Control and Cybernetics

vol. **42** (2013) No. 3

Stiffness control of robot manipulators in the operational space using fuzzy mapping of dynamic functions*

by

Douglas Wildgrube Bertol[1], Victor Barasuol[1], Nardênio Almeida Martins[1,2] and Edson Roberto De Pieri [1]

[1] Universidade Federal de Santa Catarina, Departamento de Automação e Sistemas,Grupo de Pesquisa Robótica, C. P. 476, 88040-900, Florianópolis, SC, Brasil
{dwbertol, victor, nardenio, edson}@das.ufsc.br
[2] Universidade Estadual de Maringá, Departamento de Informática Avenida Colombo, 5790, 87020-900, Maringá, PR, Brasil
nardenio@din.uem.br

Abstract: In this paper a stiffness control strategy based on the fuzzy mapped nonlinear terms of the robot manipulator dynamic model is proposed. The proposed stiffness controller is evaluated on a research robot manipulator performing a task in the operational space. Tests attempted to achieve fast motion with reasonable accuracy associated with lower computational load compared to the non-fuzzy approach. The stability analysis is presented to conclude about the mapping error influence and to obtain precondition criteria for the gains adjustment to face the trajectory tracking problem. Simulation results that supported the implementation are presented, followed by experiments and results obtained. These tests are conducted on a robot manipulator with SCARA configuration to illustrate the feasibility of this strategy.

Keywords: stiffness control, fuzzy mapping, robot manipulators, operational space, trajectory tracking

1. Introduction

Since the first robot manipulators involved in industrial processes were required to perform positional tasks (e.g., spray painting), robot manipulators were manufactured to be very rigid. This rigid design allowed the robot manipulator control designer to obtain reasonable positional accuracy by utilizing simple control laws (Natale, 2010; Lewis et al., 2004). As one might expect, force control applications (e.g., grinding or sanding) are extremely difficult to accomplish with

*Submitted: August 2012; Accepted: March 2013

a stiff robot manipulator. Therefore, if the robot manipulator stiffness could be controlled, force control applications could be accomplished more easily (Natale, 2010; Lewis et al., 2004).

In stiffness control, the joint stiffness matrix is modulated to achieve the desired relationship between position and applied force (Tischler, 2000; Yu, 2000). Stiffness control can be further classified into passive stiffness control and active stiffness control. Passive stiffness control is achieved by equipping the robot manipulator end-effector with a mechanical device composed of passive springs and dampers. In active stiffness control, the robot manipulator end-effector stiffness is changed based on the position and/or force feedback signals (Tischler, 2000). A stiffness control scheme was implemented in Cartesian coordinates in Tischler (2000), where the stiffness was specified in the workspace. Three translational and three rotational stiffness coefficients were specified for force control. Based on the difference between desired and actual end-effector position, a desired force is obtained. Stiffness may be changed under computer control to match varying task requirements. The implementation of this type of stiffness control was reported for tendon tension robot manipulator control by Tischler (2000). End-effector stiffness control for over-constrained systems was considered by Yu (2000). An over-constrained system contains more independent inputs than the number of kinematic degrees of freedom.

Stiffness control is sensitive to the combined stiffness of the environment, the end-effector, and the force sensor. Uncertainty in stiffness value leads to poor force control performance. The effect of robot manipulator wrist stiffness on robot manipulator control was discussed by Tischler (2000). Stiffness control combined with adaptive, learning, and robust control techniques were considered by Yu (2000) to handle stiffness uncertainty. Adaptive strategies were included into the stiffness controller to maintain consistent performance in the presence of unknown parameters of the robot manipulator and the environment in Yu (2000). Also, in Yu (2000), feedback controller gains were changed based on online estimates of stiffness matrix. A reinforcement learning process was proposed by Yu (2000) to compute the stiffness values of the end-effector for repetitive tasks.

When robot manipulators are requested to perform an agile motion, conventional strategies based on PD (proportional-derivative) control action can be inaccurate due to inherent nonlinear dynamics existing in real mechanisms. Facing these problems, the control action must be enhanced in such a way as to include some model features.

Other difficulties arise when characteristics of the model comprise the control action and tasks are performed in the operational space and controlled in the joint space. This control action becomes even more complex, since mathematical transformations require a substantial number of matrix operations, resulting in an excessive computational load (considering the average capacity of processing cores embedded in robot manipulators) and more expressive magnitudes of the partial error, due to parametric uncertainties.

A control methodology has been used to circumvent these inconveniences

by using kinematic transformations, carrying the control paradigm to the operational space, relating the joint torques to forces in the operational space by means of the Jacobian matrix (Siciliano et al., 2011; Sciavicco and Siciliano, 2009). To illustrate, Fig. 1 shows the control structure in the joint space (left) and in the operational space (right), where $X_d = [x_d \ \dot{x}_d \ \ddot{x}_d]^T$ is the desired trajectory matrix of the end-effector in the operational space, $X = [x \ \dot{x} \ \ddot{x}]^T$ is the posture matrix of the end-effector in the operational space, $Q_d = [q_d \ \dot{q}_d \ \ddot{q}_d]^T$ is the desired trajectory matrix in the joint space, $Q = [q \ \dot{q} \ \ddot{q}]^T$ is the posture matrix in the joint space and τ is the control torques vector in the joint space.

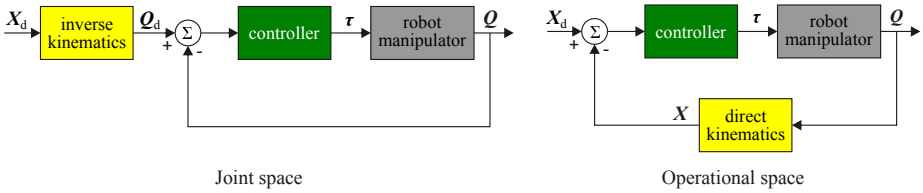

Figure 1. Control structures

Fuzzy sets and fuzzy systems have undergone a substantial development since their introduction by Zadeh (1973). The most used inference process was proposed by Mamdani (1974). However, in this paper the Sugeno inference process is used (Takagi and Sugeno, 1985), also known as Takagi-Sugeno or T-S fuzzy inference. A more recent debate about those pioneering works can be found in Zhang and Liu (2008); Lilly (2010); Piegat (2010).

Among the features of fuzzy systems, two justify its use in this study: the possibility of identifying systems using only sets of input-output pairs (Zhang and Liu, 2008) and the possibility to reduce the order of some terms of the system model with consequent reduction of computational load (Tanaka and Wang, 2001; Lilly, 2010).

Unlike the aforementioned works on stiffness control for robot manipulators and in order to face the operational space control problem with agile motion and low computational load, a stiffness controller is proposed with addition of model-based nonlinear terms mapped by a fuzzy inference system. This mapping provides the ability of reducing the amount of mathematical operations of the control law, decreasing the computational load.

This paper is organized as follows. In Section 2 the mathematical model of the robot manipulator as well as its relationships and transformations required to perform the operational space control are presented. In Section 3 the proposed control action is presented, and the stability analysis of the control system is given in Section 4. Section 5 provides the details of the mapping process. Finally, simulations, experimental results and conclusions are presented in Sections 6, 7 and 8, respectively.

2. Mathematical model of the robot manipulator

For the experiments performed in this paper a robot manipulator with a SCARA configuration (Selective Compliant Assembly Robot Arm), shown in Fig. 2, has been used. This robot manipulator, made in Robotics Institute of Zurich University in Switzerland by *Eidgenössische Technische Hochschule (ETH)*, is used for academic research at the *Laboratório de Controle e Automação (LCA)* of the *Departamento de Automação e Sistemas (DAS)* in the *Universidade Federal de Santa Catarina (UFSC)*.

The joints of this robot manipulator are actuated by four motors of Infranor brand. For joints 1 and 2, transmission is done by harmonic drives with reduction of (1 : 100). For joints 3 and 4 there is a combined linear/revolution spindle driven by timing belts (joint 4 has a planetary gear reduction (1 : 4.5)). It is emphasized that position measurements are obtained by means of incremental encoders. Additional information can be found in Passold (2009); Vargas (2005); Vargas et al. (2004); Mendes et al. (2002); Garcia et al. (2002); Hüipi and Gruener (2001); Weihmann (1999).

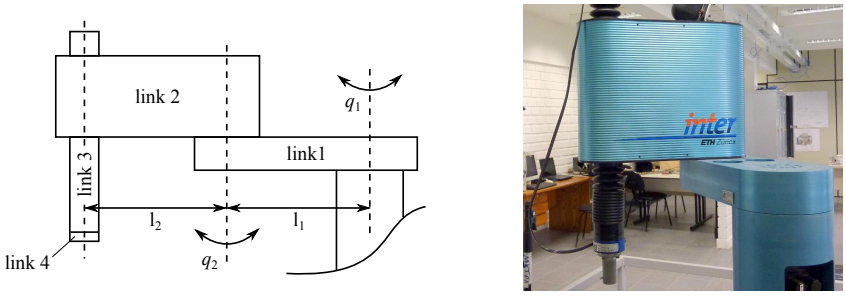

Figure 2. Inter SCARA robot manipulator

The robot manipulator dynamics can be represented by the following mathematical model (Siciliano and Khatib, 2008):

$$B(q)\ddot{q} + C(q,\dot{q})\dot{q} + F\dot{q} + g(q) = \tau. \tag{1}$$

This equation describes the robot manipulator dynamics in the joint space, where: $B(q)$ is the inertia matrix, $C(q,\dot{q})$ is the Coriolis and centrifugal forces matrix, F is the friction forces matrix, $g(q)$ is the gravitational forces vector and τ is the joint torques vector.

By using the relational property $\dot{x} = J(q)\dot{q}$, $J(q)$ being the Jacobian matrix rewritten as J for simplification, Eq. (1) can be converted to describe the robot manipulator dynamics in the operational space:

$$J^{-T}B(q)J^{-1}\ddot{x} + J^{-T}[C(q,\dot{q})J^{-1} - B(q)J^{-1}\dot{J}J^{-1}]\dot{x} + J^{-T}FJ^{-1}\dot{x}$$

$$J^{-T}g(q) = J^{-T}\tau. \tag{2}$$

By renaming some of the equation terms, Eq. (2) can be resumed to:

$$\overline{B}(q)\ddot{x} + \overline{C}(q,\dot{q})\dot{x} + \overline{F}\dot{x} + \overline{g}(q) = \overline{\tau}, \tag{3}$$

where $\overline{B}(q)$ is the symmetric inertia matrix, $\overline{C}(q,\dot{q})$ is the Coriolis and centrifugal forces matrix, $\overline{F}$ is the frictional forces matrix, $\overline{g}(q)$ is the gravitational forces vector and $\overline{\tau}$ is the forces vector. All now concern the operational space.

For the tests, a reduced mathematical model based on the first two joints, q_1 and q_2, of the robot manipulator is used, i.e., two degrees of freedom related to two rotational joints. Thereat, the vector of position joints is reduced to $q = [q_1\, q_2]^T$ and, since the two rotational joint axes are parallel, the workspace is reduced to a plane. Below we give the detailed matrices of the dynamic model that need to be computed to perform the control, as shown by Vargas (2005):

- Inertia matrix:
$$B(q) = \begin{bmatrix} b_1 + 2b_3\cos(q_2) & b_2 + b_3\cos(q_2) \\ b_2 + b_3\cos(q_2) & b_2 + j_2k_r^2 \end{bmatrix}, \tag{4}$$
where
$$b_1 = b_2 + i_1 + (m_2 + m_3 + m_4)l_1^2 + m_1l_{c1}^2 + j_1k_r^2,$$
$$b_2 = i_2 + i_3 + i_4 + m_2l_{c2}^2 + (m_3 + m_4)l_2^2 \quad \text{and}$$
$$b_3 = (m_2l_1l_{c2} + (m_3 + m_4)l_2l_1),$$
- Coriolis and centrifugal forces matrix:
$$C(q,\dot{q}) = \begin{bmatrix} -c_1\sin(q_2)\dot{q}_2 & -c_1\sin(q_2)(\dot{q}_1 + \dot{q}_2) \\ c_1\sin(q_2)\dot{q}_1 & 0 \end{bmatrix}, \tag{5}$$
where $c_1 = (m_2l_1l_{c2} + (m_3 + m_4)l_2l_1)$;
- Jacobian matrix:
$$J(q) = \begin{bmatrix} -l_1\sin(q_1) - l_2\sin(q_1 + q_2) & -l_2\sin(q_1 + q_2) \\ l_1\cos(q_1) + l_2\cos(q_1 + q_2) & l_2\cos(q_1 + q_2) \end{bmatrix}, \tag{6}$$
- Friction forces matrix:
$$F = \begin{bmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{bmatrix}, \tag{7}$$
where, for simplicity, the friction forces matrix is considered to be composed only of dynamic friction coefficients.

Due to the parallelism of the joints 1 and 2 with the gravity, the gravitational forces vector becomes redundant, i.e., $g(\mathrm{q}) = 0$. Table 1 lists the parameter values that compose the dynamic model matrices.

3. Control design

The goal of the proposed controller is to perform tasks in the operational space based on a reduced algorithm, which means low computational load. This is achieved by applying fuzzy logic to the control action, i.e. replacing the nonlinear terms of the matrix by means of fuzzy sets mapping.

For the design and implementation of the proposed controller as well as for the comparison of performance, the study was initiated with the conventional PD (Proportional-Derivative) controller, which is represented as follows:

$$\tau = J^T(k_p\tilde{x} + k_d\dot{\tilde{x}}), \tag{8}$$

Table 1. Parameters of the robot manipulator

Parameter	Description	Value
l_1	Length of link 1	0.25 [m]
l_2	Length of link 2	0.25 [m]
l_{c1}	Link 1 center of mass	0.118 [m]
l_{c2}	Link 2 center of mass	0.116 [m]
m_1	Mass of link 1	11.40 [kg]
m_2	Mass of link 2	19.40 [kg]
m_3	Mass of link 3	2.00 [kg]
m_4	Mass of link 4	1.50 [kg]
i_1	Inertia of link 1	0.23 [kg.m^2]
i_2	Inertia of link 2	0.16 [kg.m^2]
i_3	Inertia of link 3	0.10 [kg.m^2]
i_4	Inertia of link 4	0.10 [kg.m^2]
j_1	Inertia of rotor 1	5.00×10^{-5} [kg.m^2]
j_2	Inertia of rotor 2	5.00×10^{-5} [kg.m^2]
μ_1	Viscous friction of joint 1	11.50 [Nms/rad]
μ_2	Viscous friction of joint 2	6.00 [Nms/rad]
k_r	Joints gear relation	100

where $\tilde{x} = [x_d - x]^T$ is the end-effector position error vector, $\dot{\tilde{x}} = [\dot{x}_d - \dot{x}]^T$ is the end-effector velocity error vector, $x = [xy]^T$ is the end-effector position vector, $x_d = [x_d\ y_d]^T$ is the end-effector desired position vector, $\dot{x} = [\dot{x}\ \dot{y}]^T$ is the end-effector velocity vector, $\dot{x}_d = [\dot{x}_d\ \dot{y}_d]^T$ is the end-effector desired velocity vector, all in the operational space, k_p and k_d are the proportional and derivative gains, respectively.

The fundamental idea of the control action refers to the conventional PD controller (8) improved by terms which interact with inertial forces. Moreover, the terms in the control action related to accelerations are based on the properties of the inertia matrix (positive definite) so one can consider it as a gain with positive variable values. Thus, the basic control action is given by:

$$\tau = J^T(k_p\tilde{x} + k_d\dot{\tilde{x}} + \overline{B}(q)\ddot{x}_d), \tag{9}$$

where $\ddot{x}_d = [\ddot{x}_d\ \ddot{y}_d]^T$ is the end-effector desired acceleration vector in the operational space.

Compared with a conventional PD controller (8), the control action of (9) presents a considerable reduction of the tracking errors provided by the addition of the term $\overline{B}(q)\ddot{x}_d$. However, the proposed control structure is based on a stiffness control strategy presented by Slotine and Li (Slotine and Li, 1987, 2004; Lewis et al., 2004; Khalil and Dombre, 2004), which uses the concept of an auxiliary error to achieve better rates of convergence to the tracking errors. Thus, the control action of (9) is modified to provide such convergence properties, i.

e.,

$$\tau = J^T(k_d\sigma + \overline{B}(q)\ddot{x}_r), \tag{10}$$

where $\sigma = \dot{\tilde{x}} + k_p\tilde{x}$ is a filtered tracking error and $\dot{x}_r = \dot{x}_d + k_p\tilde{x}$ is the reference velocity based on the end-effector desired velocity vector and the end-effector position error vector, Fig. 3 shows the block diagram of this controller.

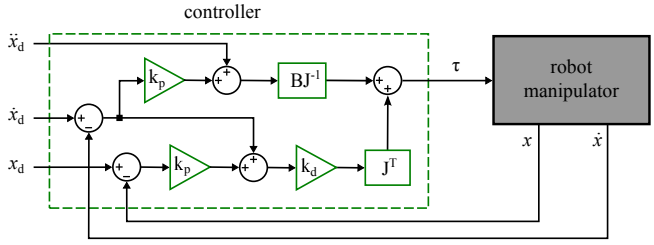

Figure 3. Block diagram of the controller proposed by Slotine and Li (1987)

To provide satisfactory results over tasks performed in the operational space with a low computational load, in this paper the insertion of fuzzy logic mapping to replace the nonlinear functions of the controller (10) is proposed. To do this, the inertia matrix $\overline{B}(q)$ is expanded, rewriting $B(q)$ as B for simplicity, to obtain:

$$\tau = J^T k_d\sigma + BJ^{-1}\ddot{x}_r. \tag{11}$$

Then, the mapping is made over the transposed Jacobian J^T, which is represented by $\overline{\overline{J^T}}$, and over the matrix BJ^{-1}, which is represented by $\overline{\overline{BJ^{-1}}}$, obtaining the following expression for the control law:

$$\tau = \overline{\overline{J^T}}k_d\sigma + \overline{\overline{BJ^{-1}}}\ddot{x}_r. \tag{12}$$

To complete the mathematical analysis of the controller, in the next section the stability analysis of the closed-loop system is provided.

4. Stability analysis

Consider the mapped terms rewritten as follows:

$$\overline{\overline{J^T}} = J^T(I + \Psi_1), \tag{13}$$

$$\overline{\overline{BJ^{-1}}} = BJ^{-1}(I + \Psi_2), \tag{14}$$

where Ψ_1 and Ψ_2 are matrices that contain error functions related to the mapping and vary with the robot manipulator posture. Substituting Eqs. (13) and (14) into Eq. (12), one can describe the closed-loop system dynamics as:

$$(I + \Psi_1)\overline{B}\dot{\sigma} + (I + \Psi_2)k_d\sigma = (\overline{C} + \overline{F})\dot{x} + \overline{B}\Psi_1\ddot{x}_d. \tag{15}$$

It is possible to conclude from (15) that the system will be stable when conditions $||\Psi_1||<1$ and $||\Psi_2 < 1||$ are satisfied. This is verified during the mapping stage. Such mapping process is explained in Section 5.

One can also note that it is impossible to achieve null errors when the robot manipulator performs motion, because the perturbation is directly related to the velocity and acceleration of the end-effector in the operational space.

This can be better noticed by expanding the variable σ in such a way as to rewrite (15) in terms of errors:

$$(I+\Psi_1)\overline{B}(\ddot{\tilde{x}}+k_p\dot{\tilde{x}})+(I+\Psi_2)k_d(\dot{\tilde{x}}+k_p\tilde{x}) = (\overline{C}+\overline{F})(\dot{x}_d-\dot{\tilde{x}})+\overline{B}\Psi_1\ddot{x}_d, \quad (16)$$

after some manipulation and replacement, (16) can be rewritten as,

$$\alpha_1\dddot{\tilde{x}} + \alpha_2\ddot{\tilde{x}} + \alpha_3\dot{\tilde{x}} = (\overline{C}+\overline{F})\dot{x}_d + \overline{B}\Psi_1\ddot{x}_d, \quad (17)$$

where $\alpha_1 = (I+\Psi_1)$, $\alpha_2 = (I+\Psi_1)\overline{B}k_p + (I+\Psi_2)k_d + \overline{C} + \overline{F}$ and $\alpha_3 = (I+\Psi_2)k_dk_p$.

Thus, in Eq. (17) one can identify the direct relationship of the perturbation with the desired velocities, the term $(\overline{C}+\overline{F})\dot{x}_d$, and the desired acceleration, the term $\overline{B}\Psi_1\ddot{x}_d$. In this way, given the absence of desired velocity and acceleration, i.e., in regulation cases, by Eq. (17), it can be concluded that the error tends to zero when the time goes to infinity.

5. Fuzzy mapping of dynamic functions

In this study we aim to compare the performance of the controllers presented until now, over the simulated and practical tests. To conduct such tests, avoiding the concern about having singularity postures during the running time, we chose to restrict the robot manipulator workspace, shown in Fig. 4, to the first quadrant and the radius range from 295 mm to 495 mm, this area being cross hatched in the same figure.

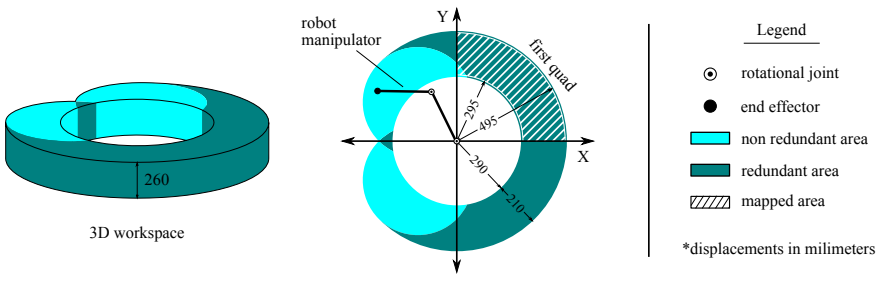

Figure 4. Workspace of the SCARA robot manipulator

First, in this section we describe the fuzzy mapping process of the nonlinear matrices J^T and BJ^{-1}, considering the delimited workspace. Each element of

these $\mathbb{R}^{2\times 2}$ matrices is shown in Fig. 5. This figure shows the values of the elements in terms of the end-effector position in polar coordinates (angle and radius).

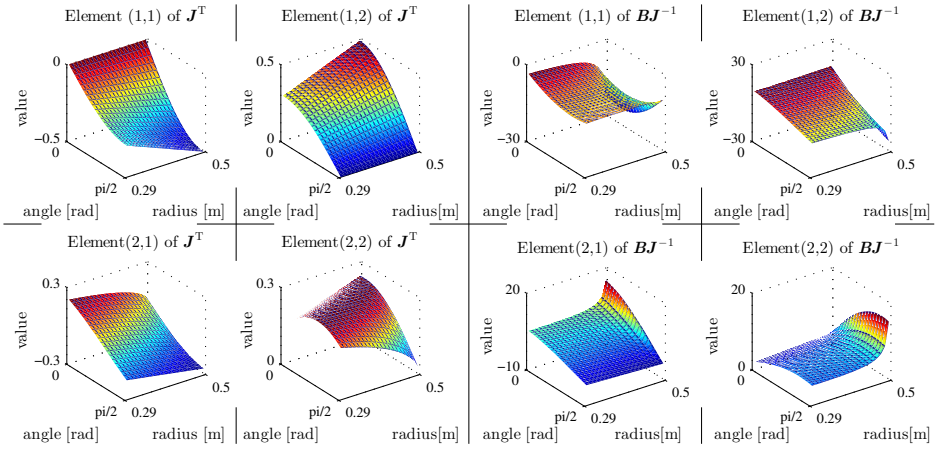

Figure 5. J^T and BJ^{-1} and elements

5.1. Mapping process

The mapping task was performed using the Adaptive Neuro-Fuzzy Inference System Editor (ANFIS Editor), which is part of the fuzzy logic toolbox of the Matlab® software. This tool provides a back-propagation algorithm combined with a least square method to tune the fuzzy rules of a Sugeno inference system using a set of conjugate input/output pairs from the function to be mapped. Since each of the matrix elements is a function of the robot manipulator posture (angle and radius), the mapping can be done using the information shown in Fig. 5.

Thus, for each element of the matrices J^T and BJ^{-1} one substitute Sugeno inference system was obtained. For these inference systems we could choose *output* membership functions (MFs), either linear or constant, and as *input* MFs any one that can represent a statistical distribution.

The selection of these input/output MFs has direct influence on the accuracy and computational load of the controller. For our tests, to reflect the cost-benefit between mapping error and number of mathematical operations, we chose two types of input MFs:

- Generalized bell-shaped

$$gbellmf(x) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}}. \tag{18}$$

This function computes the pertinence of the x element described by a bell-shaped distribution (see Fig. 6), where c is the center value of the

distribution, b is the center of the changing borders and a is the smoothness of the function. It is used here to obtain an accurate (low error) mapping, but requires a higher number of mathematical operations, compared with the *trimf* MF described below.

- Triangular-shaped

$$trimf(x) = max\left(min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right). \tag{19}$$

This function is used to compute the pertinence of x using a triangular distribution, where a is the beginning of the ascending part of the function, b is where the function attains its apex and the point of junction of the ascending and descending slopes, and c is the end point of the descendent slope. Fig. 6 illustrates this function.

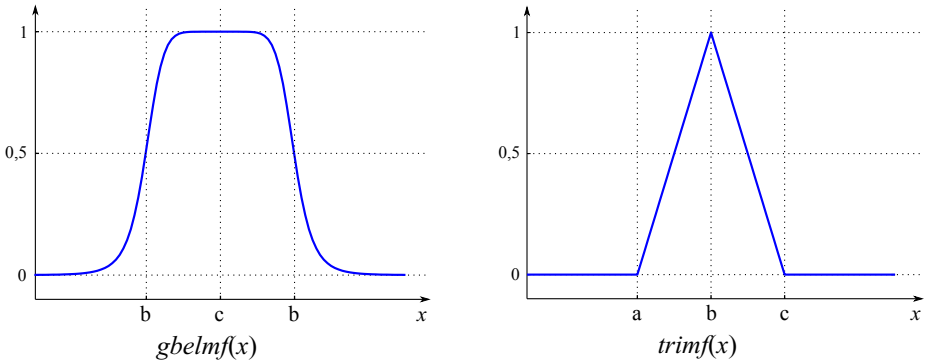

Figure 6. Membership functions

As a result of each mapping process the *ANFIS Editor* gives the quadratic error between the output reference data and the output data computed from the respective mapped fuzzy system. In the subsequent subsections the results of practical mapping of matrices J^T and BJ^{-1} are described.

5.2. Mapping of J^T

For mapping elements of the matrix J^J we used the data output of the Eq. (6) transposed, corresponding to the region defined in Fig. 4. During the mapping process we noticed that the error results of the mapping process using any combination of input/output MFs was considerably close and low. Thus the only concern about the mapping of this matrix was about the computational cost, not the accuracy. Moreover, because of the low variation between the angle and radius, we could map the function from only one of them.

Therefore, the mapping process of the matrix J^T picks the angle as the input variable, we partitioned the input region into three parts, each one described by a *trimf* MF, and selected a continuous output function for each fuzzy rule. In

Table 5.2 one can see these setup parameters used in the mapping process and the calculated quadratic error for each element of the matrix J^T. This resulting error is the main information obtained here, used to evaluate the mapping process over the stability of the system.

Table 2. J^T mapping configuration parameters and error

Setup parameters					Quadratic error
Element	Input variables	No. of rules	Input MF	Output MF	
$\overline{\overline{J^T}}_{11}$	angle	3	trimf	continuous	0.040
$\overline{\overline{J^T}}_{12}$	angle	3	trimf	continuous	0.040
$\overline{\overline{J^T}}_{21}$	angle	3	trimf	continuous	0.024
$\overline{\overline{J^T}}_{22}$	angle	3	trimf	continuous	0.046

In order to visually compare the mapped functions, in Fig. 7 one can see the original function elements of the matrix J^T on the left (2D projected over the value/angle plan) and the mapped elements on the right.

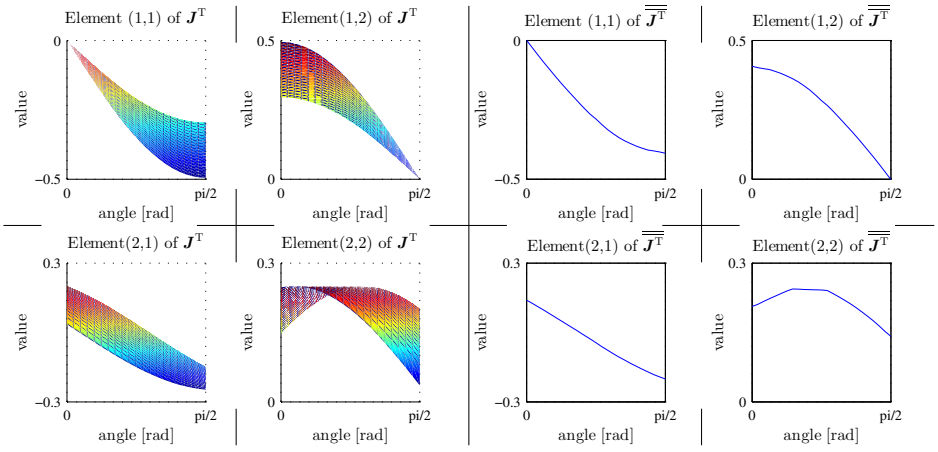

Figure 7. Mapped elements of $\overline{\overline{J}}$

5.3. Mapping of BJ^{-1}

Although the mapping process of the J^T presents low computational load and high mapping performance (low error) by using triangular functions and at most three fuzzy rules, the mapping of the matrix BJ^{-1} shows more variables of concern and finds the cost-benefit that fits a solution for the robot manipulator with low computational load and sufficient accuracy, i.e. the trade-off load/accuracy.

Two variants of fuzzy mapping were tested: one aiming at reduction of the mapping error using the generalized bell-shaped input MFs, see Eq. (18), with

linear output MFs; the other aimed at reduction of the computational load at the expense of the mapping error using the triangular-shaped input MFs, see Eq. (19), with constant output MFs.

Tables 3 and 4 depict the setup parameters and the calculated errors obtained for the matrix BJ^{-1}, for an accurate mapping process and a low cost mapping process, respectively. Of these two setups, the first one shows a very high mapping accuracy and the second shows a lower computational load.

Table 3. BJ^{-1} accurate mapping configuration parameters and error

Setup parameters					Quadratic error
Element	Input variables	No. of rules*	Input MF	Output MF	
$\overline{\overline{BJ^{-1}}}_{11}$	angle & radius	9 [3 3]	gbellmf	linear	0.098
$\overline{\overline{BJ^{-1}}}_{12}$	angle & radius	9 [3 3]	gbellmf	linear	0.070
$\overline{\overline{BJ^{-1}}}_{21}$	angle & radius	9 [3 3]	gbellmf	linear	0.050
$\overline{\overline{BJ^{-1}}}_{22}$	angle & radius	9 [3 3]	gbellmf	linear	0.066

Table 4. BJ low cost mapping configuration parameters and error

Setup parameters					Quadratic error
Element	Input variables	No. of rules*	Input MF	Output MF	
$\overline{\overline{BJ^{-1}}}_{11}$	angle & radius	[3 2]	trimf	continuous	0.563
$\overline{\overline{BJ^{-1}}}_{12}$	angle & radius	[2 2]	trimf	continuous	0.575
$\overline{\overline{BJ^{-1}}}_{21}$	angle & radius	[2 2]	trimf	continuous	0.424
$\overline{\overline{BJ^{-1}}}_{22}$	angle & radius	[2 2]	trimf	continuous	0.646

Fig. 8 shows the representation of the two mapping processes, the accurate (left) and the low cost (right). These two mappings could be compared with the original matrix in Fig. 5 (right).

Appendix in Section 9 provides all the mapping process results and the parameters needed to reproduce the fuzzy inference systems, described in Tables 6 through 14.

As shown in the next sections on simulated and practical tests, the control performance results display no significant differences. The performance results of both processes are so similar that is not possible to ensure that the most accurate is the best choice. So, the low cost solution appears to be a reasonable choice.

*The number of rules corresponds the combination of all MFs, e.g. with 6 [3 2] the fuzzy system has 6 rules: 3 MFs for the first input and 2 MFs for the second input.

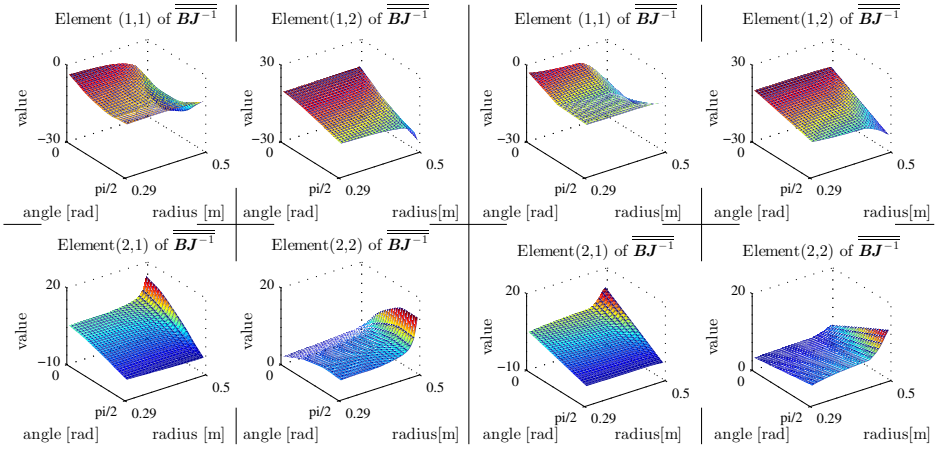

Figure 8. Mapped elements of $\overline{\overline{BJ^{-1}}}$ - accurate mapping (left) and low cost mapping (right)

6. Simulation tests

For numerical representation of the system to be controlled, as specified in Section 2 and illustrated in Fig. 2, the dynamic model for the first two links of the SCARA robot manipulator was used, see Eq. (1), with the parameters presented in Table 1. For simulation, the Matlab/Simulink software (Simulink, 2012) was used.

To evaluate the performance of the fuzzy mapping approach with J^T and BJ^{-1}, we used a desired trajectory in the workspace and tested the following four strategies: PD control given by Eq. (8); a PD control with model-based nonlinear terms (PD+MBT Control), as stated in Eq. (11); and the control given in Eq. (12), which has the model-based nonlinear terms mapped via fuzzy systems (PD+FM Control with accurate mapping and PD+FM Control with low cost mapping).

Fig. 9 shows the structure of the controllers, in which the desired trajectory block contains the reference trajectory; the control blocks represent the PD+FM (accurate mapping and low cost mapping), PD+MBT, and PD algorithms; the dynamic model is given by Inter SCARA robot manipulator dynamics block; the transformation of the angular displacements in joint space to the Cartesian positions in operational space is provided by the direct kinematics block.

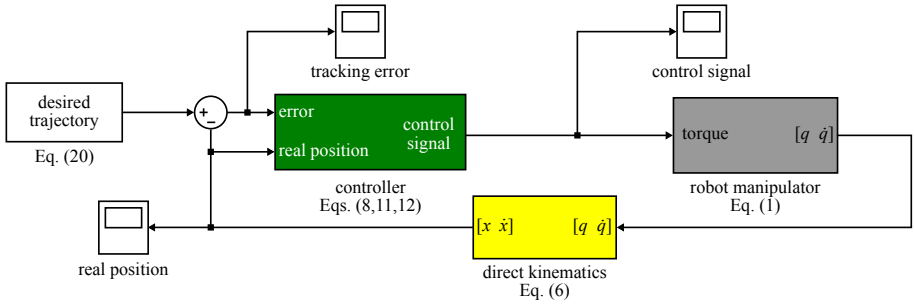

Figure 9. Block diagram of the controllers implemented in Matlab/Simulink

6.1. Robot manipulator test task

The desired task of the robot manipulator is described by a two-dimensional trajectory inside the workspace delimited in Section 5 and formulated as

$$\mathbf{x}_d = \begin{cases} (r + d_r \sin(w_r s)) \cos(c_r s + a_i) & \text{if } t \leq 10\,[s] \\ (r + d_r \sin(w_r s)) \sin(c_r s + a_i) & \text{otherwise} \end{cases}, \\ \mathbf{y}_\mathrm{d} = \begin{cases} (r + d_r \sin(w_r s)) \sin(c_r s + a_i) & \text{if } t \leq 10\,[s] \\ (r + d_r \sin(w_r s)) \cos(c_r s + a_i) & \text{otherwise} \end{cases}, \tag{20}$$

where $r = 0.39\,[m]$ is the central line of the trajectory, $d_r = 0.075\,[m]$ is the amplitude of the trajectory wave, $\mathrm{w_r} = \pi\,[\mathrm{rad/s}]$ is the angular velocity of the trajectory wave, $a_i = \frac{\pi}{180}\,[rad]$ is the initial angular position, $a_f = \frac{89\pi}{180}\,[rad]$ is the final angular position, $c_r = \frac{a_f - a_i}{10}\,[rad/s]$ is the angular velocity over the quadrant displacement and s is a parameter of time, described as

$$s = \begin{cases} 5 - 5\cos(0.3112t) + 0.07388\sin(0.3112t) & \text{if } t \leq 10\,[s] \\ 5 - 5\cos(0.3112(t-10)) + 0.07388\sin(0.3112(t-10)) & \text{otherwise} \end{cases}, \tag{21}$$

this s parameter having been selected to create a smooth transition at the discontinuity of the trajectory and to ensure a soft start and end of the execution.

An illustration of the parameters and the trajectory is shown in Fig. 10. As complementary information, Fig. 11 shows the desired velocities and accelerations required by the reference trajectories.

6.2. Results

In order to verify the effect of the nonlinear terms added in (11) and (12), all of the PD actions in all four strategies have equivalent gains, as given in Table 6.2, and all controllers have the sampling period of 1.0 ms.

The gains were set initially based on Ziegler-Nichols parameters and then better adjusted empirically using the trial-and-error method after a few observations of the amplitude of control signals and the reference tracking errors. In

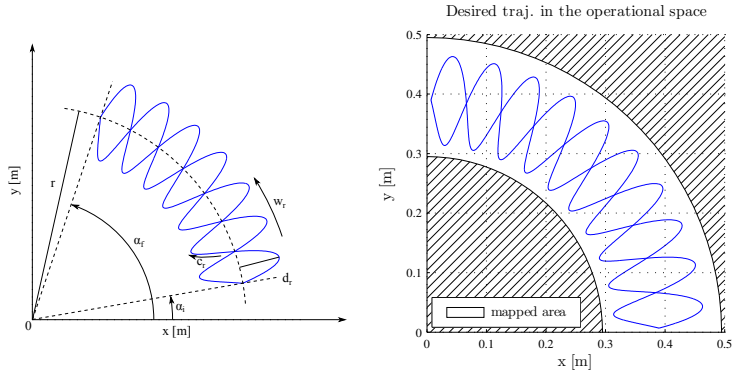

Figure 10. The desired trajectory and its parameters

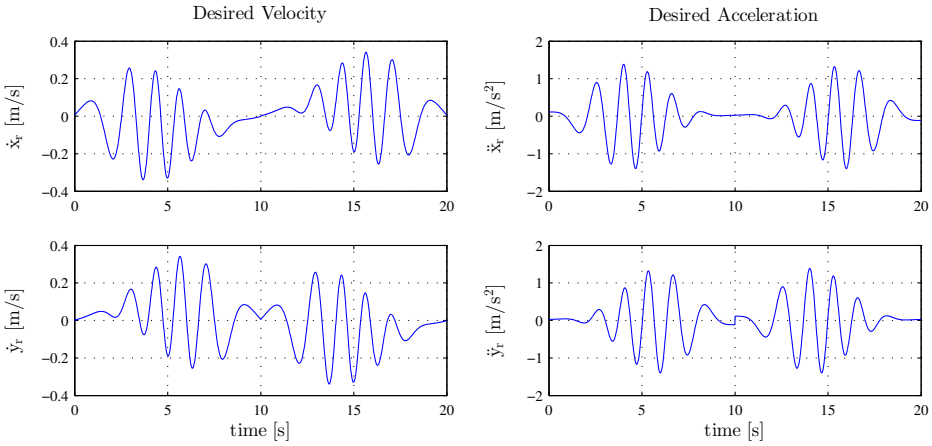

Figure 11. The desired velocities and accelerations

Table 5. Equivalence of control actions - controller parameters

Controller	k_p gain	k_d gain	Equivalent P action	Equivalent D action
PD - Eq. (8)	20000	200	20000 (k_p)	200 (k_d)
PD+MBT - Eq. (11)	100	200	20000 ($k_p k_d$)	200 (k_d)
PD+FM - Eq. (12)	100	200	20000 ($k_p k_d$)	200 (k_d)

the literature, other methodologies, based on adaptive control, neural networks and genetic algorithms were applied to the tuning of classical and advanced controllers (Mendes et al., 2002) that can be used here, but that will result in increased computational load.

It is important to mention that the robot manipulator has harmonic drives as transmission system device, therewith the friction effects are dynamically

significant. Thus, to improve the quality of the controller a simple friction compensation could be inserted in parallel to the PD strategy. However, for the tests, the controllers specified up to now were only used, i.e., friction compensation was not employed.

For a better evaluation of performance, Figs. 12 and 13 show the corresponding trajectory tracking errors. The maximum position errors are about 2.0×10^{-2} m for the PD control, 6.0×10^{-3} m for the PD+MBT control, and 6.0×10^{-3} m for the PD+FM (accurate mapping and low cost mapping) control.

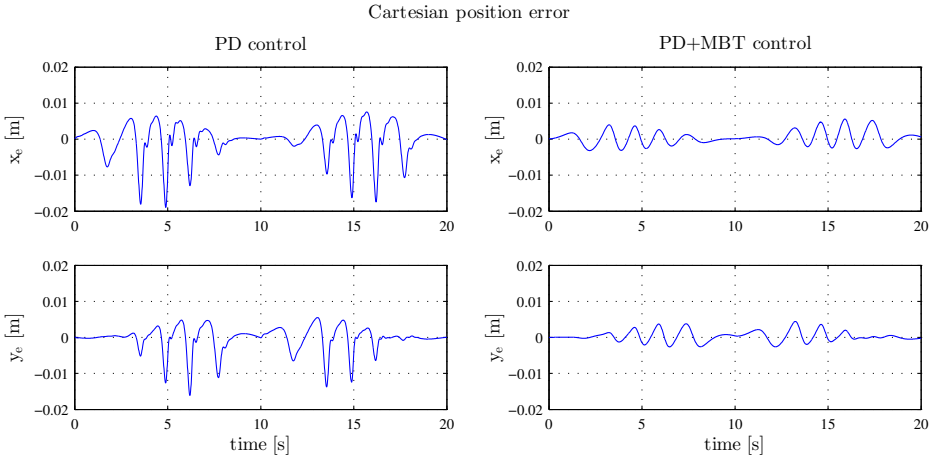

Figure 12. Simulation results: trajectory tracking errors - PD and PD+MBT controllers

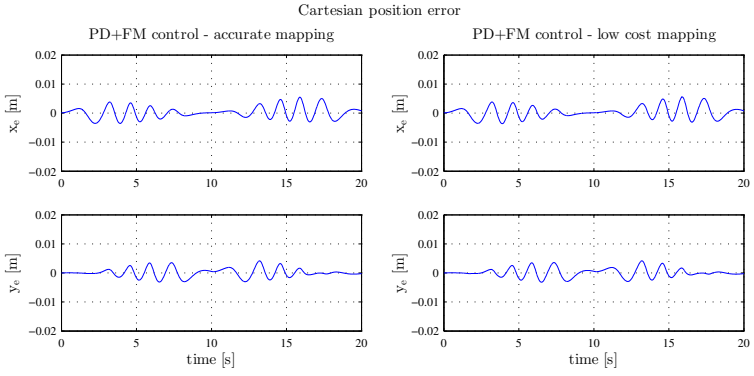

Figure 13. Simulation results: trajectory tracking errors - PD+FM (accurate mapping and low cost mapping) controllers

The control performance can be seen in Figs. 14 and 15 (left), in which

the PD controller performs the trajectory tracking in an unsatisfactory manner while PD+MBT and PD+FM controllers follow the path with a better accuracy.

Also, in Figs. 14 and 15 (right), the control action applied to the robot manipulator is shown. Note that control signals are kept between maximum levels given by the controller parameters, which are below the limits of the robot manipulator used in the practical implementation. Limit values of the robot manipulator are (Passold, 2009; Vargas, 2005; Vargas et al., 2004; Mendes et al., 2002; Hüipi and Gruener, 2001; Weihmann, 1999): 333.0 Nm for the link 01, and 157.0 Nm for the link 02. The magnitudes of the control signals were: 30.0 Nm for the PD control, 15.0 Nm for the PD+MBT control, and 15.0 Nm for the PD+FM control.

It should be noted that the PD+FM controller with low cost mapping can require a lower computational load for implementation in comparison to the PD+MBT controller as well as the PD+FM controller with accurate mapping.

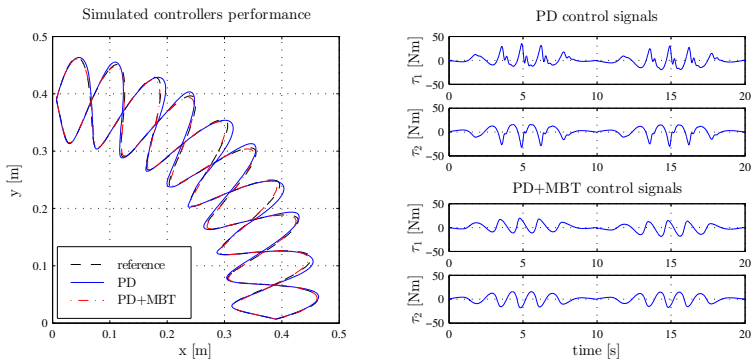

Figure 14. Simulation results: trajectory tracking performance and control signals - PD and PD+MBT controllers

7. Experimental results

In this section, details of the experiment are described. The PD, PD+MBT, and PD+FM (low cost mapping) controllers simulated numerically are implemented for the XOberon operational system (Nikitin, 1998; Brega, 1998; Wirth and Gutnecht, 1992; Reiser and Wirth, 1992; Reiser, 1991) to control an industrial SCARA robot manipulator whose model served as the basis for the simulations. In the practical experiment, basically the same controller parameters (Table 6.2) are used, including the sampling period (1.0 ms). The desired trajectory is the same as the one used in the simulations (Fig. 10).

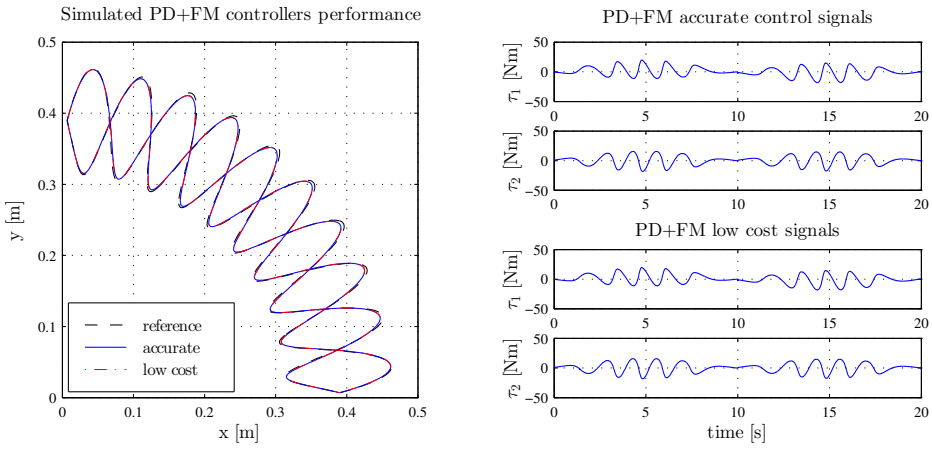

Figure 15. Simulation results: trajectory tracking performance and control signals - PD+FM (accurate mapping and low cost mapping) controllers

7.1. Control architecture

Fig. 16 shows the control architecture used for one servo axis (driver, motor, optical encoder). Each motor is connected to an electronic rack containing a VME-bus (VersaModular Eurocard bus) based CPU (Central Processing Unit, i.e., processor board Motorola MVME 2600-2). The optical encoder measures the instantaneous rotation angle of each motor. The analog output block is connected to each motor power amplifier (driver). The power amplifier is configured so that the current is proportional to the control voltage applied.

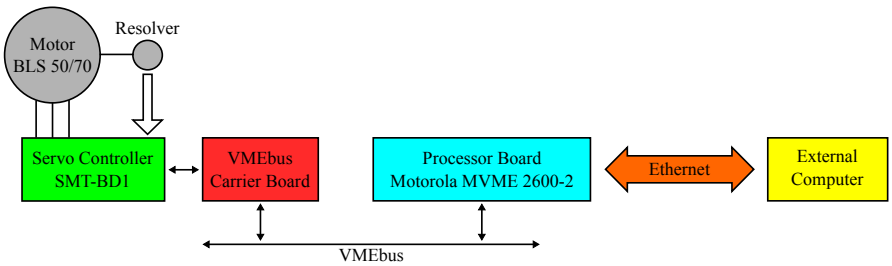

Figure 16. Control system of one servo axis

The controller (implemented at the CPU) uses the information of optical encoders to generate the control torque in the joint space. The control torque is amplified and sent to each motor through the drivers. The sampling frequency is 1 kHz. The controller is linked to an external computer using Ethernet, as

shown in Fig. 16. The real time version of the operational system and software programming is based on XOberon.

The robot manipulator runs a version of XOberon operational system over a PowerPC 200 MHz equipped with 16 MB of memory (electronic rack), which communicates with its I/O device using an industrial VME bus (67 MHz). The user develops the whole control system of the robot manipulator (including text command interface with the user, initialization and security functions) and through a cross compiler the execution program is downloaded to the CPU of this robot manipulator (up to 4 MB of code).

7.2. PD and PD+MBT control strategies performance

Based on the experiments, in Fig. 17 one can observe that the PD+MBT controller presents a considerable reduction of tracking errors, compared with the PD controller. From the information given for the trajectory tracking performance, see Fig. 18 (left), one can verify that the PD controller shows the quadratic error of 6.01×10^{-3} m with variance of 15.05×10^{-6} and the PD+MBT controller shows the error of 3.77×10^{-3} m with variance of 3.48×10^{-6}. That is, the PD+MBT controller features quadratic error about two times smaller and variance five times smaller compared with the PD controller.

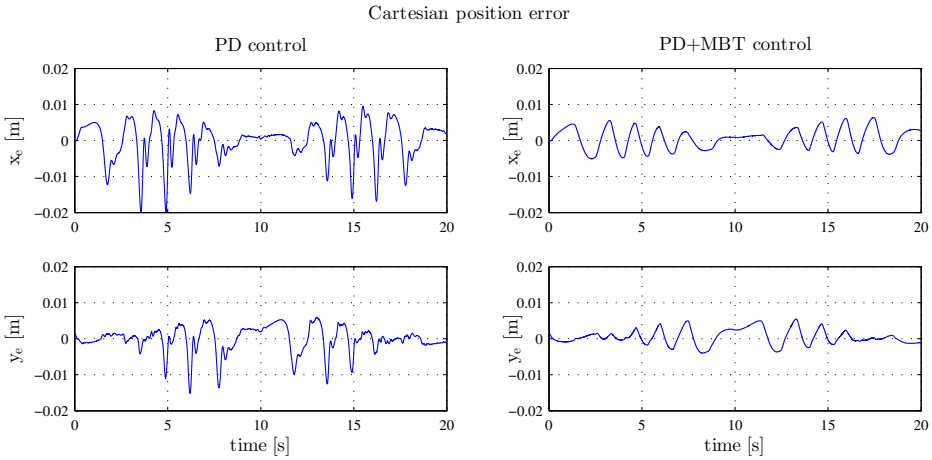

Figure 17. Experimental results: trajectory tracking errors - PD and PD+MBT controllers

Another important feature to be mentioned is the difference between control actions, as shown in Fig. 18 (right). The magnitude of the control action is reduced for the PD+MBT controller once tracking errors are reduced by inclusion of the model-based terms in the control law.

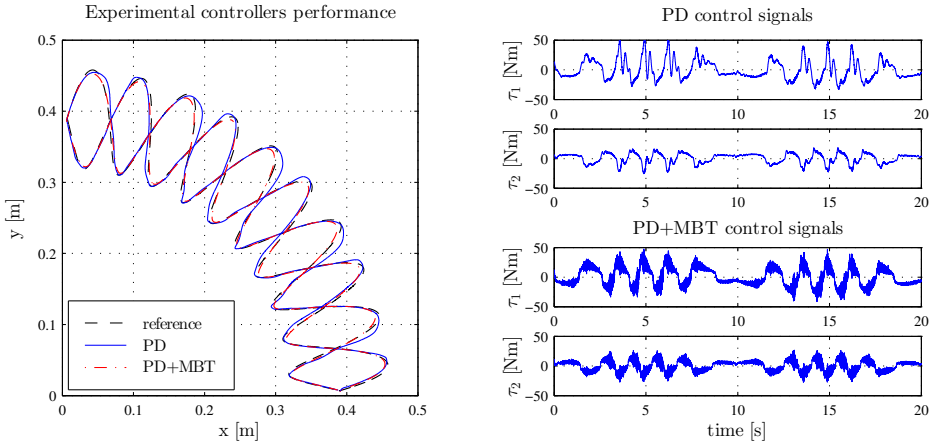

Figure 18. Experimental results: position control performance and control signals - PD and PD+MBT controllers

7.3. PD+FM control strategy - computational load reduction

As the last result, the performance of the PD+FM controller with low cost mapping is shown in Fig. 19. Compared to the desired trajectory, the calculated quadratic error is 3.85×10^{-3} and its variance is 3.40×10^{-6}. This validates the mapping process as a good choice to maintain the performance of the controller with a lower computational load.

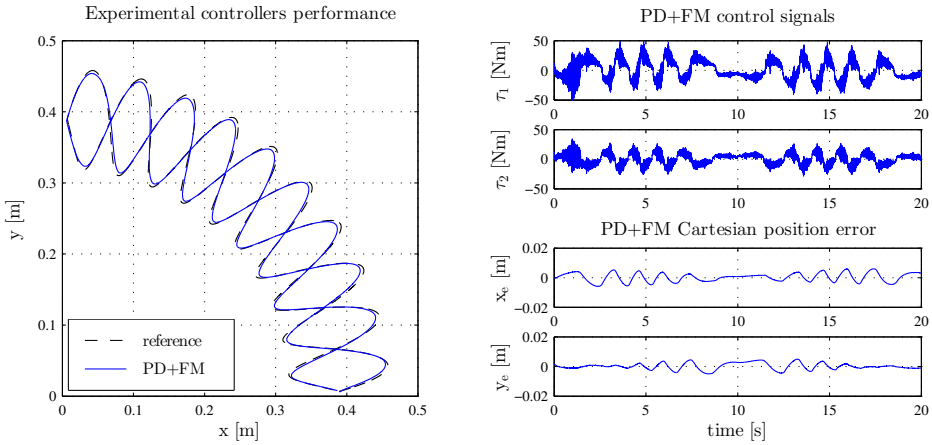

Figure 19. Experimental results: trajectory tracking performance, control signals and trajectory tracking errors - PD+FM controller (low cost mapping)

It is important to emphasize that not all parameters of the robot manipulator

used in the simulations are exact. Besides, frictions, flexibility, nonlinearities, actuator dynamics, unmodeled dynamics and noise, given by the practical setup are not considered in simulations. Thus, as expected, results from the practical implementation are different from the simulation, as verified in Figs. 12 through 19. Observing the control signals in Figs. 14, 15, 18, and 19, one sees that the noise in the practical implementation implies larger variations in the control signals. This noise in the control signal comes from the indirect joint velocity measurement, which is obtained by differentiating joint position. The PD+FM (low cost mapping) and PD+MBT control strategies amplify even more this noise on the control action when compared with the PD control strategy (see Appendix 10).

8. Conclusions

In this paper a stiffness control strategy was proposed with model-based nonlinear terms mapped via fuzzy systems, where the task and the control were performed in the operational space.

The proposed approach is based on a well known model-based stiffness control strategy that requires more processing power as complex as the robot manipulator model. To overcome this drawback a fuzzy mapping was used to accomplish the exchange of the processing power requirement by memory.

To circumvent the computational load problem, a mapping of the nonlinear terms of the controller was proposed and experimental tests were conducted on a SCARA configuration robot manipulator. The mapping approach allowed for a significant reduction of computational load without significant loss of performance. Also, simulation results showed the ability of the controller to follow a desired trajectory. The aim of these tests was to demonstrate the capabilities of the control method and its viability for practical use.

The results confirm that the theoretical development as well as the performance obtained were in the expected range of accuracy, even with results of the practical implementation being slightly different from the simulation results due to aforementioned reasons.

The tuning of the controllers, in simulations and practical implementations, was done empirically. Probably the best possible performance (response time, overshoot, etc.) was not reached. To overcome this deficiency, further studies are necessary on techniques for automatic tuning that give optimum values for controller parameters. Nevertheless, regarding the authors' experience, the trial-and-error tuning is adequate for producing performance with acceptable accuracy. As a future work, the aim is to circumvent the effect or influence of noise incident on the control signals.

References

Brega, R. (1998) A real-time operating system designed for predictability and run-time safety. Inst. of Robotics, Swiss Federal Institute of Technology.

Garcia, A., De Pieri, E.R. and Guenther, 4. (2002) Experimental study applied to an industrial robot by using variable structure controllers and friction compensation. *Journal of the Brazilian Society of Mechanical Sciences*, **24**,4, 302–308.

Hüipi, R. and Gruener, G. (2001) Software documentation for the SCARA robot Inter. Institute of Robotics, Swiss Federal Institute of Technology.

Khalil, W. and Dombre, E. (2004) *Modeling, Identification and Control of Robots*. 1st ed., Butterworth-Heinemann, London, UK.

Lewis, F.L., Dawson, D.M. and Abdallah, C.T. (2004) *Robot Manipulator Control Theory and Practice*. 2nd ed., Marcel Dekker, NY, USA.

Lilly, J.H. (2010) *Fuzzy Control and Identification*. 1st ed., John Wiley and Sons, NJ, USA.

Mamdani, E.H. (1974) Application of fuzzy algorithms for control of simple dynamic plant. *Procedings of IEEE*, **121**,12, 1585–1588.

Matlab (2012) Fuzzy Logic Toolbox Users Guide R2012a Matlab®. Technical report. The MathWorks, Inc.

Mendes, M.F., Kraus, Jr,W.and De Pieri, E.R. (2002) Variable structure position control of an industrial robotic manipulator. *Journal of the Brazilian Society of Mechanical Sciences*, **24**, 169–176.

Natale, C. (2010) *Interaction Control of Robot Manipulators: Six-Degrees-of-Freedom Tasks*. 1st ed., Springer, NJ, USA.

Nikitin, E. (1998) *Into the Realm of Oberon: An Introduction to Programming and the Oberon-2 Programming Language*. 1st ed., Springer, NY, USA.

Passold, F. (2009) Applying RBF neural nets for position control of an Inter SCARA robot. *International Journal of Computers, Communications and Control*, **4**,2, 148–157.

Piegat, A. (2010) *Fuzzy Modeling and Control*. 1st ed., Springer, NY, USA.

Reiser, M. (1991) *The Oberon System: User Guide and Programmer's Manual*, 1st ed., Addison-Wesley, NY, USA.

Reiser, M. and Wirth, N. (1992) *Programming in Oberon: steps beyond Pascal and Modula*. Addison-Wesley ACM Press, NY, USA.

Sciavicco, L. and Siciliano, B. (2009) *Modelling and Control of Robot Manipulators*. 2nd ed., Springer, London.

Siciliano, B. and Khatib, O. (2008) *Springer Handbook of Robotics*. Springer, Berlin, Heidelberg.

Siciliano, B., Sciavicco, L., Villani, L. and Oriolo, G. (2011) *Robotics: Modelling, Planning and Control*. 1st ed., Springer, London.

Simulink (2012) Simulink® Getting Started Guide R2012a Matlab® /Simulink®. Technical report. The MathWorks, Inc.

Slotine, J.J.E. and Li, W. (1987) On the adaptative control of robot manipulators. *Int. J. Rob. Res.*, **6**, 49–59.

Slotine, J.J.E. and Li, W. (2004) *Applied Nonlinear Control*. China Machine Press, Beijing.

Takagi, T. and Sugeno, M. (1985) Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems,*

Man, and Cybernetics, **15**,(1), 116–132.

Tanaka, K. and Wang, H.O. (2001) *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*. John Wiley and Sons, NY, USA.

Tischler, N. (2000) *Experimental Investigation of Stiffness Control for a Robotic Manipulator*. University of Toronto.

Vargas, F.J.T. (2005) *Analysis and synthesis of force-position controllers of robot manipulators: theoretical and experimental aspects*. Ph.D. thesis, Federal University of Santa Catarina.

Vargas, F.J.T., De Pieri, E.R. and Castelan, E.B. (2004) Identification and friction compensation for an industrial robot using two degrees of freedom controllers. In: *Proceedings of the 8th International Conference on Control, Automation, Robotics and Vision*. IEEE, 1146–1151.

Weihmann, L. (1999) *Description, installation, programming and operation of a SCARA type robot manipulator*. Master's thesis, Federal University of Santa Catarina.

Wirth, N. and Gutnecht, J. (1992) *Project Oberon - the Design of an Operating System and Compiler*. Addison-Wesley ACM Press, NY.

Yu, B (2000) *Modeling, control design and mechatronic implementation of constrained robots for surface finishing applications*. Ph.D. thesis, Oklahoma State University.

Zadeh, L.A. (1973) Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man and Cybernetics*, **SMC-3**, **1**, 28–44.

Zhang, H. and Liu, D. (2008) *Fuzzy Modeling and Fuzzy Control*, 1st ed., Springer, NY, USA.

Appendices

9. Mapping results

Table 6. J^T fuzzy rules setup and calculated quadratic errors

Setup parameters					
Element		Input MF	Parameters [a, b, c]	Output MF	Value
$\overline{\overline{J^T}}_{11}$	rules	$trimf(angle)$	$[-0.7853, 0.0045, 0.8977]$	$constant$	$[\ 0.0056]$
		$trimf(angle)$	$[-0.0098, 0.7331, 1.6849]$	$constant$	$[-0.3211]$
		$trimf(angle)$	$[\ 0.9199, 1.3722, 2.3533]$	$constant$	$[-0.4195]$
$\overline{\overline{J^T}}_{12}$		$trimf(angle)$	$[-0.7826, 0.1985, 0.6508]$	$constant$	$[\ 0.4195]$
		$trimf(angle)$	$[-0.1141, 0.8376, 1.5806]$	$constant$	$[\ 0.3211]$
		$trimf(angle)$	$[\ 0.6730, 1.5662, 2.3561]$	$constant$	$[-0.0056]$
$\overline{\overline{J^T}}_{21}$		$trimf(angle)$	$[-0.7853, 0.0425, 0.8146]$	$constant$	$[\ 0.1497]$
		$trimf(angle)$	$[-0.0414, 0.7696, 1.6326]$	$constant$	$[-0.0514]$
		$trimf(angle)$	$[\ 0.8126, 1.4882, 2.3559]$	$constant$	$[-0.2131]$
$\overline{\overline{J^T}}_{22}$		$trimf(angle)$	$[-0.7853, -0.1079, 0.8613]$	$constant$	$[0.2062]$
		$trimf(angle)$	$[\ 0.0157,\ \ 0.8389, 1.5607]$	$constant$	$[0.2749]$
		$trimf(angle)$	$[\ 0.4486,\ \ 1.6401, 2.3561]$	$constant$	$[0.1416]$

Table 7. BJ_{11}^{-1} accurate setup fuzzy mapping

Input MF type		Input MF	Parameters [a, b, c]
$gbellmf(angle)$		a_1	$[0.3975, 1.9997, 0.0036]$
		a_2	$[0.3963, 1.9994, 0.7857]$
		a_3	$[0.3978, 1.9997, 1.5670]$
$gbellmf(radius)$		r_1	$[0.0749, 2.0000, 0.3178]$
		r_2	$[0.0854, 2.0002, 0.4194]$
		r_3	$[0.0141, 1.9995, 0.5366]$
Output MF Type	Rule No.	Rule	Parameters
linear	1	$[a_1\ r_1]$	$[-7.2222, -33.4583, 5.1862]$
	2	$[a_1\ r_2]$	$[-6.2989, -54.0797, 18.2109]$
	3	$[a_1\ r_3]$	$[137.2256, -337.0226, -422.3745]$
	4	$[a_2\ r_1]$	$[-4.3441, -23.4850, 0.6265]$
	5	$[a_2\ r_2]$	$[-1.7401, -42.2961, 9.7199]$
	6	$[a_2\ r_3]$	$[487.2411, -392.2798, -599.1970]$
	7	$[a_3\ r_1]$	$[-0.0675, 0.8271, -9.4073]$
	8	$[a_3\ r_2]$	$[3.5014, -6.6289, -11.7076]$
	9	$[a_3\ r_3]$	$[637.8705, -412.8825, -803.6321]$

Table 8. BJ_{12}^{-1} accurate setup fuzzy mapping

Input MF type		Input MF	Parameters [a, b, c]
$gbellmf(angle)$		a_1	[0.3934, 1.9999, 0.0005]
		a_2	[0.3930, 1.9999, 0.7851]
		a_3	[0.3932, 1.9999, 1.5703]
$gbellmf(radius)$		r_1	[0.0683, 1.9999, 0.3107]
		r_2	[0.0815, 2.0001, 0.4245]
		r_3	[0.0137, 1.9995, 0.5417]
Output MF Type	Rule No.	Rule	Parameters
linear	1	$[a_1\ r_1]$	[−4.8050, −2.7552, 9.9436]
	2	$[a_1\ r_2]$	[−7.7183, 2.8306, 7.9756]
	3	$[a_1\ r_3]$	[−1163.7192, −14.0762, 30.0290]
	4	$[a_2\ r_1]$	[−8.05774, −17.3037, 15.0041]
	5	$[a_2\ r_2]$	[−10.3594, −23.6060, 20.8033]
	6	$[a_2\ r_3]$	[−946.1839, −22.6433, 14.0975]
	7	$[a_3\ r_1]$	[−8.5493, −22.1582, 15.9287]
	8	$[a_3\ r_2]$	[−9.5871, −37.2410, 25.5024]
	9	$[a_3\ r_3]$	[−360.2554, −230.7200, −410.0439]

Table 9. BJ_{21}^{-1} accurate setup fuzzy mapping

Input MF type		Input MF	Parameters [a, b, c]
$gbellmf(angle)$		a_1	[0.3934, 1.9999, 0.0004]
		a_2	[0.3931, 1.9999, 0.7852]
		a_3	[0.3934, 1.9999, 1.5702]
$gbellmf(radius)$		r_1	[0.0664, 1.9999, 0.3088]
		r_2	[0.0796, 2.0001, 0.4240]
		r_3	[0.0125, 1.9995, 0.5394]
Output MF Type	Rule No.	Rule	Parameters
linear	1	$[a_1\ r_1]$	[−2.8632, 8.7167, 2.7256]
	2	$[a_1\ r_2]$	[−4.1807, 15.6978, −1.1295]
	3	$[a_1\ r_3]$	[−184.9605, 326.5142, 679.2908]
	4	$[a_2\ r_1]$	[−4.5627, 2.4389, 5.1260]
	5	$[a_2\ r_2]$	[−6.0324, 7.1068, 3.6007]
	6	$[a_2\ r_3]$	[−619.3488, 397.4153, 873.6399]
	7	$[a_3\ r_1]$	[−4.6154, −5.6079, 6.8668]
	8	$[a_3\ r_2]$	[−6.0396, −5.5412, 8.9980]
	9	$[a_3\ r_3]$	[−693.3766, 434.8291, 883.0551]

Table 10. BJ_{22}^{-1} accurate setup fuzzy mapping

Input MF type		Input MF	Parameters [a, b, c]
$gbellmf(angle)$		a_1	[0.3951, 1.9998, 0.0018]
		a_2	[0.3944, 1.9997, 0.7855]
		a_3	[0.3951, 1.9998, 1.5690]
$gbellmf(radius)$		r_1	[0.0659, 1.9999, 0.3083]
		r_2	[0.0794, 2.0001, 0.4237]
		r_3	[0.0120, 1.9996, 0.5390]
Output MF Type	Rule No.	Rule	Parameters
linear	1	$[a_1\ r_1]$	[3.7161, 6.2729, 0.1971]
	2	$[a_1\ r_2]$	[4.4040, 6.7177, −0.1013]
	3	$[a_1\ r_3]$	[875.7292, −3.2664, 4.4106]
	4	$[a_2\ r_1]$	[1.9610, 11.0823, 0.2959]
	5	$[a_2\ r_2]$	[1.9712, 15.9162, −2.5474]
	6	$[a_2\ r_3]$	[684.4734, 6.4300, 110.7776]
	7	$[a_3\ r_1]$	[−0.4756, 9.4954, 3.2384]
	8	$[a_3\ r_2]$	[−1.0836, 16.3792, 0.2437]
	9	$[a_3\ r_3]$	[223.9125, 219.9400, 483.5207]

Table 11. BJ_{11}^{-1} low cost setup fuzzy mapping

Input MF type		Input MF	Parameters [a, b, c]
$trimf(angle)$		a_1	[−0.7853, −0.0015, 0.7879]
		a_2	[0.0011, 0.7846, 1.5702]
		a_3	[0.7844, 1.5716, 2.3561]
$trimf(radius)$		r_1	[0.0980, 0.4314, 0.4919]
		r_2	[0.2434, 0.5454, 0.6949]
Output MF Type	Rule No.	Rule	Parameters
constant	1	$[a_1\ r_1]$	[0.4878]
	2	$[a_1\ r_2]$	[−15.5197]
	3	$[a_2\ r_1]$	[−6.1488]
	4	$[a_2\ r_2]$	[−17.8846]
	5	$[a_3\ r_1]$	[−9.2307]
	6	$[a_3\ r_2]$	[−10.1661]

Table 12. BJ_{12}^{-1} low cost setup fuzzy mapping

Input MF type		Input MF	Parameters [a, b, c]
$trimf(angle)$		a_1	$[-1.5707, 0.0044, 1.5670]$
		a_2	$[-0.0043, 1.5751, 3.1415]$
$trimf(radius)$		r_1	$[0.0978, 0.4323, 0.4919]$
		r_2	$[0.2446, 0.5457, 0.6949]$
Output MF Type	Rule No.	Rule	Parameters
constant	1	$[a_1\ r_1]$	[10.2577]
	2	$[a_1\ r_2]$	[9.3243]
	3	$[a_2\ r_1]$	[1.6012]
	4	$[a_2\ r_2]$	$[-15.8773]$

Table 13. BJ_{21}^{-1} low cost setup fuzzy mapping

Input MF type		Input MF	Parameters [a, b, c]
$trimf(angle)$		a_1	$[-1.5707, 0.0129, 1.5594]$
		a_2	$[-0.0125, 1.5835, 3.1415]$
$trimf(radius)$		r_1	$[0.0999, 0.4535, 0.4897]$
		r_2	$[0.2343, 0.5475, 0.6949]$
Output MF Type	Rule No.	Rule	Parameters
constant	1	$[a_1\ r_1]$	[2.1462]
	2	$[a_1\ r_2]$	[12.8496]
	3	$[a_2\ r_1]$	$[-2.0857]$
	4	$[a_2\ r_2]$	$[-2.7350]$

Table 14. BJ_{22}^{-1} low cost setup fuzzy mapping

Input MF type		Input MF	Parameters [a, b, c]
$trimf(angle)$		a_1	$[-1.5707, -0.0104, 1.5811]$
		a_2	$[0.0086, 1.5602, 3.1415]$
$trimf(radius)$		r_1	$[0.0984, 0.4438, 0.4920]$
		r_2	$[0.2413, 0.5470, 0.6949]$
Output MF Type	Rule No.	Rule	Parameters
constant	1	$[a_1\ r_1]$	[2.5506]
	2	$[a_1\ r_2]$	[5.3972]
	3	$[a_2\ r_1]$	[2.8141]
	4	$[a_2\ r_2]$	[13.4410]

10. Noise effect in the PD+MBT control action

Consider the indirect measurement of end-effector velocity, described as:

$$\hat{\dot{x}} = \dot{x} + \delta, \tag{22}$$

where $\dot{\mathbf{x}}$ represents the real end-effector velocity and δ the noise signal in the measurement process.

By inserting 22 into the PD control action in the operational space, the control torque due to noise for the PD control strategy can be calculated as:

$$\tau_{PD}^{\delta} = J^T k_d \delta. \tag{23}$$

Likewise, substituting (22) in (10), the control torque due to noise for PD+MBT control strategy can be calculated as:

$$\tau_{MBT}^{\delta} = J^T (\bar{B} k_p + k_d) \delta. \tag{24}$$

Therefore, the PD+MBT control strategy leads to a more noisy control action when compared to the PD control strategy due to the $J\bar{B}k_p\delta$ term. This statement is also valid for the PD+FM (accurate mapping and low cost mapping) control strategy.