Simulation Hybrid Fuzzy Control of SCARA Robot

MARIUS-CONSTANTIN POPESCU, ILIE BORCOŞI, ONISIFOR OLARU, LUMINIȚA POPESCU, FLORIN GROFU
Department of Electromechanics
University of Craiova
str. A. I. Cuza nr.13, Craiova
Department of Automatics and Informatics
University of C-tin Brancusi
str. Geneva nr.3, Tg-Jiu, Gorj
ROMANIA

http://www.em.ucv.ro, http://www.utgjiu.ro/ing

Abstract: - This paper presents the simulation of a Hybrid Fuzzy Controller suitable for industrial applications. The developed Hybrid Fuzzy Controller consists of a weighted combination of the Direct Fuzzy Controller and Indirect Fuzzy Controller and a gradually activated supervisory controller. The unique feature of the Hybrid Fuzzy Controller is that no mathematical models of the plant are required and the proposed controller is able to adaptively estimate the bound functions on-line, which are required for the determination of the supervisory controller. The supervisor controller in the sense that all signals are bounded guarantees global stability of the closed-loop system. The approach of rapid prototyping is employed to implement the Hybrid Fuzzy Controller so as to control a Selectively Compliance Assembly Robot Arm. Simulink, an interactive graphical software for simulating dynamic systems, is used to model, simulate and analyse the dynamic system.

Key-Words: - SCARA Robot, Simulink, Hybrid Fuzzy Controller

1 Introduction

The non-linear dynamics governing robot motion presents a challenging control problem in the field of control engineering. As the manipulators are usually used to perform high-precision tasks, robot controllers are required to control the motion of a robot effectively and accurately. However, as the manipulator is a multivariable non-linear coupling dynamic system with uncertainties, it is difficult to obtain an accurate mathematical model so that classical or modern control laws can be applied. To accommodate system uncertainties and variations, learning or adaptive techniques must be incorporated. While conventional controllers cannot effectively control the motion of a robot due to complications of these nonlinear effects, fuzzy logic, however, could offer a promising approach to circumvent the deficiency. This is because in contrast with conventional controllers, fuzzy logic allows one to formulate the control problem of a plant in terms of linguistic rules drawn from the behaviour of a 'human operator' instead of an algorithm synthesised from a rigorous mathematical model of the plant. In this paper, a Hybrid Fuzzy Controller (HFC) suitable for real-time industrial applications is designed, developed and simulation. The HFC is made up of two on-line adaptive fuzzy controllers, namely Direct Fuzzy Controller (DFC) and Indirect Fuzzy Controller (IFC) with adaptive laws that can guarantee global stability of the closed-loop system. In IFC, the dynamics of the robot manipulator are estimated on-line and the controller is chosen assuming that the estimated dynamics represents the true dynamics of the system. In AFC, the parameters of the controller are adjusted directly in order to reduce some norm of the output error between the plant and the reference model. The HFC is able to compensate for environmental variations such as payload mass and disturbance torque during the operation process. It is found to be superior to conventional adaptive controllers primarily because boundaries of uncertainties are being adaptively estimated. However, one major drawback of the HFC is that it requires massive computing resources and speeds crucial for its implementation. Therefore, the main challenge for real-time execution of this controller is to minimise the amount of processing that need to be performed by the controller before the next input data arrives. The approach of rapid prototyping is employed to implement the HFC so as to control a Selectively Compliance Assembly Robot Arm (SCARA) Robot in real time. Simulink, interactive graphical software for simulating dynamic systems, is used to model, simulate and analyse the dynamic system.

2 Problem Formulation

2.1. Background on the SCARA robot

Each of the four axes provides a different motion and contributes to one degree of freedom of the robot (see Fig. 1). The basic SCARA geometry is realised by arranging two revolute joints (T1 and T2 axes) and one prismatic joint (Z-axis) in such a way that all axes of motion are parallel. The acronym SCARA characterises the mechanical features of a structure offering high stiffness to vertical loads and compliance to horizontal loads. Motion control is implemented only for axes Z, T1 and T2 in this work, which are designated as Joint 1, 2 and 3 respectively.

2.2. The robot dynamic model

The dynamic equations of a rigid body robot are a set of highly non-linear coupled differential equations. The dynamics (of a nth joint robot arm) is generally expressed in the form of [1], [4]:

$$\mathbf{M}(\theta)\ddot{\theta}+\mathbf{C}(\dot{\theta},\ddot{\theta})\dot{\theta}+\mathbf{G}(\theta)+\mathbf{F}_V\dot{\theta}+\mathbf{F}_C=\tau,$$

where $\mathbf{M}(\theta)$ is the $n\times n$ inertia matrix of the manipulator, $\mathbf{C}(\dot{\theta},\ddot{\theta})$ is the $n\times n$ vector of centrifugal and Coriolis terms, $\mathbf{G}(\theta)$ is an $n\times 1$ vector of gravity terms, $\mathbf{F}_V$ is the $n\times 1$ vector of viscous friction terms, $\mathbf{F}_C$ is the $n\times 1$ vector of coulomb terms and τ is the $n\times 1$ vector of the input torque (generated by the joint motor). Each element of $\mathbf{M}(\theta)$ and $\mathbf{G}(\theta)$ is a complicated function which depends on θ, angular positions of all joints of the manipulator. Similarly, each element of $\mathbf{C}(\dot{\theta},\ddot{\theta})$ is a complicated function of both θ and $\dot{\theta}$. The movement of the end-effector in a desired trajectory, at a particular velocity, thus requires a complex set of torque functions to be applied to the actuating system of the robot at each joint. The dynamic model of the SCARA robot arm has been developed in [2], [3], with most of its parameters determined and verified through experiments. Based on this known mathematical model of the robot, simulation analysis and design of the HFC were carried out using MATLAB simulation tools.

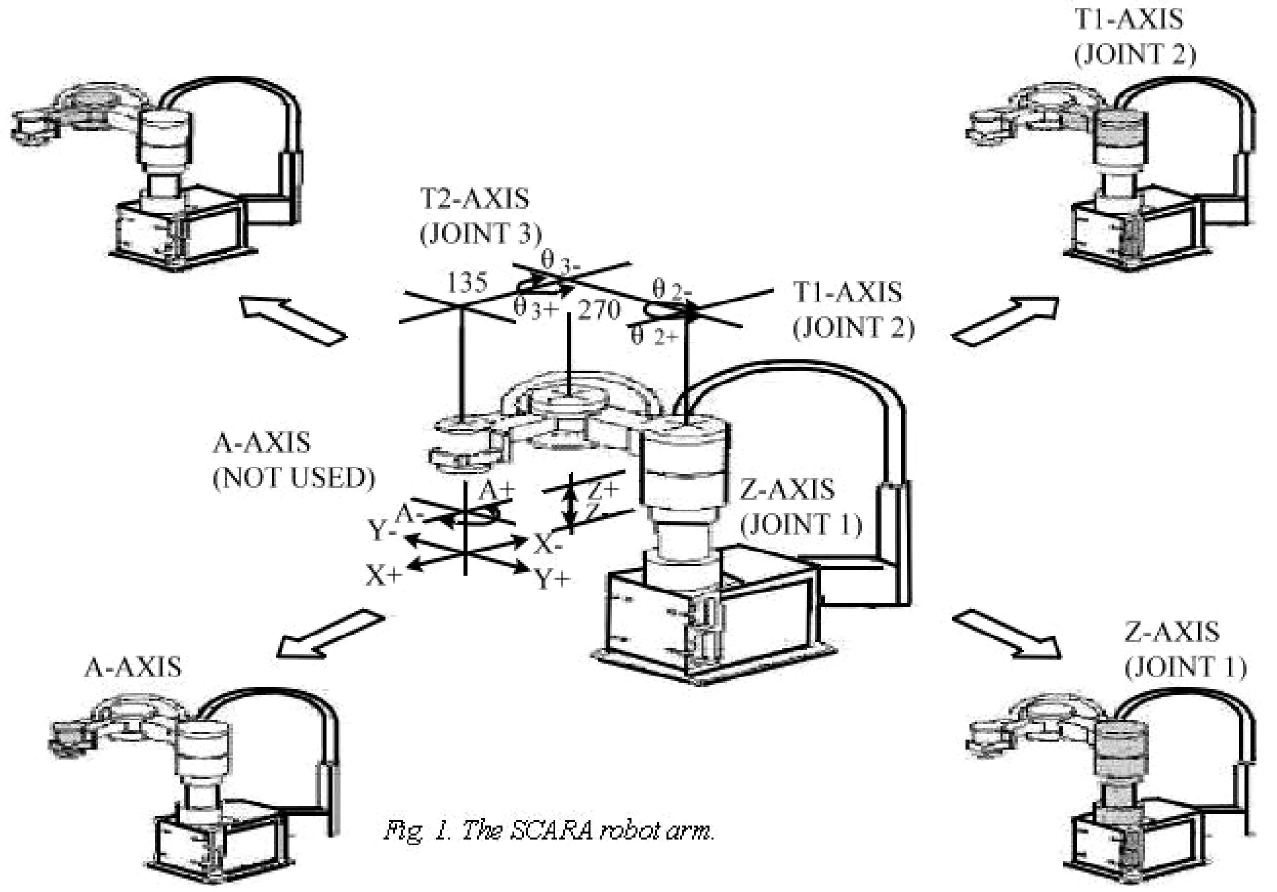


Fig. 1. The SCARA robot arm.

2.3. Motion control

A two-joint control structure is used in this work where the motion of the *T1*-axis and *T2*-axis will be controlled simultaneously. In this scheme, the Z-axis will not be controlled since the earlier work is based on a two-joint controller. The dynamics of the two joints can be represented by a multiple-input multiple-output (MIMO) system. The degree of difficulty will increase with the number of joints moving together.

3 Problem Solution

3.1. Hybrid fuzzy controller

The proposed control algorithm is given by [5]:

$$\tau = \alpha\tau_1 + (1-\alpha)\tau_D + \tau_S , \qquad (1)$$

where τ_1 is the control torque contributed by the IFC, τ_D is the control torque contributed by the DFC, τ_S is the supervisory controller and $0 \le \alpha \le 1$ is a weighting factor. If fuzzy control rules are more important and reliable than fuzzy descriptions, choose α to be small; otherwise, choose α to be large. The resulting certainty equivalent controller is:

$$\tau_I = \hat{G}^{-1}(\theta/\Phi) - [\hat{F}(\theta,\dot{\theta}/\Phi) + \ddot{\theta}_M + K_V\dot{E} + K_P E], \qquad (2)$$

$$\ddot{\theta} = \mathbf{F}(\dot{\theta},\ddot{\theta}) + \mathbf{G}(\theta)\,\tau_D(\theta/\Phi), \qquad (3)$$

where $\tau_D(\theta/\Phi)$, $\hat{F}(\theta,\dot{\theta}/\Phi)$ and $\hat{G}(\theta/\Phi)$ assume the following form:

$$f(\bar{x}) = \frac{\sum_{l=1}^{M} y^l \left(\prod_{i=1}^{n} e^{-\left(\frac{x_i - x_i^l}{\sigma_i^l}\right)^2} \right)}{\sum_{l=1}^{M} \left(\prod_{i=1}^{n} e^{-\left(\frac{x_i - x_i^l}{\sigma_i^l}\right)^2} \right)}, \qquad (4)$$

where $f(\bar{x})$ is the fuzzy logic system with *Gaussian* membership function, centre average defuzzifier and product-inference rule, and y^l, x_i^l and σ_i^l are adjustable parameters. Now adding and subtracting $\mathbf{G}(\theta)\ \tau$ to Eq. (3) and after some manipulations, we obtain the closed-loop error equation due to the DFC which is given as:

$$\ddot{E} = -K_V\dot{E} - K_p E + \mathbf{G}(\theta)\left[\tau - \tau_D(\theta/\Phi)\right]. \qquad (5)$$

The closed-loop error dynamics due to the IFC becomes:

$$\ddot{E} = -K_v\dot{E} - K_p E + [\hat{F}(\theta,\dot{\theta}/\Phi) - \mathbf{F}(\dot{\theta},\ddot{\theta})] + [\hat{G}(\theta/\Phi) - \mathbf{G}(\theta)]\,\tau_I . \qquad (6)$$

Since the overall controller of Eq. (1) consists of a weighted combination of DFC and IFC, the respective weighted errors become:

$$(1-\alpha)\ddot{E} = -(1-\alpha)K_v\dot{E} - (1-\alpha)K_p E + (1-\alpha)\mathbf{G}\left[\tau - \tau_D(\theta/\Phi)\right] \qquad (7)$$

and

$$\alpha\ddot{E} = -\alpha K_v\dot{E} - \alpha K_p E + \alpha[\hat{F}(\theta,\dot{\theta}/\Phi) - \mathbf{F}(\dot{\theta},\ddot{\theta})] + \alpha[\hat{G}(\theta/\Phi) - \mathbf{G}(\theta)]\,\tau_I . \qquad (8)$$

The overall error governing the closed-loop system is given by the sum of Eqs. (7) and (8), i.e.

$$\ddot{E} = -K_V\dot{E} - K_p E + \alpha[\hat{F}(\theta,\dot{\theta}/\Phi) - \mathbf{F}(\dot{\theta},\ddot{\theta})] + \alpha[\hat{G}(\theta/\Phi)\,\tau_I + (1-\alpha)\mathbf{G}(\theta)\left[\tau - \tau_D(\theta/\Phi)\right]. \qquad (9)$$

The overall error dynamics in the state-space form is:

$$\dot{X} = \mathbf{AX} + \alpha\,\mathbf{B}_I[(\hat{F}(\theta,\dot{\theta}/\Phi) - \mathbf{F}(\dot{\theta},\ddot{\theta})) + (\hat{G}(\theta/\Phi) - \mathbf{G}(\theta)\,\tau_I] + (1-\alpha)\mathbf{B}_D\left[\tau - \tau_D(\theta/\Phi)\right] = \mathbf{AX} + \mathbf{BU}, \qquad (10)$$

where $\mathbf{X} = (e_1, \dot{e}_1, e_2, \cdots, \dot{e}_n)^T$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots \\ -K_{p1} & -K_{v1} & 0 & 0 & \cdots \\ 0 & 0 & 0 & 1 & \cdots \\ 0 & 0 & -K_{p2} & -K_{v2} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}, \quad \mathbf{B} = [\mathbf{B}_I\ \mathbf{B}_D]$$

$$\mathbf{B}_I = \begin{bmatrix} 0 & 0 & \cdots \\ 1 & 0 & \cdots \\ 0 & 0 & \cdots \\ 0 & 1 & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix}, \quad \mathbf{B}_D = \begin{bmatrix} 0 & 0 & \cdots \\ b_{11} & b_{12} & \cdots \\ 0 & 0 & \cdots \\ b_{21} & b_{22} & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix},$$

$$\begin{bmatrix} b_{11} & b_{12} & \cdots \\ b_{21} & b_{22} & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix} = \mathbf{G}(\theta) = \mathbf{M}^{-1}(\theta)$$

and

$$\mathbf{U} = \begin{bmatrix} \alpha[\hat{F}(\theta,\dot{\theta}/\Phi) - F(\theta,\dot{\theta})] + \alpha[\hat{G}(\theta/\Phi) - G(\theta)\tau_I] \\ (1-\alpha)[\tau - \tau_D(\theta/\Phi)] \end{bmatrix},$$

with K_p and K_v chosen such that $\mathbf{A}$ is a stable matrix, there exists a unique $2n \times 2n$ positive definite symmetric matrix, **P** which satisfies the Lyapunov equation:

$$\mathbf{A}^{T}\mathbf{P} + \mathbf{PA} = -\mathbf{Q}, \qquad (11)$$

where **Q** is an arbitrary $2n \times 2n$ positive definite matrix. Let the Lyapunov function candidate be:

$$V_e = \frac{1}{2} X^T \mathbf{P} X. \qquad (12)$$

Differentiating Eq. (12) with respect to time and using Eqs. (10) and (11) leads to:

$$\begin{aligned} V_e &= \frac{1}{2}\dot{X}^T \mathbf{P} X + \frac{1}{2} X^T \mathbf{P} \dot{X} = -\frac{1}{2} X^T \mathbf{Q} X \\ &+ \alpha X^T \mathbf{P}\mathbf{B}_I[(\hat{F}(\theta,\dot{\theta}/\Phi) - \mathbf{F}(\dot{\theta},\ddot{\theta})) + (\hat{G}(\theta/\Phi) \qquad (13) \\ &- \mathbf{G}(\theta)\tau_I] + (1-\alpha) X^T \mathbf{P}\mathbf{B}_D[\tau - \tau_D(\theta/\Phi)] . \end{aligned}$$

In order for the system to be bounded, we require $\dot{V}_e$ to be bounded which means that $\dot{V}_e < 0$ whenever V_e is greater than a constant $\ddot{V}$ specified by the designer. However, from Eq. (13), we see that it is difficult to design $\tau_D(\theta/\Phi)$ and $\tau_I(\theta/\Phi)$ such that the last two terms of Eq. (13) are less than zero. We can solve this problem by appending another control term $\tau_S(\theta)$ which is called supervisory control law. The final control law becomes:

$$\tau_F(\theta/\Phi) = \alpha\tau_I(\theta/\Phi) + (1-\alpha)\tau_D(\theta/\Phi) + \tau_S(\theta) \qquad (14)$$

3.2. Simulation model of HFC

The simulation program for the HFC is written in Simulink to facilitate Rapid Prototyping, a process that allows one to conceptualize solutions using block diagram modeling environment and predict the system performance prior to laying out hardware, writing any production software, or committing to a fixed design [8]. Fig. 2 shows the rapid prototyping process in greater details. The control system model and dynamics of the SCARA robot are created in Simulink. The first step in the design is to develop the SCARA robot model prior to algorithm development.

Once a simulation model, which is able to model the robot with sufficient accuracy, has been developed, the rapid prototyping process continues by using Simulink and toolboxes to develop the algorithm as well as analyse the results.

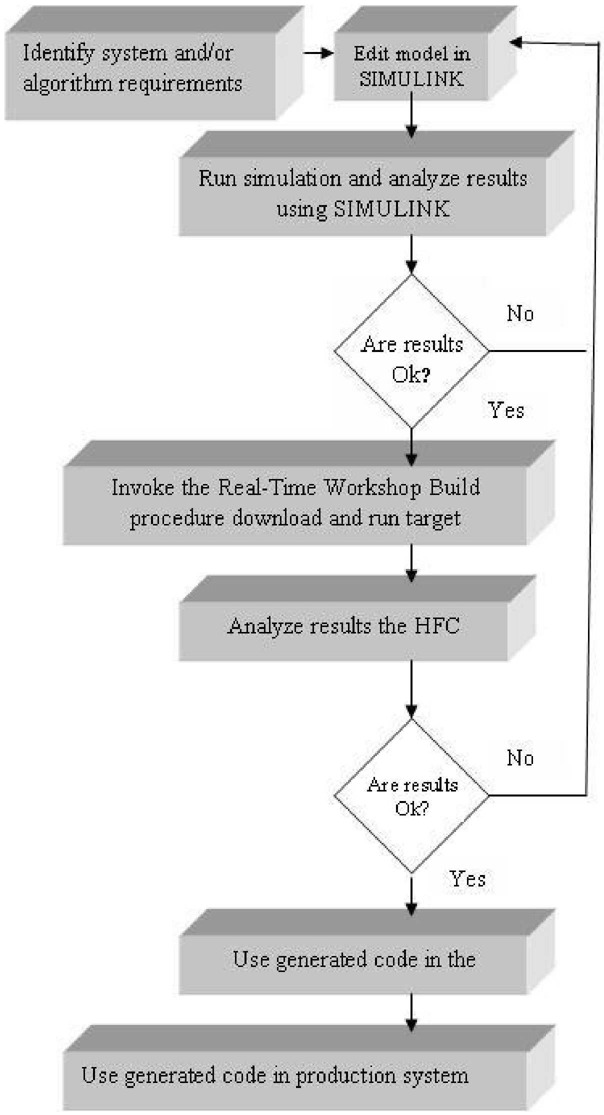


Fig. 2 Rapid prototyping process

If the results are not satisfactory, we iterate the modelling/analysis process until satisfactory results are obtained. Fig. 3 shows a Simulink diagram for simulating the HFC system for controlling Joint 2 (*T1*-axis) and Joint 3 (*T2*-axis) of the SCARA robot. This prototype is modelled closely based on the SCARA robot model and the HFC algorithm. The input variables to the HFC are *position error* and link *velocity* of T1 and T2 axis. The *output* variable is used to control the SCARA robot.

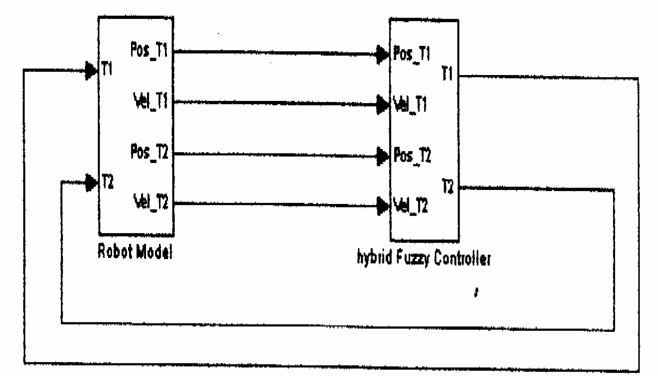
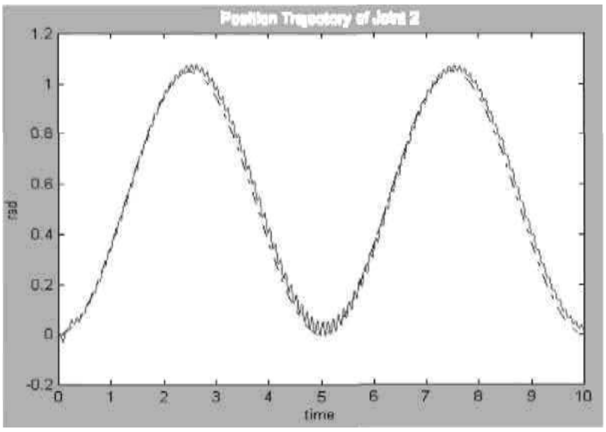


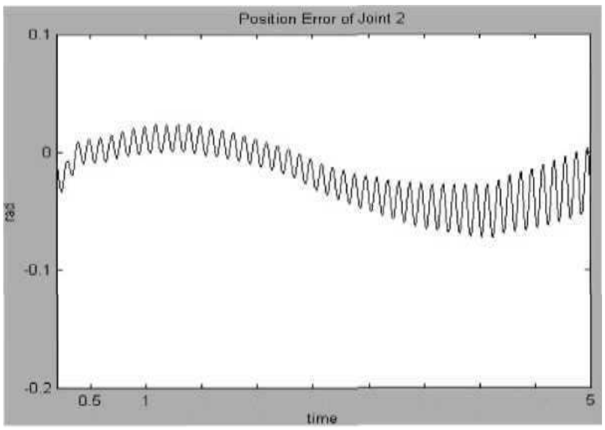
Fig. 3. Control system model in Simulink

3.3. Simulation results

To evaluate the performance of our HFC, a series of simulation runs were performed. The joint trajectory responses were plotted against time. Overshoots are kept strictly to zero and errors of less than 1 % recorded for all 2 joints.

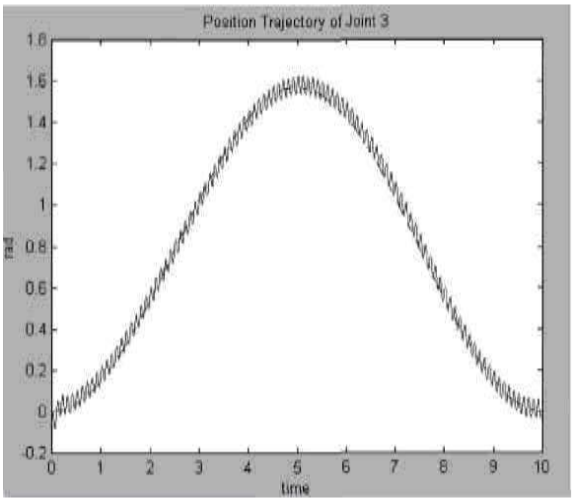


a)

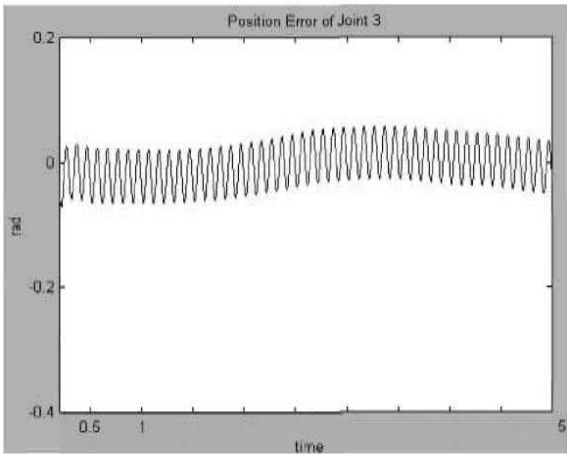


b)

Fig. 4. Position trajectory of joint 2, a; Position error of joint 2, b.



a)



b)

Fig. 5. Position trajectory of joint 3, a; Position error of joint 3, b.

Fig. 4a shows the position tracking trajectories and the corresponding position tracking errors for joint 2 using the HFC with α set to 0.95 i.e., there are joint contributions from the DFC and the IFC, with the IFC having 95% of the total contribution. Fig. 5a depicts those of joint 3. Desired position trajectories are indicated in dashed lines and actual trajectories after incorporating the HFC are indicated in solid lines. Figs. 4b and 5b show the position errors of joint 2 and joint 3 respectively. Both results show that overshoots are kept strictly to zero and errors of less than 1 % recorded for both joints. The HFC software consists of primarily 2 modules: (1) The main control module, and (2) The GUI for signal monitoring and parameters tuning. Module (1) is written in Simulink whereas Module (2) is developed using Control-Desk [1], [6]. The overall Simulink model for the HFC is shown in Fig. 6.

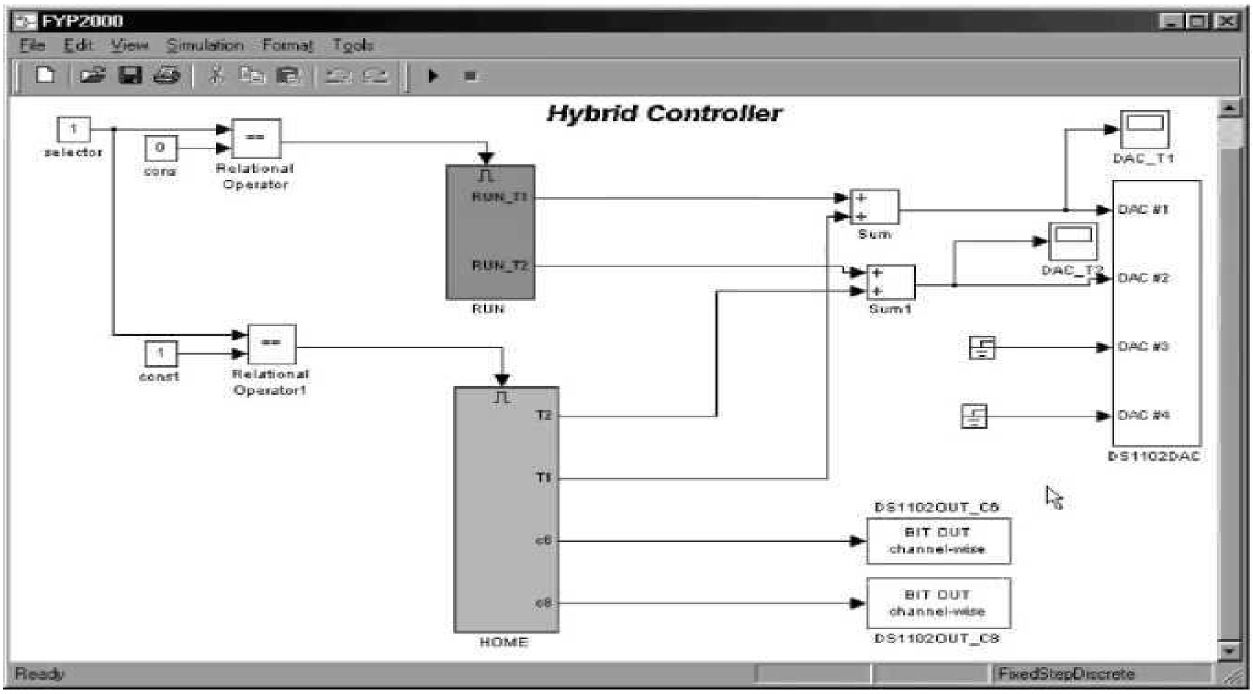


Fig. 6. Overall Simulink graphical models for the HFC

4 Conclusion

Once encouraging system performances are achieved, the Hybrid Adaptive Fuzzy Controller is implemented in real-time through Real-Time Workshop (RTW). The performance of the Hybrid Fuzzy Controller was found to be superior and it matches favourably the simulation results. A HFC suitable for real-time industrial applications has been successfully designed, developed and simulation in this work. The approach of rapid prototyping process was adopted to shorten the development time, thus reducing the cost of development. The amount of time saved is substantial especially when no rebuilding of codes is involved. The performance of the HFC was found to be superior and it matches favorably the simulation results. With the promising results being accomplished, the authors are confident that the proposed HFC will be very useful in many other real-time industrial applications [7].

References:

[1] Abdelkarim A., Terra Z., Implémentation d'un système de communication FM par l'utilisation du DSP TMS320VC5402, Editeur Ecole Nationale Polytechnique , 2005

[2] Nianzu Zhang, *Experimental Research on Robot Control*, M.Eng. thesis, Singapore, Nanyang Technological University, 1996.

[3] Meng Joo Er ' , Moo Teng Lim, Hui Song Lim, *Real-time hybrid adaptive fuzzy control of a SCARA robot,* Microprocessors and Microsystems 25, 2001.

[4] Onisifor O., Amplificatoare integrate in echipamente de automatizare, Editura Universitaria Craiova , 2003.

[5] Popescu M.C., *Estimarea şi identificarea proceselor*, Editura Sitech, Craiova, 2006.

[6] Popescu M.C, *Utilisation des ordinateurs*, Editura Universitaria, Craiova, 2004.

[7] www.ens-lyon.fr, *Texas Instruments TMS320C54.1*

[8] *** User Guide Matlab for Simulink