

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 $d(\dot{q})q \in R^{n \times 1}$ 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 ρ_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 δx . 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 μ is the dry friction coefficient between the end-effector and the contact surface.

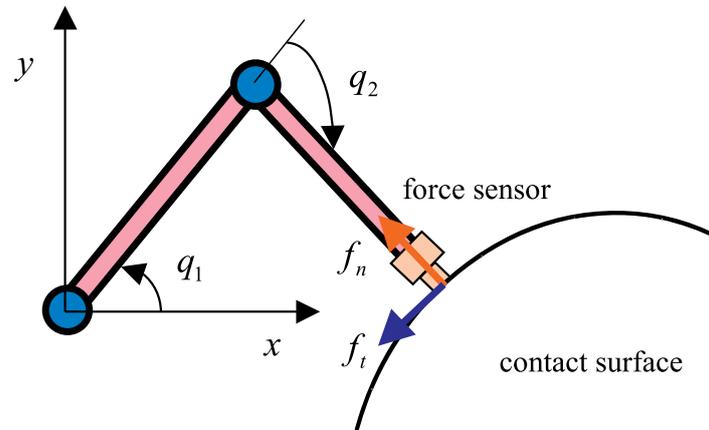


Figure 1. Robot manipulator applying a desired force on the environment.

(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:

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

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:

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

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:

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

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

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

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.

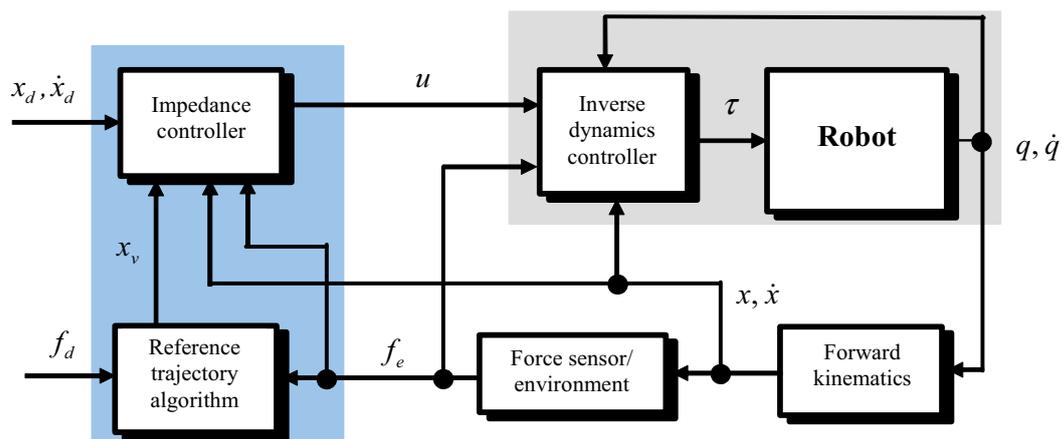


Figure 2. Impedance control with force tracking (ICFT) block diagram.

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.

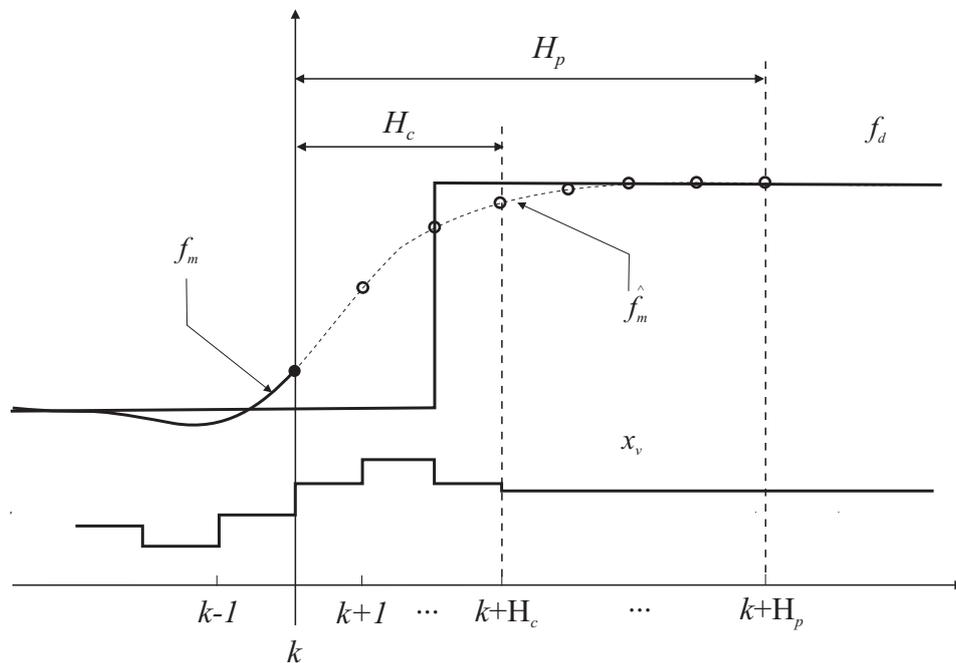


Figure 3. Basic principle of MPA applied to robot force control.

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

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

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

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

The distribution λ_l must be chosen taking into account that $0 \leq \lambda_l \leq 1$. In this way, the choice of λ_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 Ω_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 Ω_k^* in order to obtain the scaled inputs Ω_k of the optimization routine, the B&B in this case:

$$\Omega_k = \gamma(k) \cdot \Omega_k^* \quad (24)$$

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

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

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

$$\Delta e(k) = e(k) - e(k-1). \quad (26)$$

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 μ_γ :

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

Summarizing, the set of inputs Ω_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 Ω_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: λ_l , and the membership functions μ_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 λ_l .

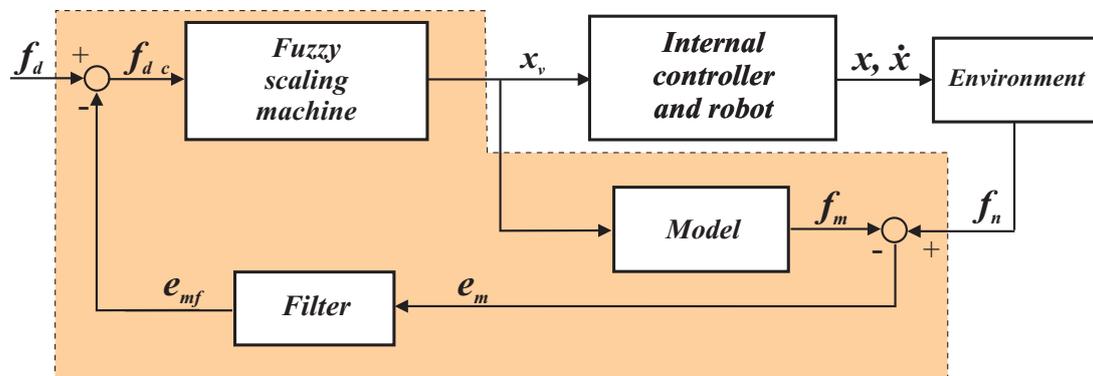


Figure 4. Block diagram of proposed predictive force control algorithm with fuzzy scaling machine.

(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/m² 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.

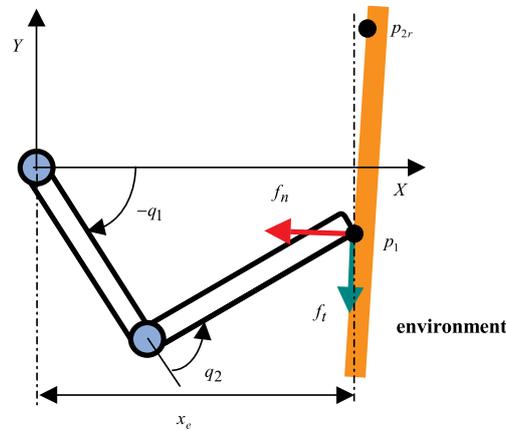


Figure 5. 2-DOF planar robot in contact with the environment.

(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 Δx_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.

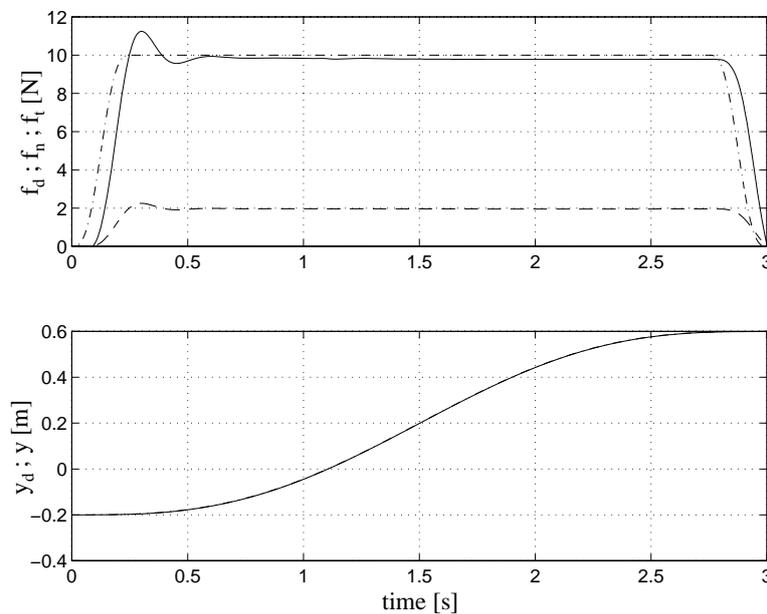


Figure 6. Impedance control with force tracking: desired force (dashdot), normal force (solid) and friction force (dashed) – top view; desired y-axis trajectory (dashdot) and actual position trajectory (solid) – bottom view.

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

Thank You for previewing this eBook

You can read the full version of this eBook in different formats:

- HTML (Free /Available to everyone)
- PDF / TXT (Available to V.I.P. members. Free Standard members can access up to 5 PDF/TXT eBooks per month each month)
- Epub & Mobipocket (Exclusive to V.I.P. members)

To download this full book, simply select the format you desire below

