimated environment stiffness and contact forces. (Note: dashdot - estimated contact force  $\hat{f}_n = \hat{k}_e \delta x$ ; solid - actual force  $f_n$ ).  
 (Reprinted from Baptista, L.; Sousa, J. & Sá da Costa, J. (2001b) with permission of Elsevier).

The maximum allowed increments  $\Delta x_v$  at each time step are the following (in mm):

$$\Omega^* = \begin{bmatrix} -0.75 \times 10^{-2} & 0 & 0.75 \times 10^{-2} \end{bmatrix} \tag{44}$$

The impedance control algorithm use the parameters presented in Table 3 for the real-time control experiments. The force compensation gain is set to  $\kappa_f = 0.025$ , and was selected from experimental trials.

Axis	$K_d$	$K_i$	$B_d$
x-axis	3000	4000	6
y-axis	2000	3000	2

Table 3. Impedance control parameters.

Figure 14 presents the end-effector’s position in the  $x$ - $y$  axis coordinates using the real-time implementation of the predictive force control algorithm with fuzzy scaling. For the same controller, the optimized virtual position given by (43) and the actual contact force are presented in fig.15. Figure 16 presents the obtained position and force errors. The experimental results obtained with the classical impedance controller are presented in fig. 17, where the position and

force errors are depicted. The parameters for the hybrid position/force controller used in the real-time tests are presented in Table 4 and the position and force errors are depicted in fig.18.

Axis	$K_P$	$K_I$	$K_D$
x-axis	3000	4000	6
y-axis	570	2500	2

Table 4. Parameters of the hybrid position/force controller.

The results presented for the predictive force control algorithm reveal an accurate force tracking accuracy when compared to the equivalent results obtained with the impedance and hybrid control algorithms. The proposed controller exhibit significantly better results, especially during the time-varying force reference trajectory. The impedance controller with force tracking reveal some difficulties in following the force reference path when this is not constant. The force control performance of the hybrid controller drops significantly during the final part of the trajectory due to the geometric uncertainty, which is not efficiently compensated by the algorithm.

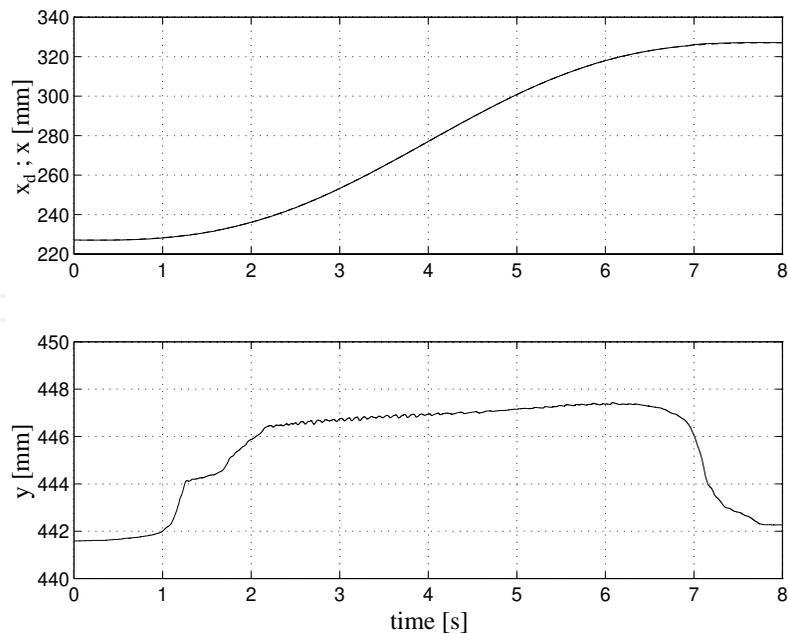


Figure 14. Predictive force control algorithm with fuzzy scaling: X-axis position coordinate, Y-axis constrained position coordinate.

(Reprinted from Baptista, L.; Sousa, J. & Sá da Costa, J. (2001b) with permission of Elsevier).

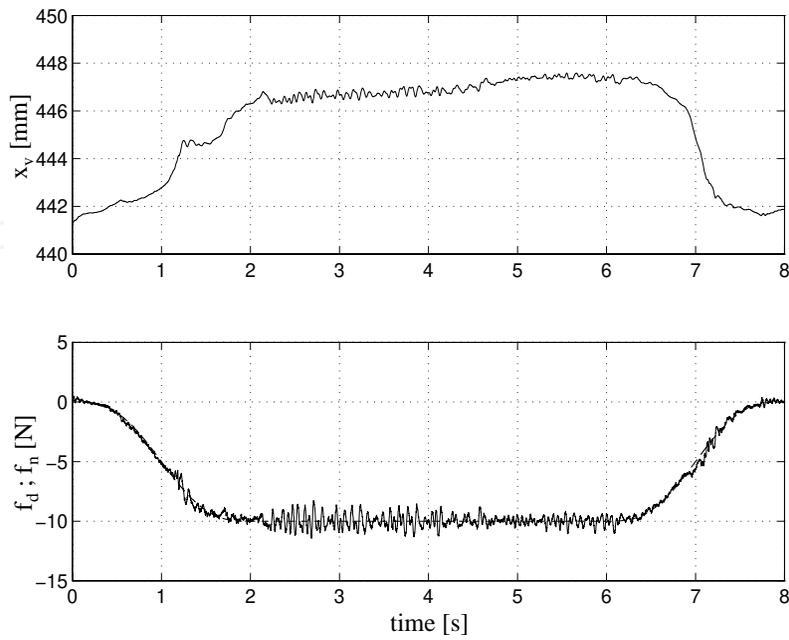


Figure 15. Predictive force control algorithm with fuzzy scaling: Y-axis virtual trajectory, desired and actual contact force (Note: solid-actual; dashdot-desired). (Reprinted from Baptista, L.; Sousa, J. & Sá da Costa, J. (2001b) with permission of Elsevier).

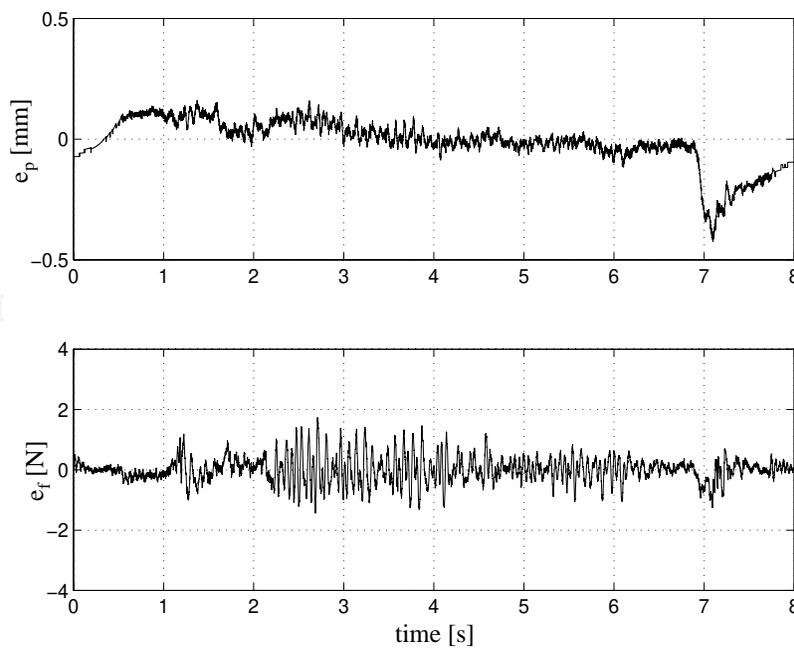


Figure 16. Predictive force control algorithm with fuzzy scaling: x-axis position errors and y-axis contact force errors. (Reprinted from Baptista, L.; Sousa, J. & Sá da Costa, J. (2001b) with permission of Elsevier).

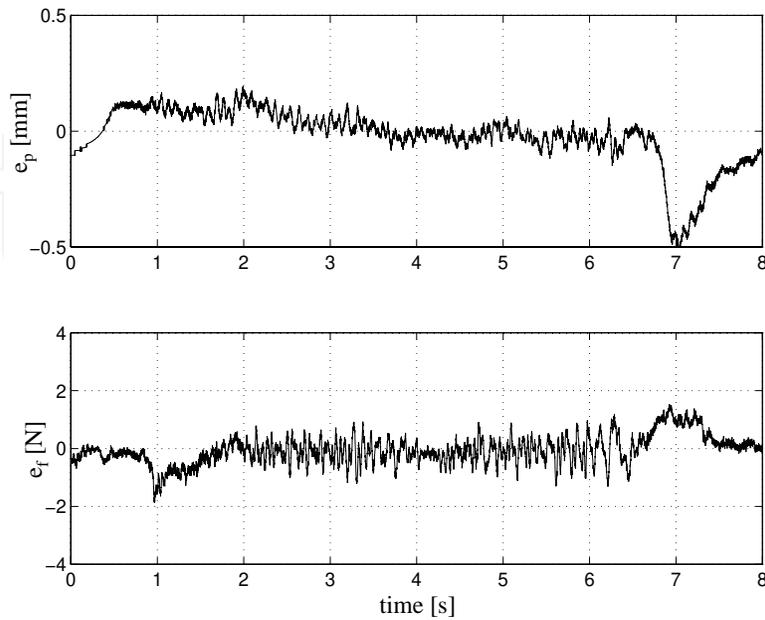


Figure 17. Impedance control with force tracking:  $x$ -axis position errors and  $y$ -axis contact force errors.

(Reprinted from Baptista, L.; Sousa, J. & Sá da Costa, J. (2001b) with permission of Elsevier).

In order to compare the performance of the controllers, the Euclidian norm ( $\|\cdot\|$ ) of position and force errors and the absolute maximum error values for the given task are presented in Table 5.

Force controllers	$\ e_p\ $ [m]	$\text{Max}(e_p)$ [mm]	$\ e_f\ $ [N]	$\text{Max}(e_f)$ [N]
Hybrid position/force control	8.1	0.43	63.89	2.76
Impedance control with force tracking	8.5	0.49	50.63	2.33
Predictive force control with fuzzy scaling	8.1	0.42	37.40	1.73

Table 5. Euclidian norm of position, force errors and absolute maximum errors.

From Table 4 results, is possible to conclude that predictive force control algorithm proposed in this paper reduces the norm of the force error in 26% when compared to the impedance controller with force tracking, which is a quite significant value. The reduction in the norm of the force error is even more significant when compared to the hybrid position/force controller results (41%). The experimental results reveal the successful implementation of the proposed predictive force control strategy in real-time, in the presence of non-rigid environments with uncertainties.

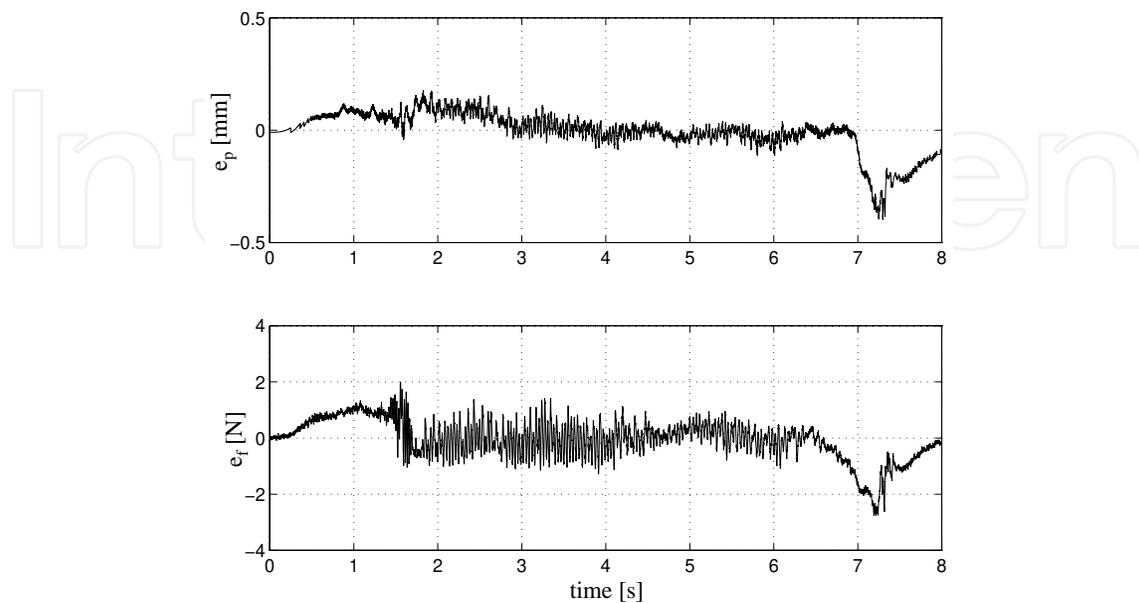


Figure 18. Hybrid position/force control:  $x$ -axis position errors and  $y$ -axis contact force errors.

(Reprinted from Baptista, L.; Sousa, J. & Sá da Costa, J. (2001b) with permission of Elsevier).

## 8. Conclusion

In this article, a force control scheme integrating a model predictive algorithm and an impedance controller in the presence of nonrigid environments is proposed. The force controller uses a MPA with a fuzzy scaling machine to alleviate the problems introduced by the discretization of inputs required by the B&B algorithm. The optimization procedure is nonconvex since a nonlinear model of the environment is considered. Simulation results were carried out considering a 2-DOF robot model interacting with a nonrigid environment with nonlinear characteristics. A real-time implementation of the fuzzy predictive algorithm is derived, which uses an optimal virtual trajectory computed off-line. The experimental results are compared to other classical force control techniques, which are the impedance control with force tracking and the hybrid position/force control. Simulation and experimental results reveal that the proposed MPA with fuzzy scaling force controller presents a significant better force tracking performance when compared to the conventional force controllers.

Future research will concentrate on the on-line implementation of the full version of the fuzzy predictive force control algorithm and the extension of the control scheme to a robot with more than two joints.

## ACKNOWLEDGMENTS

Part of the material in this chapter appeared in various journal and conference publications. These publications are referred to in the relevant paragraphs, and they are listed in the bibliography. The material, including figures and tables, is reproduced with the kind permission of the respective copyright holders. The list of the copyright holders is as follows: Springer Science and Business Media: Baptista, L.; Sousa, J. & Sá da Costa, J. (2001a). Elsevier Science: Baptista, L.; Sousa, J. & Sá da Costa, J. (2001b).

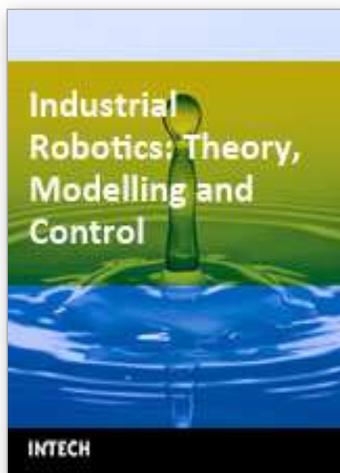
## 9. References

- Baptista, L.; (2000a). *Adaptive force and position control of robot manipulators*, PhD Thesis, Technical University of Lisbon, Mechanical Engineering Department, Lisbon (in portuguese)
- Baptista,L; Ayala Botto, M. & Sá da Costa, J. (2000b). A predictive force control approach of robot manipulators in nonrigid environments. *International Journal of Robotics and Automation*, 15(2) pp.94-102, ISSN 0826-8185
- Baptista,L; Sousa, J. & Sá da Costa, J. (2001a). Force control of robot manipulators using a fuzzy predictive approach. *Journal of Intelligent and Robotic Systems*, 30(4), April 2001, pp.359-376, ISSN 0921-0296
- Baptista,L; Sousa, J. & Sá da Costa, J. (2001b). Force predictive algorithms applied to real-time force control. *Control Engineering Practice*, 9(4), pp.411-423, ISSN 0967-0661
- Corbet, T. ; Sepehri, N. & Lawrence, P. (1996). Fuzzy control of a class of hydraulically actuated industrial robots. *IEEE Transactions Control Systems Technology*, 4(4), pp.419-426
- De Witt, C. ; Siciliano, B. & Bastin, G. (1997). *Theory of robot control*. Springer-Verlag, ISBN 3-540-76054-7, London, UK
- De Schutter, J. ; Bruyninckx, H. & Spong, M. (1997). *Force control: a bird's eye*. *Proceedings of IEEE CSS/RAS International Workshop on Control Problems in Robotics and Automation: Future directions*. San Diego, December 1997, USA
- Economou, C. ; Morari, M. & Palsson, B. (1986). *Internal model control. 5. Extension to nonlinear systems*. *Industrial Engineering Chemical Process Design and Development*, 25, American Chemical Society, pp.404-411
- Hogan, N. (1985). Impedance control: An approach to manipulation: Part I-II-III. *Journal of Dynamic Systems, Measurement and Control*, 107(1), pp.1-24
- Jaritz, A. & Spong, M. (1996). An experimental comparison of robust control algorithms on a direct drive manipulator. *IEEE Transactions on Control Systems Technology*, 4, pp.614-626

- Jung, S. & Hsia, T. (1995). On neural network application to robust impedance control of robot manipulators. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 869-874
- Lin, S. & Huang, A. (1997). Position-based fuzzy force control for dual industrial robots. *Journal of Intelligent and Robotic Systems*, 19, pp.393-409
- Liu, M. (1995). Force-controlled-fuzzy-logic-based robotic deburring. *Control Engineering Practice*, 3(2), pp.189-201, ISSN 0967-0661
- Love, L. & Book, W. (1995). Environment estimation for enhanced impedance control. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1854-1859
- Kieffer, J. & Yu, K. (1999). Robotic force/velocity control for following unknown contours of granular materials. *Control Engineering Practice*, 7(10), October 1999, pp.1249-1256, ISSN 0967-0661
- Klir, G. & Yuan, B. (1995). *Fuzzy set and fuzzy logic: theory and applications*, Englewood Cliffs, Prentice Hall, ISBN 0792377338, NJ
- Maciejowski, J. (2002). *Predictive control with constraints*, Prentice Hall Inc., New York, USA, ISBN 0201398230
- Marhefka, D. & Orin, D. (1996). Simulation of contact using a nonlinear damping model. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp.1662-1668
- Mandal, N. & Payandeh, S. (1995). Control strategies for robotic control tasks: an experimental study. *Journal of Robotic Systems*, 12(1), pp.67-92
- Raibert, M & Craig, J. (1981). Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement and Control*, 102(2), pp.126-133
- Siciliano, B. & Villani, L. (2000). *Robot force control*, Springer-Verlag, New York, ISBN 0792377338
- Singh, S, & Popa, D. (1995). An analysis of some fundamental problems in adaptive control of force and impedance behavior: theory and experiments. *IEEE Transactions on Robotics and Automation*, 11, pp.912-921
- Sousa, J. ; Babuska, R. & Verbruggen, H. (1997). Branch-and-bound optimization in fuzzy predictive control: an application to an air conditioning system. *Control Engineering Practice*, 5(10), October 1997, pp.1395-1406, ISSN 0967-0661
- Sousa, J. & Setnes, M. (1999). Fuzzy predictive filters in model predictive control. *IEEE Transactions on Industrial Electronics*, 46(6), December 1999, pp.1225-1232
- Volpe, R. & Khosla, P. (1995). On the equivalence of second order impedance control and proportional gain explicit force control. *International Journal of Robotics Research*, 14(6), pp.574-589
- Yao, B. & Tomizuka, M. (1995). Adaptive control of robot manipulators in constrained motion - controller design. *Journal of Dynamic Systems, Measurement and Control*, 117, pp.320-328

- Wada, H. ; Fukuda, T. ; Kosuge, K. ; Arai, F. & Watanabe, K. (1993). Damping control with consideration of dynamics of the environment. *Proceedings of the International Conference on Intelligent Robots and Systems*, 1, pp.1516-1520
- Zeng, G. & Hemami, A. (1997). An overview of robot force control. *Robotica*, 15(5), pp.473-482

IntechOpen



## **Industrial Robotics: Theory, Modelling and Control**

Edited by Sam Cubero

ISBN 3-86611-285-8

Hard cover, 964 pages

**Publisher** Pro Literatur Verlag, Germany / ARS, Austria

**Published online** 01, December, 2006

**Published in print edition** December, 2006

This book covers a wide range of topics relating to advanced industrial robotics, sensors and automation technologies. Although being highly technical and complex in nature, the papers presented in this book represent some of the latest cutting edge technologies and advancements in industrial robotics technology. This book covers topics such as networking, properties of manipulators, forward and inverse robot arm kinematics, motion path-planning, machine vision and many other practical topics too numerous to list here. The authors and editor of this book wish to inspire people, especially young ones, to get involved with robotic and mechatronic engineering technology and to develop new and exciting practical applications, perhaps using the ideas and concepts presented herein.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

L.F. Baptista, J.M.C. Sousa and J.M.G. Sa da Costa (2006). Predictive Force Control of Robot Manipulators in Nonrigid Environments, *Industrial Robotics: Theory, Modelling and Control*, Sam Cubero (Ed.), ISBN: 3-86611-285-8, InTech, Available from:

[http://www.intechopen.com/books/industrial\\_robotics\\_theory\\_modelling\\_and\\_control/predictive\\_force\\_control\\_of\\_robot\\_manipulators\\_in\\_nonrigid\\_environments](http://www.intechopen.com/books/industrial_robotics_theory_modelling_and_control/predictive_force_control_of_robot_manipulators_in_nonrigid_environments)

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2006 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen