An Adaptive Controller for Multimodal Systems Based on Fuzzy Reference Model Generator

Sukumar Kamalasadan, *Member, IAENG*

Abstract— **This paper presents a novel adaptive controller for multimodal systems based on Fuzzy Reference Model Generator (FRMG). The proposed scheme consists of a fuzzy logic switching method working along with a Model Reference Adaptive Control (MRAC) framework. The fuzzy switching scheme produce a 'soft' way of generating the reference model, combining a group of weighted reference models at each plant operating point. Unlike static multiple model algorithms, this scheme provides an interactive multiple model environment with soft switching. Simulation results conducted on two practical examples of multimodal systems show that the scheme is computationally feasible, efficient and fault tolerant in nature.**

Index Terms—**Fuzzy Reference Model Generator, Model Reference Adaptive Controller, Multimodal Systems, Soft Switching.**

I. Introduction

Control of multimodal systems requires an intelligent controller which efficiently determines the system changes, keeps track of system uncertainties and takes appropriate control action in real time. Multimodal systems are divided as temporal or spatial ones. In the temporal multimodal systems, the mode transition is an arbitrary event, while in the spatial ones it depends on the systems auxiliary states or other derived variables [1]. Multiple model adaptive control is one of the effective solutions for controlling such systems. This approach was first initiated in [2]. The three main constituents of a Multiple Model Adaptive Control (MMAC) are multiple reference models, a switching scheme and a controller.

The research in this area branches out into two main directions. The first direction deals with mathematical MMAC based on deterministic and stochastic approaches and the second direction deals with newly emerging heuristic based MMAC mainly with fuzzy systems. Some of the relevant research efforts are as follows. In a stochastic mathematical MMAC representation, [3], [4] proposed a standard multiple model estimation algorithms; using standard probabilistic methods and Kalman filter residuals to assign conditional probabilities for each modeled hypotheses. However, these algorithms are often prone to numerical underflows. In a deterministic fashion, [5] proposed different switching and tuning schemes combining fixed and adaptive models. While [6] dealt with adaptive control of LTI discrete-time system using multiple models, [7], [8] proposed a stable multiple model approach to overcome the poor transient response of an adaptive controller. On the application side, reconfigurable flight control system using multiple model adaptive control methods is proposed in [9], and in [10] application of MMAC algorithm to control F-8c aircraft is discussed. This feasibility studies indicates that the approach is a reasonable candidate for aircraft adaptive control. In [11] a method which recovers the eigenvalues, and eigenvectors of the original closed loop system and controls the changing dynamic system under varied operating conditions is illustrated.

Reference [12] proposed a stable adaptive control using a Takagi-Sugeno fuzzy system, in which the adaptive schemes can 'learn' how to control the plant and achieve asymptotically stable tracking of reference input. In an attempt, [13] proposed a fuzzy model based adaptive control algorithms for a class of continuous-time nonlinear dynamic system and recently, an important consideration of using multiple fuzzy model generation including fault tolerant control has been proposed in [14]. It has been observed in [15] that the fuzzy model can be derived by tuning the fuzzy parameters. However, these tuned parameters cannot effectively develop the unknown changes online. Adaptive controller by fuzzy structure has been developed in [16, 17] which needs an online learning algorithm and effective tuning in order to provide a stable controller.

An important concept of changing the reference model structure instead of explicitly identifying the plant models is originated in [18-20]. In this technique, the reference models are changed along with the plant movement based on several offline studies. Further these reference models are utilized at different plant operation depending on certain heuristic switching scheme.

This paper present a heuristic based fuzzy reference model generation, which changes the reference model without implicit identification, and therefore the direct model reference adaptive control framework can be used even when the system shows multi-modality. In the proposed scheme, an Intelligent Supervisory Loop (ISL) is incorporated into the traditional MRAC framework in order to generate the plant reference model at every control interval. The paper is organized as follows: In section 2, the problem statement is presented, followed by the concept of multiple fuzzy reference model generation in section 3. Section 4 shows the concept behind the design of fuzzy logic scheme. Section 5 discusses the direct adaptive control laws and section 6 briefly explains the fault tolerant capability of the overall scheme. In section 7, an investigation of the proposed technique on a practical

Manuscript received May 31 2006.

Sukumar Kamalasadan is with the Department of Engineering and Computer Technology, The University of West Florida, Pensacola, FL 3251 USA (phone: 850-857-6451; fax: 850-474-2804; e-mail: skamalasadan@ uwf.edu).

application example is discussed, followed by conclusions in section 8.

II. PROBLEM STATEMENT

The theme behind the approach can be described using fig. 1. If the system under consideration can be modeled using one reference model, then the adaptive controller can be utilized to track the desired reference model output. It is well known that MRAC can effectively control the system with a mechanism adapting the control law based on the output error. However, if the reference model is rigid and the system output is away from the reference model output then the controller will be stressed and finally fail to control the plant. The main purpose of the intelligent module is to alleviate this rigidness in the reference model.

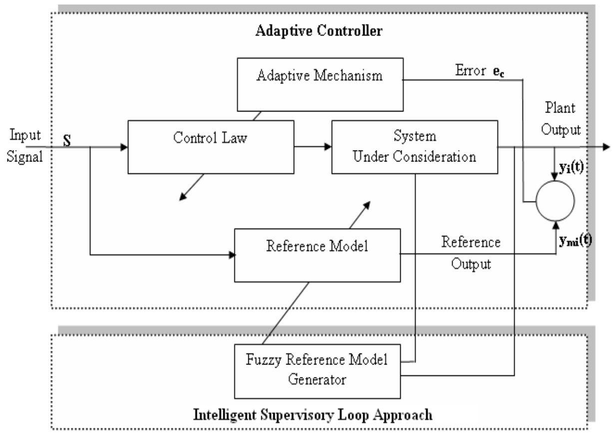

Fig. 1. Intelligent Supervisory Loop Approach

The intelligent module uses the plant output and auxiliary measurements/states, generates an appropriate reference model at each control interval, thereby providing a moving reference structure with respect to the plant without losing its desired characteristics. Thus it acts as an ISL supervising the plant controller closed loop. The system representation and problem details are as follows.

The system to be controlled has input U and output y_i. The objective is to make the control error $e_c = (y_{mi}-y_i) \rightarrow 0$

Where

y_{mi} is the output of the reference model at a specific mode and y_i is the corresponding system output.

It can be represented in state space form by

$$y_i(t) = A_{m(t)}X(t) + b_{m(t)}U(t) \tag{1}$$

Where

$y_i(t)$ is the plant output at a specific mode

U(t) is the control input

X(t) is the state vector $[X_1(t),\dots,X_n(t)]^T \in R^n$

$A_i(t)=[a_{1i},a_{2i},\dots,a_{ni}]^T \in R^{nxn}$ and

b_i takes values from the set of H constant elements which represents the known modes as indexed by subscripts $i \in \{1,2,\dots,H\}$.

Thus the parameter vector can be represented by the triple $\{(A_i,b_i,c_i),\dots,(A_H,b_H,c_H)\}$ which changes its values depending on the modes of operation. Let the above vector denote scheduled jump parameter sets for each mode specified by the parameter index i. The mode variable m(t) takes the form mapped into any of the values in the domain i and correspondingly $A_{m(t)}$ and $b_{m(t)}$ are time varying. Let the mapping of m(t) be denoted by $m(t) = \gamma[X(t-d)]$ where d represents the time delay. Please note that the above-mentioned system is of a spatial multimodal type because the dynamics are scheduled through the states with a nonlinear mapping γ.

III. MULTIPLE FUZZY REFERENCE MODEL GENERATION

Consider a fuzzy system output denoted by a function $f(\Omega)$. This can then be represented as

$$f(\Omega) = \frac{\sum_{i=1}^{r} r_i \mu_i}{\sum_{i=1}^{r} \mu_i} \tag{2}$$

Where

$\Omega \in \Re^m$ is a vector containing the relevant auxiliary states.

The fuzzy system has r rules and μ_i is the membership function of the antecedent of i^{th} rule given by the input Ω. Assume that this fuzzy system is constructed in such a way that $\sum_{i=1}^{r} \mu_i \neq 0$ for all relevant auxiliary states and the parameter r_i is the consequent of i^{th} rule. Then (2) can be written in the parameterized form as

$$f(\Omega) = \frac{\sum_{i=1}^{r} r_i \mu_i}{\sum_{i=1}^{r} \mu_i} = M^T P \vartheta = \Phi * \vartheta \tag{3}$$

Where

$$M = \begin{bmatrix} 1/\chi_1(\Omega) \\ 1/\chi_2(\Omega) \\ \vdots \\ 1/\chi_{m-1}(\Omega) \end{bmatrix} \qquad P = \begin{bmatrix} p_{1,0} & \cdots & p_{r,0} \\ \vdots & \vdots & \vdots \\ p_{1,m-1} & \cdots & p_{n,m-1} \end{bmatrix}$$

$$\vartheta = \frac{\begin{bmatrix} \mu_1 \\ \vdots \\ \vdots \\ \mu r \end{bmatrix}}{\sum_{i=1}^{r} \mu_i} \text{ and } \Phi = M^T P$$

Thus the function approximation by fuzzy scheme is equal to the product of a parameter vector Φ and weight matrix ϑ. Now let the reference model in state space form be

$$\dot{X}_m(t) = Am_{m(t)}X_m(t) + bm_{m(t)}S(t) \tag{4}$$

$$\text{and } y_{m(t)} = C^T X_{m(t)}$$

Where

$y_{m(t)}$ is the reference model output

S(t) is the command signal

$Am_{m(t)}$, $bm_{m(t)}$ are model parameters at mode m(t) and $X_m(t)$ is the state space vector.

The reference model in transfer function form will then be

$$W_{mi}(s) = y_{mi}(t) / S(t) = K_{mi} * Z_{mi} / R_{mi} \quad (5)$$

Where

K_{mi} is gain matrix

Z_{mi} is the zeros matrix with suitable locations in the system domain

R_{mi} is the polynomial matrix whose entries are monic and Hurwitz in nature.

From (5), it can be seen that the numerator and the denominator are functions of state variables X and the location of the poles and zeros are further influenced by the modes of operation of the plant. In order to include these modal transitions (5) has to be combined with (3). This can be observed value can be written as

$$\hat{W}_{mi}(s) = f(\Omega) * (K_{mi} * Z_{mi} / R_{mi}) \quad (6)$$

Thus the changes in the system dynamics can be mapped through auxiliary states to the changes in system polynomial roots or the poles/zero combination. Considering the above facts, reference model transfer function can be written as a function of the fuzzy logic output, which yields

$$\hat{y}_{mi}(t) = \hat{W}_{mi} * S(t) \underline{\underline{\Delta}} (\Phi_i * \vartheta_i) * W_{mi} * S(t) \quad (7)$$

Thus for a constant command signal, the observed reference model output will be

$$\hat{y}_{mi}(t) = \hat{\nu}(\Phi_i, \vartheta_i, W_{mi}) \quad (8)$$

Where

$\hat{W}_{mi(t)}$ is the estimate of $W_{mi(t)}$

$\hat{y}_{mi(t)}$ is the observed reference model output for the i[th] mode

Φ_i is the parameter vector developed by the fuzzy system depending on the system operating points

ϑ_i is the membership function weights and W_{mi} is the corresponding reference model transfer function.

It can be seen that the membership function weights act as a performance index function in modifying the reference model output. Based on each system modal transition the parameter vector Φ and ϑ also changes. Subsequently the reference model output moves such that the closed loop system provides a stable output with the roots on the left half s plane. This movement in the reference model secures the system from becoming unstable. Moreover, the modal transitions are smooth in nature, which reduce the transients during the changes in system mode. More importantly this avoids any pre-defined calculation. It acts like a performance index as in the case of the identified plant model as the appropriate fuzzy rules firing selects an appropriate reference model. Thus the time required for the calculation and settlement of the parameter values is zero. It also helps the controller to optimize its performance over any operating conditions.

IV. Design Methods of Fuzzy Reference Model Generator

Two ways of designing the fuzzy logic scheme based on the two different methodologies to develop stable desired closed loop system trajectories is proposed. The fact that the closed loop system trajectories pattern changes with change in the plant operating modes gives us a relation between desired trajectory and plant operating mode.

A. *Method 1*

In this method, the fuzzy logic rules are developed by changing the natural frequency of the reference model in such a way that the system closed loop performance at each mode is stable. This will develop the dominant poles and zeros of the system transfer function at each operating point. Consider the reference model transfer function as in (9), which shows the required system response.

$$W_{mi}(s) = \frac{G(s)}{1 + G(s)H(s)} \quad (9)$$

Let $G(s) = \frac{p(s)}{q(s)}$ and $H(s) = \frac{n(s)}{d(s)}$

Where

p(s),q(s),n(s) and d(s) are polynomials in s domain.

Then the closed loop transfer function will be

$$W_{mi}(s) = \frac{K(s+z_1)(s+z_2)...(s+z_m)}{(s+p_1)(s+p_2)...(s+p_n)} \quad (10)$$

Where

Z's are zeros, p's are poles and K is the gain.

Since there are real and complex conjugate poles, the closed loop response of the system to a unit step input can be represented as

$$W_{mi}(s) = \frac{K\prod_{i=1}^{m}(s+z_i)}{s\prod_{j=1}^{q}(s+p_j)\prod_{k=1}^{r}(s^2 + 2\varsigma_k w_k s + w_k^2)} \quad (11)$$

A pair of complex conjugate poles, which yields a second order term in s for each set of complex pole pairs is represented in (11). Further the location of the poles and/or zeros of the closed loop system is changed depending on the system operating modes. All the closed loop poles mentioned in (11) play an important role in the system transient response. The one, which has the dominant effect, is termed as the dominant closed-loop poles. Let these dominant complex poles for k=1 be $s^2 + 2\varsigma_1\omega_1 s + \omega_1^2$

Where

ω_1 is the natural frequency and

ς_1 is the damping ratio

From (9), the roots of the characteristic equation for the dominant poles are

$$s_1, s_2 = -\varsigma_1\omega_1 +/- \omega_1\sqrt{\varsigma_1^{\ 2} - 1} \quad (12)$$

At each mode of the system operation, the desired closed loop response with the plant and the controller in the form of ζ and ω_n is obtained first. Further reference model roots are developed to match with the corresponding roots of the characteristic equation. Thus (9) can be rewritten as

$$W_{mi}(s) = \Psi(\omega_n, \varsigma) \quad (13)$$

Combining (8) and (13)

$$\hat{y}_{mi}(t) = \lambda(\Phi_i, \vartheta_i, \omega_n, \varsigma) \quad (14)$$

Where

$\lambda = [\Phi_i, \vartheta_i, \omega_n, \varsigma]$ is the input parameter vector

However, from (2) & (3) it can be seen that Φ_i, ϑ_i are dependent on the system auxiliary states Ω. Thus the developed reference output depends on the systems auxiliary states, ς and ω_n. The fuzzy decision rules consist of inputs and outputs [21]. The inputs are the command signal and significant auxiliary outputs from the system. The outputs are the parameters in vector λ. The decision is performed by fuzzy rule, which has the following form:

R_i: **IF** Ω is A_j **AND** y is B_j **THEN** z is C_j

Where

z is the output and the subscript j indicate the j^{th} rule.

Fuzzy rules thus has two parts; the IF part which is called the premise and the THEN part, which is termed as the consequent. The former is used to describe the system within certain mode, which then triggers certain fuzzy rules. Corresponding to each of these modes the fuzzy parameter vector Φ_i and membership function weights ϑ_i will be changed. Subsequently, the crisp set of output vector λ is established. This process is continued for each time instant during system operation.

B. *Method 2*

The second method is a development inspired by the work in [11, 22] where the desired feedback gain matrix is formulated for each mode of operation so that eigenvalues and eigenvectors of the original closed loop system are recovered. In our architecture the gain matrix serves as static control gains for individual system operating points. Corresponding to this gain matrix, the desired closed loop system is derived first which acts as desired reference model structure for each mode of operation. Further, adjusted poles and zeros based on eigenvalues determine the fuzzy outputs where the significant system auxiliary states comprise the input vector. Let us assume that there are H modes of operation. Thus for each mode of operation, the closed loop system gain matrix will change and desired system response is identified with corresponding active poles and zeros. Based on that, fuzzy rules have been developed to meet the following form:

R_i: **IF** Ω is A_i **AND** y is B_i **THEN** z is $C_{i1...}C_{in}$ & $D_{i1...}D_{im}$

Where

1—n corresponds to the number of poles based on the roots of the characteristic equation and

1—m corresponds to the number of active zeros.

It has been shown in [11] that for a given system, which has undergone large variations due to some failures, or operating mode changes, a new system model can be represented as in (15). Using a reconfigurable control matrix an output gain matrix K can be established such that the maximum number of closed-loop eigenvalues of the reconfigured system is same as that of the original system. Mathematically it is rewritten as

$$\lambda_i^{\ m} = \lambda(A_m + B_m K_m C_m) = \lambda_i \quad (15)$$

Where

λ_i, is the eigenvalues and A, B and C are the system parameters

The gain matrix will satisfy the following equation

$$\lambda_i^{\ m} v_i^{\ m} = (A_m + B_m K C_m) v_i^{\ m} \quad (16)$$

The steps for the development of fuzzy inference engine are

Step 1: Prepare closed loop response of the system either by calculating natural frequency and damping constant (method one) or by finding the eigenvalues and vectors (method two) for each mode of operation.

Step 2: Find the natural frequency and damping ratio (method one) or the corresponding poles and zeros from the eigenvalues (method two) of modes.

Step 3: Develop the fuzzy inference engine using the command signal and auxiliary states as premise and the outputs corresponding to reference structure as in step two by projecting the input corresponding to each mode. Set the fuzzy inference engine for each plant mode of operation based on the projection domain and input-output mapping.

Step 4: Establish the membership function weight matrix for each mode.

Step 5: Based on step 4, establish the membership functions for each rule and create the knowledge base.

Step 6: Test the condition offline, for operating modes in order to retain the desired closed loop poles of the system.

Using this method the eigenvalues and corresponding eigenvectors of original closed loop system can be recovered regardless of the changes in the system operating modes. Instead of synthesizing the gain matrix K, here the reference model is moved such that v_i^m is close to the original eigenvector v_i (the corresponding closed-loop eigenvector of the original system) as possible. Based on this, a new reference model structure is developed at each instant of time by changing the roots of the characteristic equation. Once the offline studies are conducted reference model changes at each mode are established and the fuzzy system is developed based on recovering these poles and zeros. Please note that extracting the dominant closed loop poles as in method one and regeneration of the eigenvalues and corresponding eigenvectors as in method two are parts of the offline analysis.

V. PLANT PARAMETERIZATION AND CONTROL

From the previous section output of the plant and reference model for a multimodal system can be represented as

$$y_i(t) = A_{m(t)} X(t) + b_{m(t)} U(t) \tag{17}$$

$$\hat{y}_{mi}(t) = \hat{W}_{mi} * S(t) \underline{\underline{\Delta}} (\Phi_i * \vartheta_i) * W_{mi} * S(t) \tag{18}$$

The change in stable desired reference model looking at plant auxiliary states for each modes of operation can be obtained based on previous fuzzy reference model generator design. As the plant modal changes, the structure and dynamics of reference model also change. Thus any adaptive certainty equivalence control law can be used to control these systems. The control structure utilized here is the basic certainty equivalent control law [23] explained as follows.

Suppose a plant controller can be represented in terms of the parameter identification of the system. Then this approach proves that as the system identification routine reaches the actual values, then the controller will force the plant to the reference pattern such that the error asymptotically reduces to zero. For the proposed system control law can be expressed as

$$u = \Theta^T \omega \tag{19}$$

Where

$\Theta = \begin{bmatrix} k & \theta_0 & \theta_1^T & \theta_2^T \end{bmatrix}^T$ is the control parameter vector and $\omega = \begin{bmatrix} S(t) & y_i(t) & \omega_1^T & \omega_2^T \end{bmatrix}^T$ is the regression vector.

The regression vectors are updated online using

$$\omega_1 = \Lambda \omega_1 + Lu \tag{20}$$

$$\omega_2 = \Lambda \omega_2 + L y_i(t) \tag{21}$$

Where

Λ is a stable matrix of order $(n-1) \times (n-1)$ such that the determinant $|sI - \lambda| = Z_m(s)$ and the vector L = [0…0, 1].

Further, the control signal u which is structured earlier can be optimized as

$$k = -\text{sgn}(K_p) er \tag{22}$$

$$\theta = -\text{sgn}(K_p) e y_p \tag{23}$$

$$\theta_1^T = -\text{sgn}(K_p) e \omega_1^T \tag{24}$$

$$\theta_2^T = -\text{sgn}(K_p) e \omega_2^T \tag{25}$$

However any type of stable model reference based adaptive control can be utilized as the proposed controller. It can be seen from these equations that the control law is adjusted using the adaptive mechanism, which always look for any plant parametric changes. When there is a drastic change in the plant characteristics, this framework of the MRAC fails to control the plant efficiently. This is mainly because the plant exhibits different modes of operation, which are often referred as multi modality [1]. If appropriate reference models for the plant at these modes are provided, then exact reference model tracking may be achieved. For this purpose, previously discussed fuzzy multiple reference model is utilized. The overall scheme is as in fig. 2.

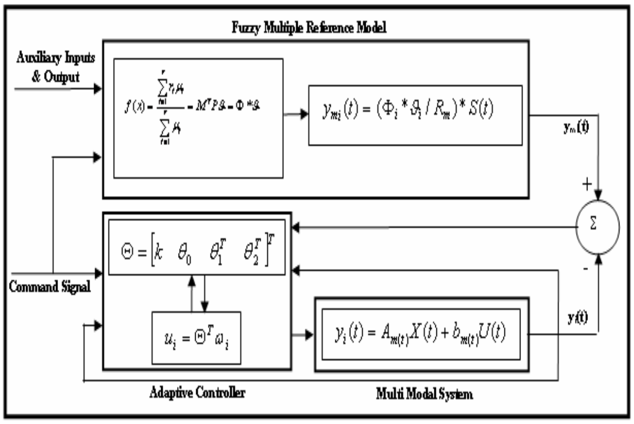

Fig. 2. Overall scheme.

VI. AN IMPORTANT FAULT TOLERANT CONTROL FEATURE

Consider the case that known faults are used to develop certain modes of operation of the plant. Further assume that nominal conditions are the rest of the modes. If a certain mode occurs at an instant, which is a combination of faulty modes and nominal condition, then the reference model developed by the fuzzy system is a fuzzified combination of models belongs to the model set Mo.

Let each mode be represented by a set of fuzzy rules. Then the model set will be the combination of the models shown in (26)

$$Mo = \sum_{i=1}^{N} \mu_i Mo_i \tag{26}$$

The membership function weights ϑ_i in (3) represent the combination of fuzzy outputs of the mode transition which in turn affect the development of model combination at each time instant. In this sense, essence of (14) is exactly represented by the output of the fuzzy logic system. It has been discussed that [14, 24], if the controllers at vertexes are stable, and the control is applied on the basis of these states, then the controller will also be stable. In this context, the membership-function weights act as the probabilistic function of each model and the controller is a direct MRAC. The reference model structure under faulted regime will be a combination of faulty modes and the nominal ones. Considering the proof of stability and developing fuzzy reference models that provide model output in even when faults occur, the overall controller is tolerant to faults.

VII. SIMULATION EXAMPLE

The system to be controlled is a single flexible link manipulator with model details as discussed in appendix. The proposed approach is applied to this manipulator for tracking the angular position. The command signal is applied for eight seconds and the tip load is varied arbitrarily at different time instant. Out of several case studies performed two cases are described below. These cases are simulated based on two different methods that are proposed in section four. The observation of tracking error and the output (angular position path of the single link manipulator) is shown in each case.

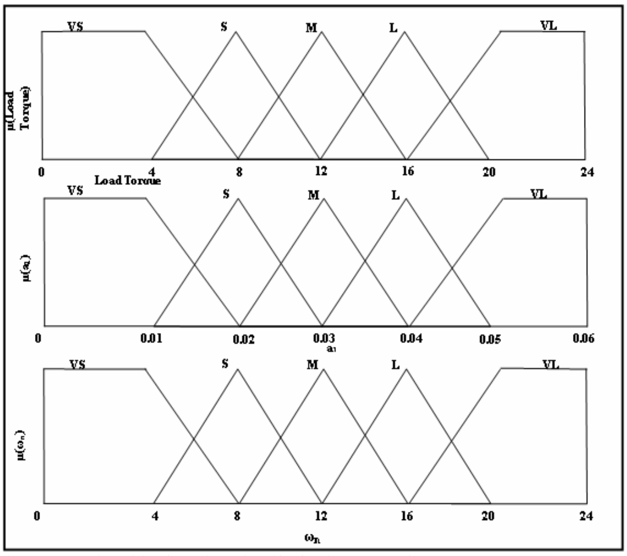

Fig. 3. Membership function details for inputs and outputs.

The value of ω_n will be evaluated at every instant of time depending on two auxiliary inputs, the system parameter a_1 and load torque. The system parameter a_1 has been derived from one of the operating state that is induced from output measurements. The importance of this state is that it is closely related to the control value. Thus the only requirement is that the fuzzy structure should acquire the auxiliary states and system output. In order to develop the fuzzy reference model generator the first step is to determine the range of the inputs and outputs. By simulating and studying the process it was found that for the range of load torque [0, 24], closed loop system response was best when ω_n is between [0, 24]. The ranges of fuzzy membership functions are as shown in fig. 3.

A. *Case 1*

In this case, the tip load of this manipulator is changed at different time instants as in Table I. The reference model representation as generalized in method one of section four has the following specific form

TABLE I
TIP LOAD VARIATION (CASE A)

Time Range (sec)	0-3	3-3.5	3.5-6	6-6.8	6.8-8.0
Load Torque (Nm)	0	15	0	20	0

$$W_m(s) = \frac{\omega_n{}^2}{s^2 + 2\varsigma\omega_n + \omega_n{}^2} \tag{27}$$

Where

the value of ς is set to 0.7.

After the division of membership functions, the rule base was created depending on system operation modes, which is divided into five operating regions; Very Small(VS), Small(S), Medium(M), Large (L) and Very Large(VL). The rule base created for this specific case is as shown in Table II.

TABLE II
RULE BASE (CASE 1)

		a_1				
	ω_n	VS	S	M	L	VL
Load Torque	VS	S	S	M	L	VL
	S	S	S	M	L	VL
	M	S	S	M	L	VL
	L	S	S	M	L	VL
	VL	S	S	M	L	VL

Thus as an example, a load torque of 10 Nm has membership functions of 0.5 in S and 0.5 in M (μ_S=0.5 and μ_M=0.5). Further, an estimated value of a_1 of 0.05 has membership functions of 0 in L and 1 in V_L, accordingly the following rules will be fired.

Rule # 9 (S/L) with μ_1=0 in S (ω_1 =5)
Rule # 10 (S/VL) with μ_2=0.5 in S (ω_2 =5)
Rule # 14 (M/L) with μ_3=0 in S ($\omega 3$ =6)
Rule # 15 (M/VL) with μ_4=0.5 in M ($\omega 4$ =6)

Where

S/L means when load torque is small and a_1 is large then the membership function is 0 and the value of ω_n is 5.

Applying the defuzzification rule the extraction process of ω_n is as follows

$[\mu_1(\omega_1) + \mu_2(\omega_2) + \mu_3(\omega 3) + \mu_4(\omega 4)]/(\mu_1 + \mu_2 + \mu_3 + \mu_4)$ which makes

$$\omega_n = \frac{0(5) + 0.5(5) + 0(6) + 0.5(6)}{1} = 5.5 \tag{28}$$

The second order reference model will then be

$$W_m(s) = \frac{30.25}{s^2 + 11\omega_n + 30.25} \tag{29}$$

Considering the proposed approach using method one, for the following manipulator tip load, fuzzy mapping is used for the reference model generation. Fig. 4 shows the position trajectory plot comparing a single reference model in which ω_n is kept as five and the above-mentioned approach during the load variation (mode change) which various modes of the system for duration of eight seconds. It can be seen that the single reference model approach shows instability while the proposed approach shows successful position trajectory tracking as illustrated in fig. 4. This figure also shows the trajectory error and the model error. The trajectory error is the error between the input command and the output of the system (which is the manipulator tip position) and the model error is the error between the outputs of the reference model and the system.

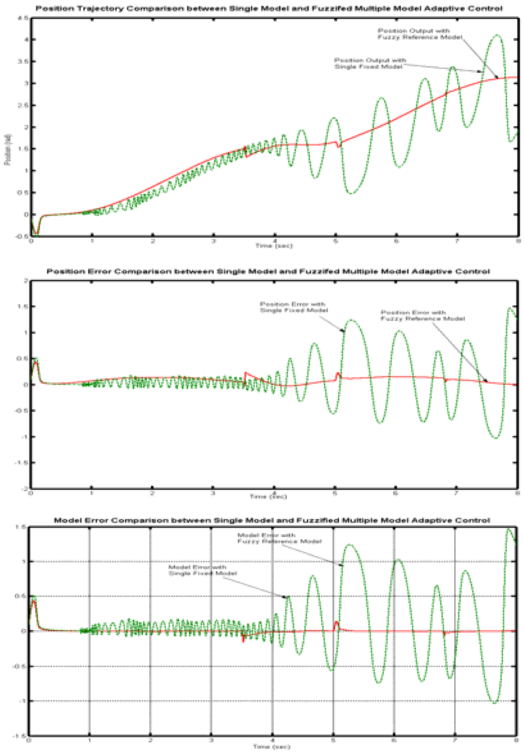

Fig. 4 . Position tracking with single reference model (dash) and multiple reference models (bold) (Case a): a) output, b) trajectory error and c) model error.

B. *Case 2*

In this case, the tip load of this manipulator is changed at different time instants as in Table III. The reference model structure based on method two, which is the identification of the eigenvalues can be represented as

$$W_m(s) = \frac{(\omega_n^{\,2}/a)(s+a)}{s^2 + 2\varsigma\omega_n + \omega_n^{\,2}} \tag{30}$$

Where

a is a factor that can be determined based on the system dynamics requirement.

TABLE III
TIP LOAD VARIATION (CASE 2)

Time Range (sec)	0-3	3-3.5	3.5-6	6-6.8	6.8-8.0
Load Torque (Nm)	10	10	0	7	0

This value is determined from system auxiliary state. The details regarding the extraction process of reference model using (30) is as follows. At first the system gain matrix has been generated based on static closed loop control. Then based on these, the desired eigen values and vectors are extracted for various system modes of operation. The basis of fuzzy reference model generator is to move the reference model structure such that the desired eigen vector is close to the system eigen vectors as much as possible. To this end based on the eigenvalue generation, the dominant poles and zeros are changed in this case unlike as in Case 1 leading to a change in the zero location. As it can be assessed, the value of the constant (a) has been synthesized from the gain matrix required to move the reference model structure. The location of the poles is also changed by a change in the value of ω_n. The effect of the change in the natural frequency and the constant 'a' derives new reference model structure at every time instant. The set of fuzzy knowledge base is shown in Table IV.

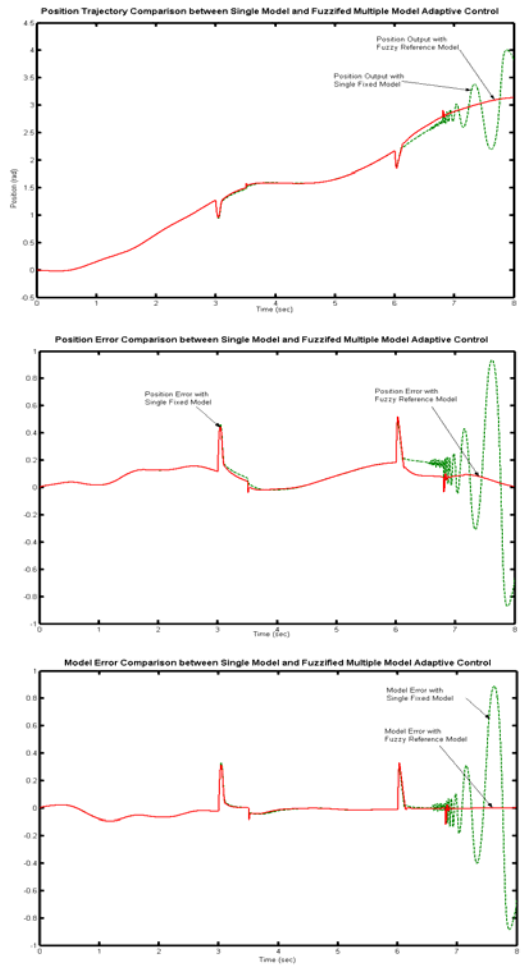

Fig. 5. Position tracking with single reference model (dash) and multiple reference models (bold) (Case b), a) output, b) trajectory error and c) model error.

TABLE IV
RULE BASE (CASE 1)
a_1

Load Torque	ω_n	VS	S	M	L	VL
	VS	S	S	M	L	S
	S	S	S	M	L	S
	M	L	S	M	L	S
	L	S	S	M	L	VL
	VL	S	S	M	L	VL

It is observed that the transient response of the system with one zero and two poles will be affected by locations of zeros. The ω_n value of the fixed model is kept constant at two for this case. For fuzzy reference model, ω_n is varied depending on the fuzzy rules. It is worth noting that even though fuzzy model generation requires various known ω_n values, the model representation of each of them separately gives the same results as when ω_n is equal to two. This is due to the reason that each of these models with various individual ω_n values fails to perform at one point or the other in the trajectory. Thus the simulation results for each models with various ω_n values are not shown in both these cases. Fig. 5 shows the responses of the system position output with fuzzy reference model generation and a fixed model. It can be seen that the fixed model shows the instability with the error increasing rapidly as in Case 1, while the response of the proposed method achieved its objective. For both these cases fig. 6 shows the desired trajectory.

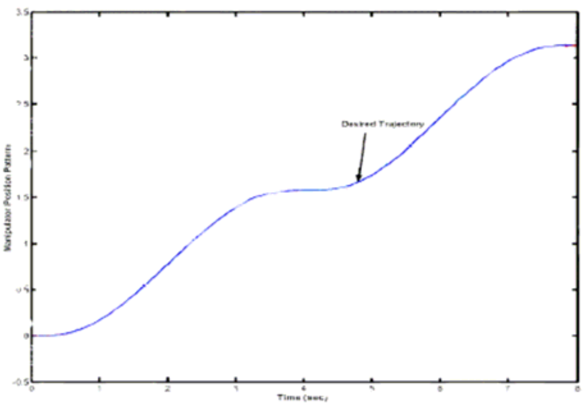

Fig. 6. Reference Position Trajectory.

VIII. CONCLUSION

An adaptive controller based on a new Fuzzy Reference Model Generator is proposed for multimodal systems. The controller concept, adaptive law and the design methods of the fuzzy switching scheme were established. The scheme provides soft switched fuzzy reference model and was found stable, especially at the modal boundaries when the 'hard switching' mathematical approach fails. Further the scheme is computationally feasible, and fault tolerant. The approach is advantageous in the fact that the development of the fuzzified knowledge base and further changing the reference model alleviates the prevailing computational complexity to develop best suitable model at each mode. The feasibility and effectiveness of the proposed scheme have been investigated by applying to an important and challenging practical system; a position control of a single link flexible robotic manipulator. Investigation results showed that the proposed scheme outperformed both traditional and single reference model adaptive controllers. It is worth noting that the development of the knowledge base is the only design process for which system modes and operating range should be known a priori.

APPENDIX

Given the system dynamics [18], the robotic link's dynamics can be represented as below

$$y(1) = \beta(t), y(2) = \dot{\beta}(t), y(3) = \phi(t), y(4) = \dot{\phi}(t) \qquad (31\text{-}34)$$

The variables y(1)- y(4) are robotic link's angular position, link's angular velocity, link's vibration mode, link's vibration mode first derivative respectively. Differentiating we get

$$\dot{y}(1) = \dot{\beta}(t), y(2) = \dot{\beta}(t), \dot{y}(3) = \dot{\phi}(t), y(4) = \dot{\phi}(t) \qquad (35\text{-}38)$$

Emerging from above the complete model of a single link manipulator is as follows.

$$\dot{y}(1) = y(2)$$

$$\dot{y}(2) = \frac{1}{r_1 r_9 - r_5{}^2}\left[r_1(\tau + r_2 - r_3 - r_4) - r_5(r_6 + r_7 - r_8)\right]$$

$$\dot{y}(3) = y(4)$$

$$\dot{y}(4) = \frac{1}{r_1 r_9 - r_5{}^2}\left[r_9(r_6 + r_7 - r_8) - r_5(\tau + r_2 - r_3 - r_4)\right] \quad \text{and}$$

$$\dot{y}(5) = \frac{1}{L_a}\left[V_{in} - R_a y(5) - nK_b y(2)\right] \qquad (39\text{-}43)$$

Where

$$\tau = n\eta\left(K_m y(5) - nK_d y(2) - T_f\right)$$

$$r_1 = \rho\zeta_1, \; r_2 = \frac{L^2}{2}\left[g\rho\cos(y(1))\right]$$

$$r_3 = 2\rho\zeta_1 y(3) y(4) y(2)$$

$$r_4 = g\rho y(3)\zeta_3 \sin(y(1))$$

$$r_5 = \rho\zeta_2, \; r_6 = \rho y(3) y(2)^2 \zeta_1$$

$$r_7 = g\rho\cos(y(1))\zeta_3$$

$$r_8 = EIy(3)\zeta_4 \quad r_9 = \frac{L^3}{3}\rho + \rho y(3)^2\zeta_1$$

Corresponding nomenclature is as follows.

R_a and L_a : D.C motor resistance and inductance respectively.
K_m : D.C motor torque constant.
K_b : Back e.m.f. constant.
T_f and K_d : Torque friction and Damping coefficient respectively.
n and η : Gear ratio and gear efficiency respectively.
E : Young's modulus of elasticity.
I : Link's area moment of inertia with respect to its neutral axis.
L : Length of the link.
g : Earth's gravity.
ρ : Mass per unit length of the link.
$\zeta_1, \zeta_2, \zeta_3$ and ζ_4 : Link's area moment of inertia w. r. to all other axes.

The values of the parameters are given in Table V.

TABLE V
PARAMETRIC VALUES OF A SINGLE LINK MANIPULATOR

R_a=8.33	n=65.5	ρ=0.1658
L_a=0.00617	η=0.66	ζ_1=0.4661541842
K_b=0.039534088	E=68.9e^9	ζ_2=0.3835861510

ACKNOWLEDGMENT

Author would like to thank Dr. Adel A. Ghandakly, Professor and

Chair, California State University at Chico, for his kind support and guidance to conduct this research.

REFERENCES

[1] S.G. Fabri and V. Kadirkamanathan, Functional Adaptive Control: An Intelligent Systems Approach, Springer-Verlag, 2001, ch 1,7-9.
[2] D.T. Magill, "Optimal Adaptive Estimation of Sampled Stochastic Processes", IEEE Trans. on Automatic Control, vol. AC-1965, pp.434-439, Oct.1965.
[3] X.Rong Li,Youmin and Zhang, "Numerically Robust Implementation of Multiple-Model Algorithms", IEEE Trans. on Aerospace and Electronic Systems,vol.36, no.1, pp. 266-278, Jan. 2000.
[4] P. D. Hanlon and P. S. Maybeck, "Multiple-Model Adaptive Estimation Using a Residual Correlation Kalman Filter Bank", IEEE Trans. on Aerospace and Electronic Systems, vol. 36, no.2, pp.393-406, April 2000.
[5] K.S. Narendra and J. Balakrishnan, "Adaptive Control Using Multiple Models", IEEE Trans. on Automatic Control,vol.42, no. 2, pp.171-187, Feb.1997.
[6] K. S. Narendra, C. Xiang, "Adaptive Control of Discrete-Time Systems Using Multiple Models," IEEE Trans. on Automatic Control, vol.45, no. 9, pp.1669-1685, Sept. 2000.
[7] K.S. Narendra and J. Balakrishnan, "Improving Transient Response of Adaptive Control Systems using Multiple Models and Switching", IEEE Trans. on Automatic Control, vol.39, no. 9, pp.1861-1866, Sept. 1994.
[8] K.S. Narendra, J. Balakrishnan, and M.K. Ciliz, "Adaptation and Learning Using Multiple Models, Switching, and Tuning", IEEE Control Systems Magazine, vol.15, no. 3, pp.37-51, June 1995.
[9] P. S. Maybeck and R. D. Stevens, " Reconfigurable Flight Control Via Multiple Model Adaptive Control Methods", IEEE Trans. on aerospace and electronic systems, vol. 27, no.3, pp.470-479, May 1991.
[10] M. Athans, D. Castanon, K.-P. Dunn, C. Greene, W. Lee, N. Sandell, and A. Willsky, "The stochastic control of the F-8C aircraft using a multiple-model adaptive control MMAC method-part I: Equilibrium flight," IEEE Trans. Autom. Contr., vol. AC-22, no. 5, pp. 768-780, Oct. 1977.
[11] J. Jiang, "Design of Reconfigurable Control Systems Using Eigenstructure Assignments", International Journal of Control, vol.59, no.2, pp.395-410, Jan.-Feb., 1994.
[12] J.T. Spooner and K. M. Passino, "Stable Adaptive Control Using Fuzzy Systems and Neural Networks", IEEE Trans. on Fuzzy Systems, vol. 4, no.3, pp.339-359, Aug. 1996.
[13] W. Shyong and C. J. Sun, "Fuzzy Model based Adaptive Control for a Class of Nonlinear Systems", IEEE Trans. on Fuzzy systems, vol.9 no.3, pp.413-425, June 2001.
[14] A. Ichtev, "Multiple Fuzzy Models for Fault Tolerant Control", International Journal of Fuzzy Systems, vol.5, no.1, pp.31-39, March 2003.
[15] C.W. Park and Y.W. Cho " T-S model based indirect adaptive fuzzy control using online parameter estimation," IEEE Trans. on Systems Man and Cybernetics- Part B, vol. 34, no. 6, pp. 2293-2302, Dec. 2004.
[16] Y.G. Lue, W.Y. Wang and T.T. Lee, "Robust adaptive fuzzy-neural controllers for uncertain nonlinear systems," IEEE Trans. Robotics and Automation, vol.15, pp.805-817, Oct. 1999.
[17] Y.G. Lue, W.Y. Wang and T. T. Lee, "Observer-based direct adaptive fuzzy-neural control for nonaffine nonlinear systems," IEEE trans. on neural networks, vol. 16, no.4, pp.853-861, July 2005.
[18] S. Kamalasadan, "A new Generation of Adaptive Control- An Intelligent Supervisory Loop Approach," Adaptive Control Algorithms with Multiple Models and Fuzzy Logic Switching , Ph.D. Dissertation, The University of Toledo, Toledo: Ohio, August 1999.
[19] S. Kamalasadan, Adel A Ghandakly, Khalid Al-Olimat , "A Fuzzy Multiple Reference Model Adaptive Control Scheme for Flexible Link Robotic Manipulator", in Proc. of IEEE International Conference on Computational Intelligence for Measurements Systems and Applications, 2004, pp.162-167.
[20] S. Kamalasadan, Adel A. Ghandakly, Khalid Al-Olimat, "A fuzzy Logic based Multiple Reference Model Adaptive Control", in Proc. of the 16th International Conference of Computer Applications in Industry and Engineering, 2003, pp. 58-61.
[21] L. Reznik , Fuzzy Controllers, Newness, 1996,ch.1-5.
[22] A.N. Andry, E.Y. Shapiro, J.C. Chung "Eigenstructure assignment of linear systems," IEEE Trans. on aerospace and electronic systems, vol.19, no.5, pp.711-728, Sept. 1983.
[23] P. A. Ioannou and Jing Sun, Robust Adaptive Control, Upper Saddle River, NJ: Prentice-Hall, 1996, ch.2-6.
[24] S. Kanev and M. Vergaegen, "Controller Reconfiguration for Non-Linear Systems," Control Engineering Practice, vol.8, no.11, pp.1223-1235, Nov. 2000.

S. Kamalasadan (M' 05) received his Ph.D. in Electrical Engineering from the University of Toledo, Ohio in 2004, M.Eng in Electrical Power Systems Management, from the Asian Institute of Technology, Bangkok Thailand in 1999 and B Tech. degree in Electrical and Electronics from the University of Calicut, India in 1991. He is currently working as an Assistant Professor at the University of West Florida. He has held industrial positions for six years and has teaching and research experience for four years. He has won several awards including the faculty summer research award at UWF, Outstanding teaching award from the graduate student association University of Toledo, 2001- 2002 and from the college of engineering, 2002-2003. His research interests include Intelligent and Autonomous Control, Power Systems dynamics, Stability and Control, AI applications in control of Dynamic Systems and Real-time Embedded Systems Applications. He is a technical reviewer of IEEE Transactions of Power Systems, IEEE transactions of Fuzzy systems, IEEE transactions of education and several other international journals and conferences.

Figures

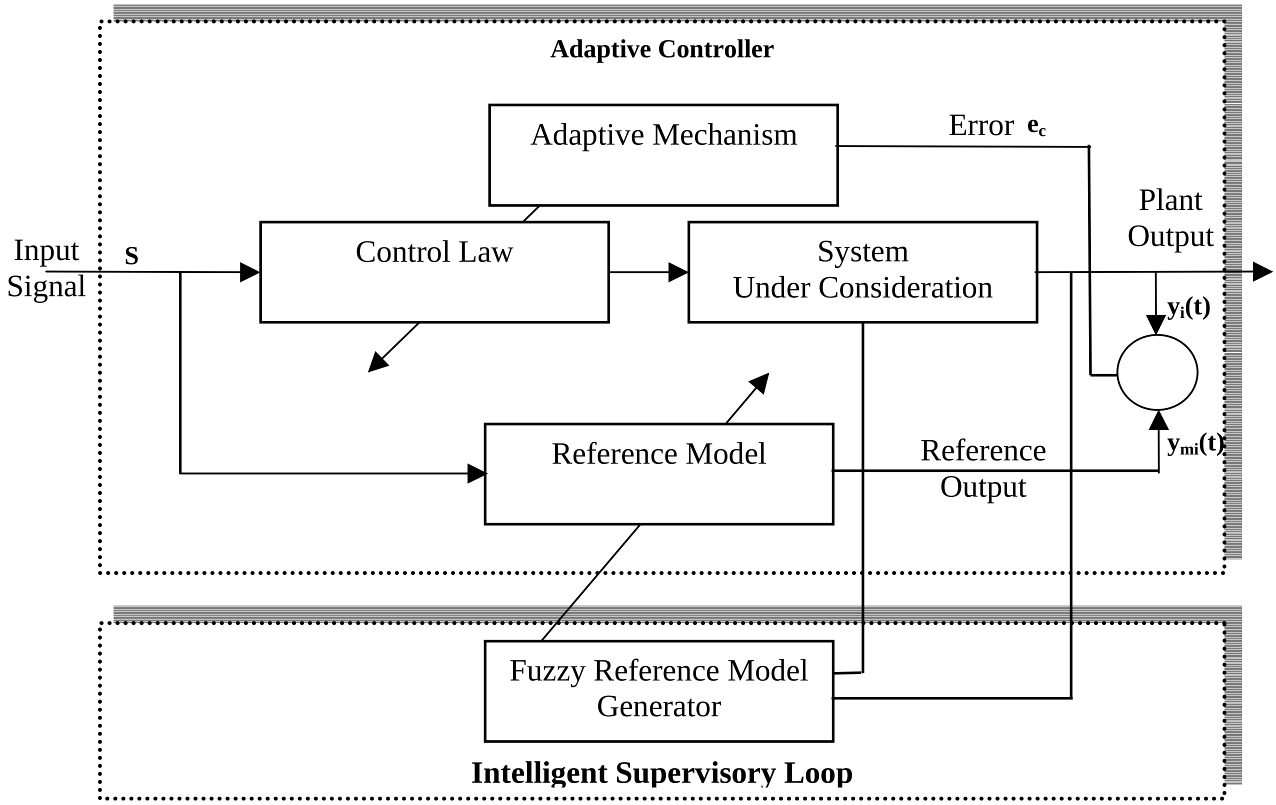

Fig. 1. Intelligent Supervisory Loop Approach

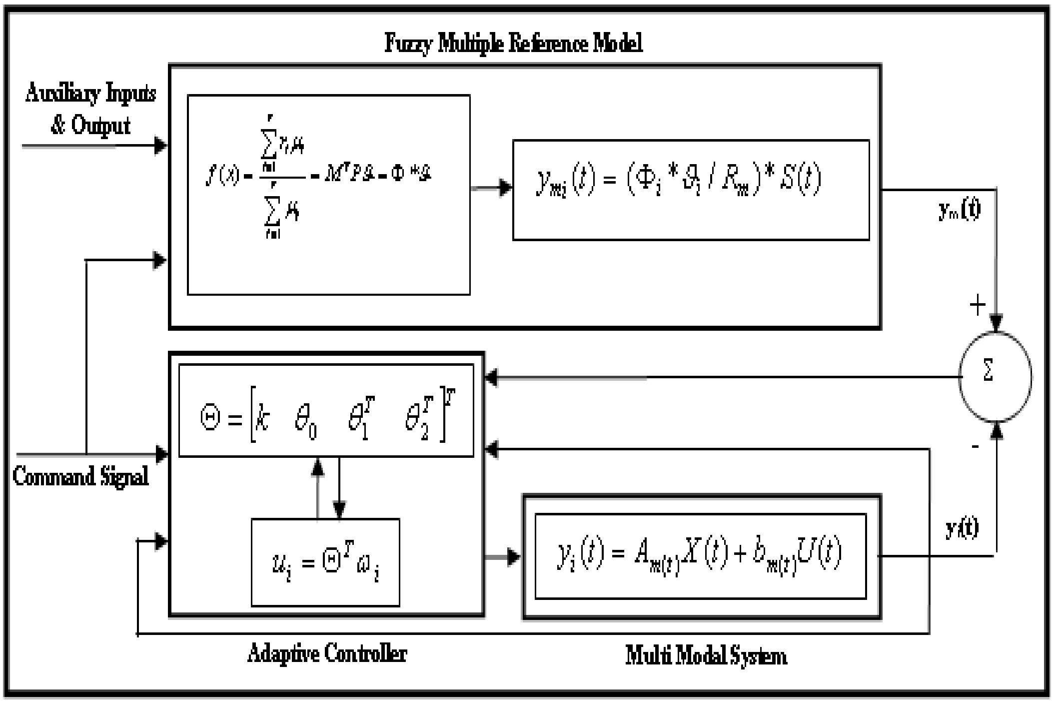

Fig. 2. Overall scheme.

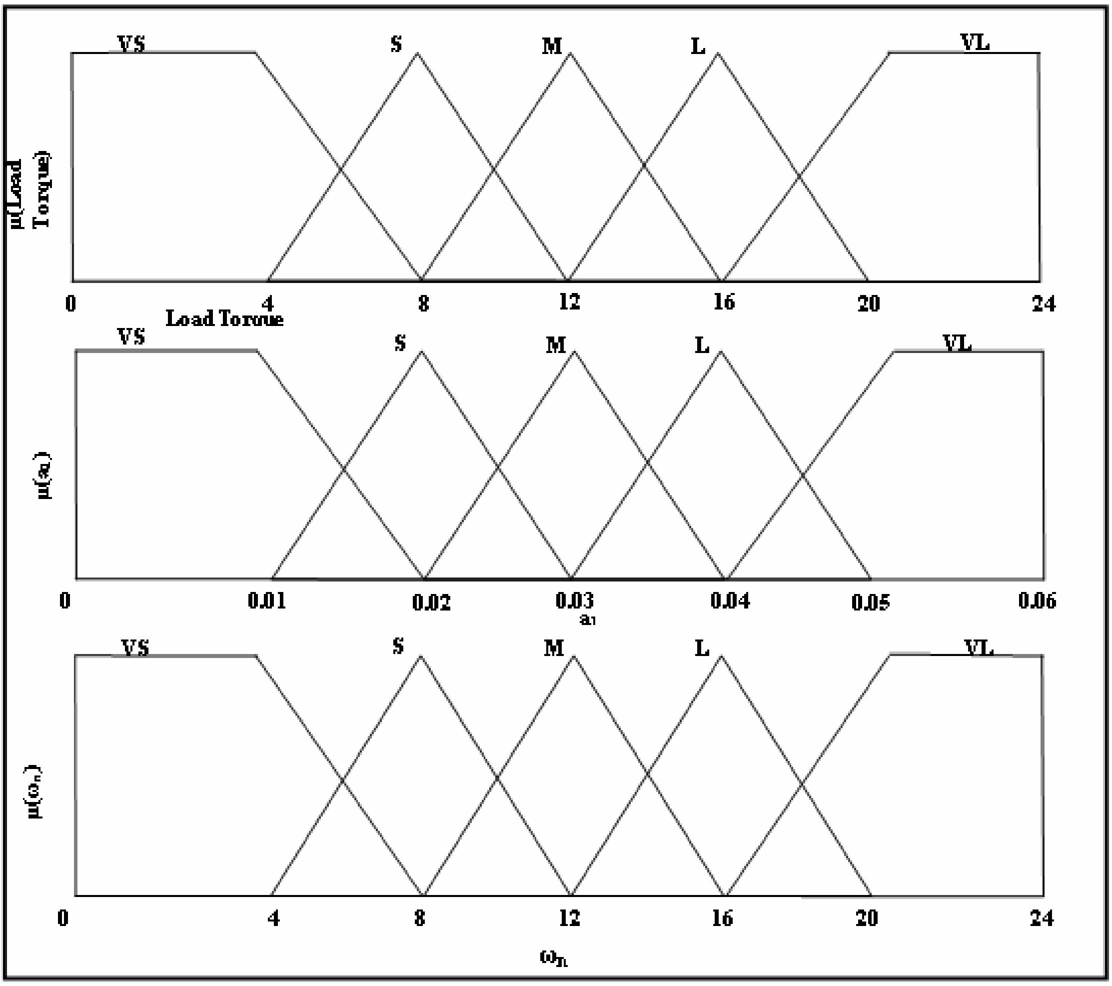

Fig. 3. Membership function details for inputs and outputs.

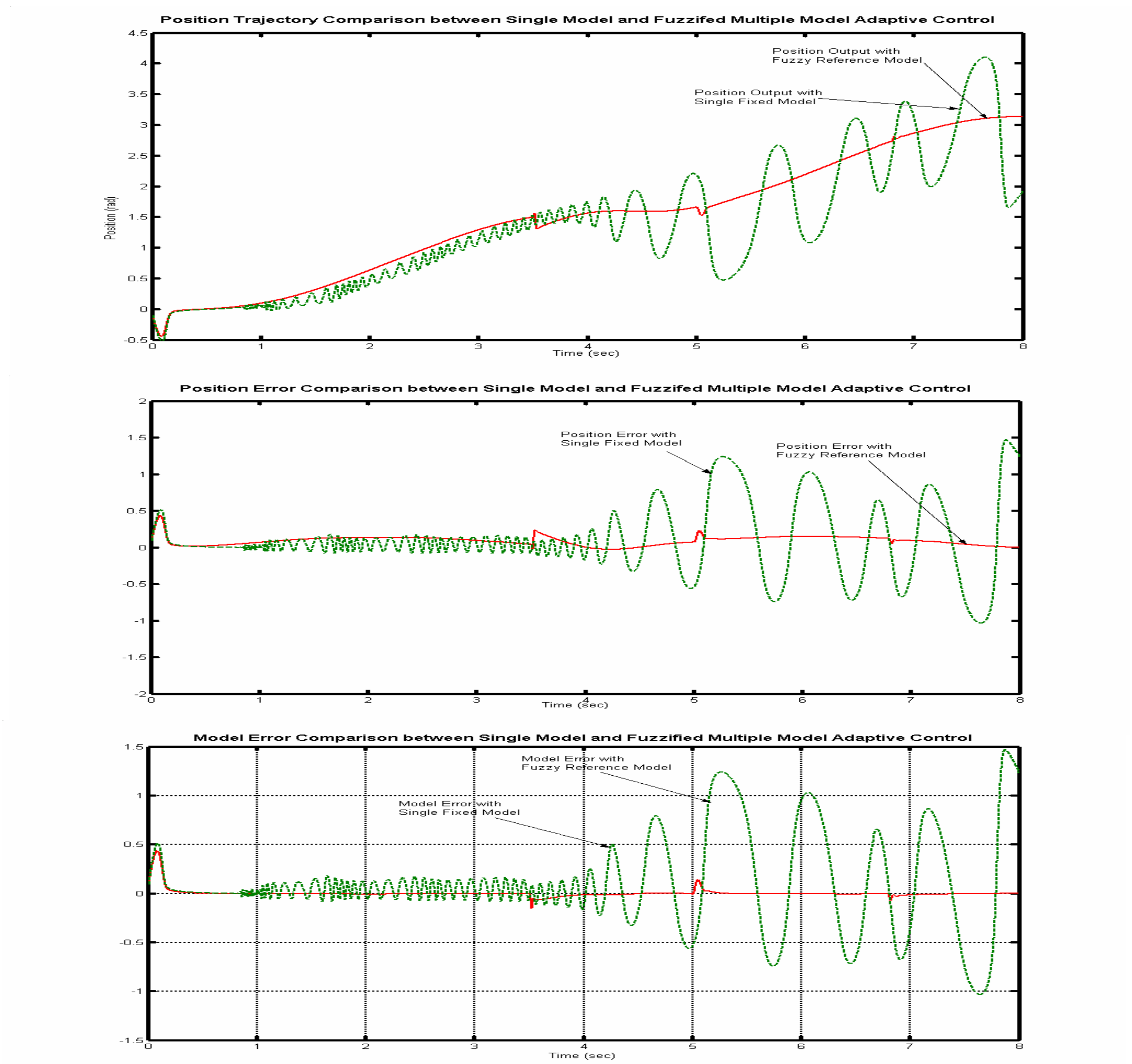

Fig. 4 . Position tracking with single reference model (dash) and multiple reference models (bold) (Case a): a) output, b) trajectory error and c) model error.

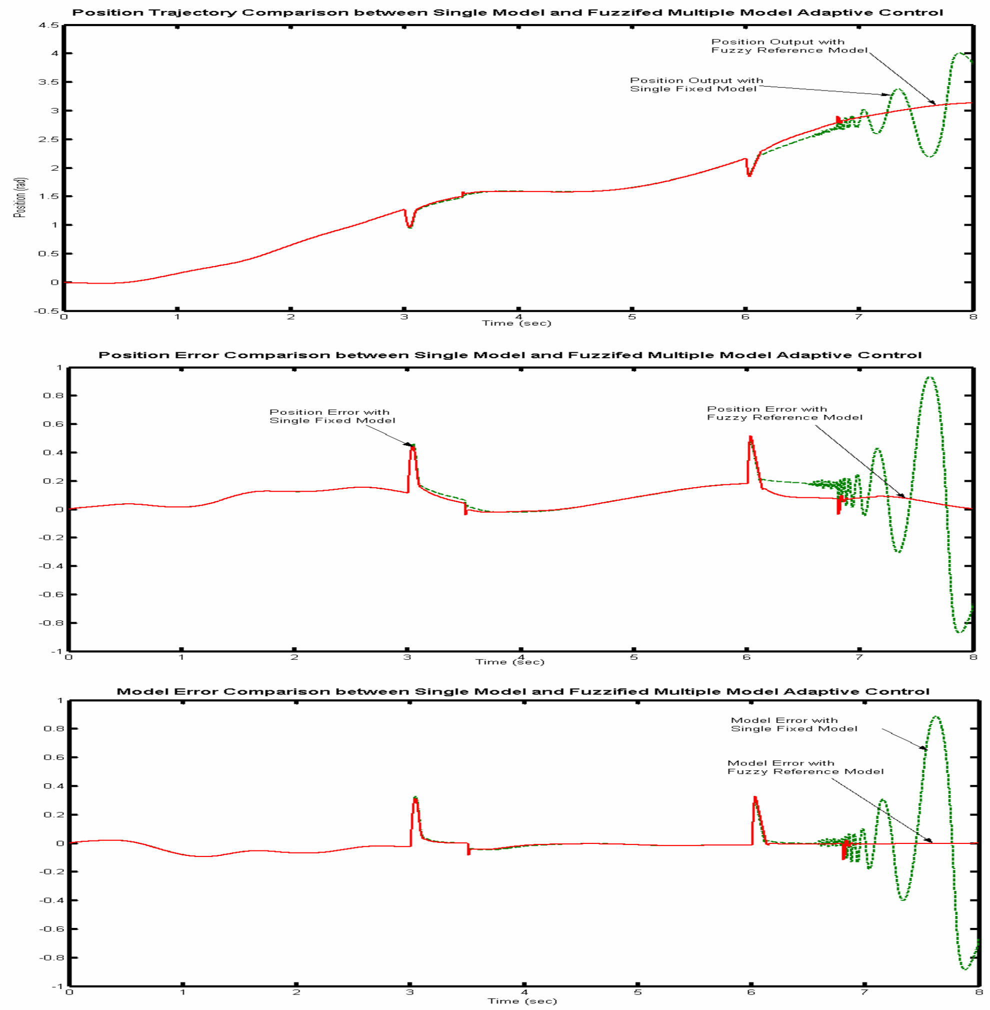

Fig. 5. Position tracking with single reference model (dash) and multiple reference models (bold) (Case b), a) output, b) trajectory error and c) model error.

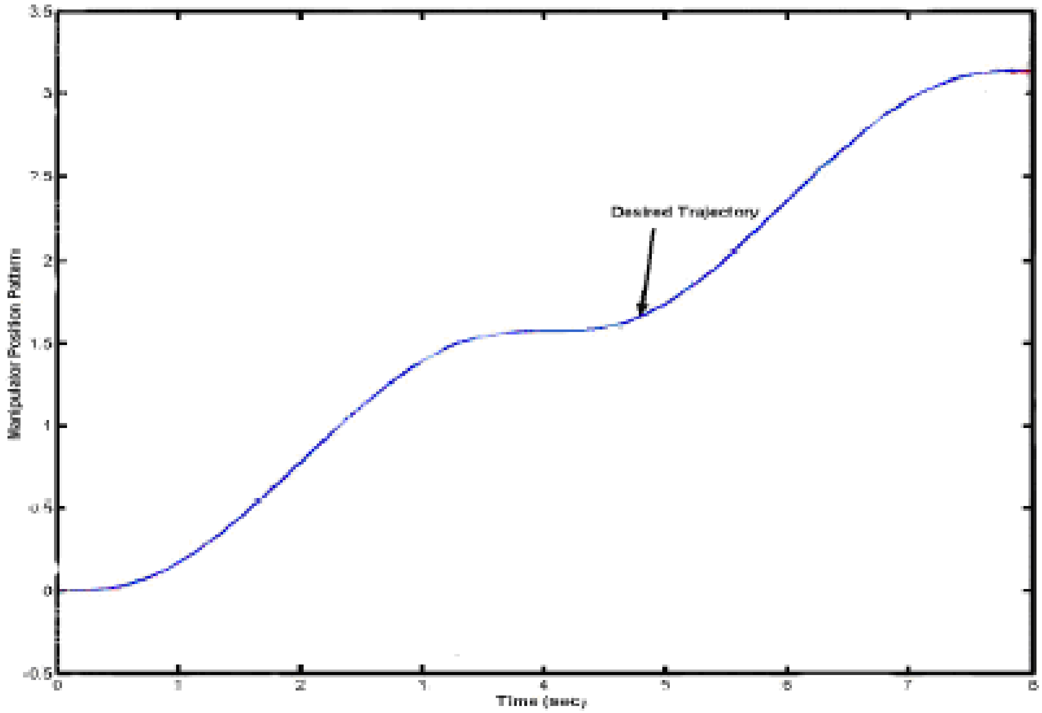

Fig. 6. Reference Position Trajectory.