

Optimized Controller for Nonlinear Time Delay Systems

Sucheta Sampatrao Yadav.

Student, Department of Electrical Engineering, G. H. Raisoni Institute of Engineering and Technology, Pune, Savitribai Phule Pune University, Pune, India

ABSTRACT: This paper proposes an optimal fuzzy adaptive model for a class of nonlinear systems and a fuzzy controller to stabilize such systems. The robust control problem is designed for a class of such nonlinear time delay systems. Takagi-Sugeno (T-S) fuzzy models can provide proper representation of nonlinear systems. In this paper, the T-S fuzzy model approach is extended to the stability analysis and control design for continuous time nonlinear systems with time delay and uncertainty. The simulation results are given to show the effectiveness of system with optimized fuzzy controller.

KEYWORDS: Fuzzy control, Genetic algorithm, nonlinear systems, Takagi– Sugeno (T–S) fuzzy approach, time-delay systems.

I. INTRODUCTION

Most of the process control systems are nonlinear and their mathematical model is also difficult to derive precisely. These systems are most common in industrial applications. So it is important to study behavior of nonlinear system and control of nonlinear system. The effect of time delay and uncertainty in nonlinear system is also important considerations. Time delay is very common and occurs in a variety of dynamic systems, such as chemical processes, various industrial processes, transmission lines etc. Time delay produces instability in the system, the stability of system and control problem related with the time-delay systems have been studied in the past decades. As compared with the linear time-delay systems, there is less progress in nonlinear time-delay systems. Takagi-Sugeno(T–S)-fuzzy-model-based fuzzy control has been shown to be an efficient approach to deal with analysis and synthesis problems for complex nonlinear systems. For modeling and control of a nonlinear plant, the T–S fuzzy control technique has become one of the most important control schemes of the last decade, providing systematic design algorithms to guarantee the stability and control performance of closed-loop systems.[1]– [5]. This paper deals with study of fuzzy controller approach for nonlinear time delayed systems. It is commonly known that T–S fuzzy-model-based control technique is simple and effective in the control of complex systems with nonlinearities. In addition, a fuzzy controller has been optimized with the help of genetic algorithm approach. Next the parallel distributed compensation (PDC) scheme is used to stabilize the T–S fuzzy model. A typical approach for the analysis and synthesis of nonlinear system with time delay is the local linearization approach. First a linearization model on the nominal operating point is obtained and then a linear feedback control is designed for this linear model. First the nonlinear plant is represented by a dynamic fuzzy model. In this type of fuzzy model, local dynamics in different state-space regions are represented by linear models. The idea is that for each local linear model, a linear feedback control is designed. The resulting overall controller, which is nonlinear in general, is again a fuzzy blending of each individual linear controller. In this paper, we extend the above procedure to the nonlinear systems with time delay. The output of the overall fuzzy controller is a fuzzy blending of each individual linear controller. The design procedure aims for getting the stable fuzzy controllers for the nonlinear system which has time delay in it.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

II. NON-LINEAR TIME-DELAY SYSTEM

Ordinary differential equations can be written in the form of,

$$\dot{x}(t) = f(t, x(t)) \quad (1)$$

In this description, the variables $x(t) \in R^n$ are known as the state variables, and the differential equations characterize the evolution of the state variables with respect to time. In other words, the value of the state variables $x(t)$, $t_0 < t < \infty$, for any t_0 , can be found once the initial condition $x(t_0) = x_0$ is known [8].

In practice, many systems cannot be satisfactorily modeled by an ordinary differential equation. For a particular class of many systems, the future evolution of the state variables $x(t)$ not only depends on their current value $x(t_0)$, but also on their past values, say $x(\xi)$, $t_0 - r < \xi < t_0$. Such a system is said to be a time-delay system.

Consider the following nonlinear time-delay system:

$$\dot{x}(t) = F(x(t), x(t-d(t))) + \Delta F(x(t), x(t-d(t))) + G(x(t))u \quad (a)$$

$$\begin{aligned} x(t) &= \varphi(t), t \\ &\in [-\bar{d}, 0] \end{aligned} \quad (b)$$

Where $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ is the input vector and $d(t)$ is the delay time of the system state with $d(t) \leq \bar{d}$, $F(\cdot), G(\cdot)$ are two continuous nonlinear functions, and $\Delta F(\cdot)$ is an uncertain nonlinear function. $\varphi(t) \in C_{n, \bar{d}}$ is a vector-valued initial continuous function.

Takagi and Sugeno proposed an effective way to represent a fuzzy model of a nonlinear dynamic system. It uses a linear input/output (I/O) relation as its consequence of individual plant rules. A TS fuzzy time-delay model is composed of r plant rules that can be represented as follows. [7], [11].

IF θ_1 is μ_{i1} and ... and θ_p is μ_{ip} THEN

$$\dot{x}(t) = A_i x(t) + A_{di} x(t-d(t)) + B_i u(t) + \Delta f_i(x(t), x(t-d(t))) \quad (c)$$

Where μ_{ij} is the fuzzy set, and A_i, A_{di}, B_i are some constant matrices of compatible dimensions, r is the number of IF-THEN rules, and $\theta_j(t) = [\theta_1(t) \theta_2(t) \dots \theta_p(t)]$ are the premise variables. It is assumed that the premise variables do not depend on the input variables explicitly.

The overall fuzzy model is achieved by fuzzy blending of each individual rule as follows:

$$\dot{x}(t) = \sum_{i=1}^r h_i(\theta) \left(A_i x(t) + A_{di} x(t-d(t)) + B_i u(t) + \Delta f_i(x(t), x(t-d(t))) \right) \quad (d)$$

Where $h_i(\theta) = \mu_i(\theta) / \sum_{i=1}^r \mu_i(\theta)$

$$\Delta f_i = \begin{bmatrix} \delta_i(x(t), x(t-d(t))) \\ v_i(x(t), x(t-d(t))) \end{bmatrix} \quad (e)$$

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

Decomposing above system into the following form:

$$\dot{x}_1(t) = \sum_{i=1}^r h_i(\theta) \left(A_{i11}x_1(t) + A_{i12}x_2(t) + A_{di11}x_1(t-d(t)) + A_{di12}x_2(t-d(t)) + \delta_i(x(t), x(t-d(t))) \right) \quad (f)$$

$$\dot{x}_2(t) = \sum_{i=1}^r h_i(\theta) \left(A_{i21}x_1(t) + A_{i22}x_2(t) + A_{di21}x_1(t-d(t)) + A_{di22}x_2(t-d(t)) + u(t) + v_i(x(t), x(t-d(t))) \right) \quad (g)$$

III. SYSTEM MODEL FOR FUZZY CONTROLLER: CONTINUOUSLY STIRRED TANK REACTOR (CSTR)

The process, shown in Fig.1, consists of a reactor and a separator. The reaction is of the form $A \rightarrow B$, where A is considered as reactant and B is the product. This reaction is exothermic. The process has first ordered reaction rate given by

$$r = K_0 \exp\left[\frac{-E}{RT(t)}\right] A(t) \quad (2)$$

Where K_0 is pre-exponential constant and E is activation energy of the reaction. T is temperature and $A(t)$ is concentration of species A in the reactor.

A process can be represented from mass and energy balances equations. The two nonlinear differential difference equations are given below:

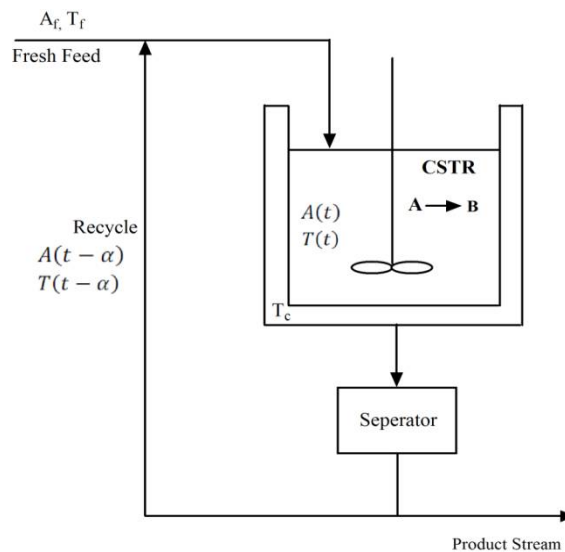


Figure 1. Continuously Stirred Tank Reactor Model

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

$$V \frac{dA}{dt} = \lambda q A_0 + q(1 - \lambda)A(t - \alpha) - qA(t) - VK_0 \exp \left[\frac{-E}{RT(t)} \right] A(t) \quad (3)$$

$$VC\rho \frac{dT}{dt} = qC\rho[\lambda T_0 + (1 - \lambda)T(t - \alpha) - T(t)] + V(-\Delta H)K_0 \exp \left[\frac{-E}{RT(t)} \right] A(t) - U(T(t) - T_w) \quad (4)$$

Where $A(t) = \varphi_1(t)$ and $T(t) = \varphi_2(t)$ for $t \in [-\alpha, 0]$, $A(t)$ is concentration of chemical A. $T(t)$ is temperature and the remaining constants $\alpha, \lambda, q, A_0, V, K_0, -E/R, C, \rho, (-\Delta H), U,$ and T_w are all positive and defined in Table 1. The constant λ is the recycle coefficient, which satisfies the conditions $\lambda \in [0, 1]$. The objective is to control the concentration of A, $A(t)$.

They can be reduced to dimensionless form using the notation;

$$x_1(t) = \frac{A_0 - A(t)}{A_0} \quad x_2(t) = \frac{T(t) - T_0}{T_0} \left(\frac{-E}{RT(t)} \right) \quad \theta_1(t) = \frac{A_0 - \varphi_1(t)}{A_0} \quad (5)$$

$$\theta_2(t) = \frac{\varphi_2(t) - T_0}{T_0} \left(\frac{-E}{RT(t)} \right) \quad t_{new} = \frac{t}{v} \quad v = \frac{V}{q\lambda} \quad \tau = \frac{\alpha}{v} \quad (6)$$

$$\gamma_0 = \frac{E}{RT(t)} \beta = \frac{Uv}{VC\rho} D_a = K_0 v \exp(-\gamma_0) \quad (7)$$

$$H = \frac{(-\Delta H)A_0 E}{C\rho T_0^2 R} u(t) = \frac{T_0 - T_w}{T_0} \left(\frac{-E}{RT(t)} \right) \quad (8)$$

TABLE I. CSTR PARAMETERS

$A(t)$	Chemical concentration of chemical specie A
$T(t)$	Reactor temperature
α	Recycle delay time
V	Reactor volume
λ	Coefficient of recirculation
q	Feed flow rate
A_0	Feed concentration
K_0	Reaction velocity constant
E/R	Ratio of Arrhenius activation energy to the gas constant
ρ	Density
C	Specific heat
$-\Delta H$	Heat of reaction (positive)
U	Heat transfer coefficient times the surface area of reactor
T_0	Feed temperature
T_w	Average coolant temperature in reactor cooling coil

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

Then (5) and (6) in dimensionless variables become

$$\dot{x}_1(t) = f_1(x) + \left(\frac{1}{\lambda} - 1\right)x_1(t - \tau) \tag{9}$$

$$\dot{x}_2(t) = f_2(x) + \left(\frac{1}{\lambda} - 1\right)x_2(t - \tau) + \beta u \tag{10}$$

Where

$$x(t) = [x_1(t) \ x_2(t)]^T \tag{11}$$

$$f_1(x) = \frac{-1}{\lambda}x_1(t) + D_a(1 - x_1(t))\exp\left(\frac{x_2(t)}{1+x_2(t)/\gamma_0}\right) \tag{12}$$

$$f_2(x) = -\left(\frac{1}{\lambda} + \beta\right)x_2(t) + HD_a(1 - x_1(t))\exp\left(\frac{x_2(t)}{1+x_2(t)/\gamma_0}\right) \tag{13}$$

$$x_i(t) = \theta_i(t) \Phi_{opt} \in [-\tau, 0], \quad i = 1, 2.$$

The state $x_1(t)$ is the conversion rate of the reaction $0 \leq x_1(t) \leq 1$; $x_2(t)$ is the dimensionless temperature. Clearly, we must restrict $\lambda \neq 0$. Constants H, β, D_a, γ_0 and τ are all positive. We assume that only the temperature can be measured on line, i.e.

$$y(t) = [0 \ 1]x(t) \tag{14}$$

The parameters are given as

$$\gamma_0 = 20; H = 8; D_a = 0.072; \lambda = 0.8; \tau = 2$$

A. Phase Plane Analysis of CSTR Model:

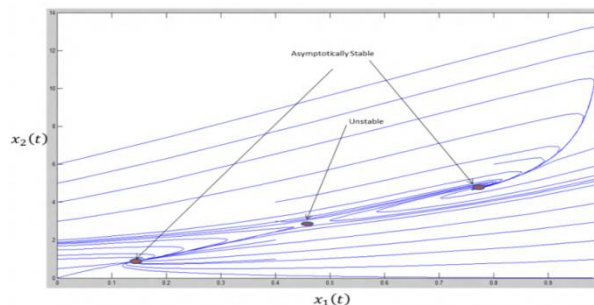


Figure 2. Phase portrait of CSTR model

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

It is easy to find the three steady-states when $u = 0$ are;

$$x^1 = (0.1440 \quad 0.8862)$$

$$x^2 = (0.4472 \quad 0.7520)$$

$$x^3 = (0.7646 \quad 4.7052)$$

B. Linearizing the System around Operating Point:

In this part of the paper, we will consider simple fuzzy control law designing for the system. For this we use the analytical approach. Consider the nonlinear system with time delay described by (8) and (9), which can be linearized around the stationary point given as

$$\dot{x}(t) = A_1(x^s, u^s)(x(t) - x^s) + A_2(x(t - \tau) - x^s) + B(u - u^s) \quad (15)$$

Where

$$x_s = [x_1^s \quad x_2^s]^T$$

$$A_1(x^s, u^s) = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_2}{\partial x} \end{bmatrix}^T (x^s) \quad (16)$$

$$A_2(x^s, u^s) = \begin{bmatrix} \frac{1}{\lambda} - 1 & 0 \\ 0 & \frac{1}{\lambda} - 1 \end{bmatrix} \quad (17)$$

$$B(x^s, u^s) = \begin{bmatrix} 0 \\ \beta \end{bmatrix} \quad (18)$$

C. Fuzzy State Feedback Controller Design

We consider the following fuzzy control law for the fuzzy model

IF $\theta_1(t)$ is μ_{i1} and ... and $\theta_p(t)$ is μ_{ip} THEN

$$u(t) = -F_i x(t), \quad i = 1, 2, \dots, r.$$

The overall state feedback fuzzy control law is represented by

$$u(t) = - \frac{\sum_{i=0}^r w_i(\theta(t)) F_i x(t)}{\sum_{i=1}^r w_i(\theta(t))}$$

D. Fuzzy Rules

Rule 1: If $x_2(t)$ is low (i.e., $x_2(t)$ is about 0.8862)

THEN

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

$$\delta \dot{x}(t) = A_1^1 \delta x(t) + A_2^1 \delta x(t - \tau) + B^1 \delta u(t)$$

$$\delta u(t) = -F_1 \delta x(t)$$

Rule 2: If $x_2(t)$ is Middle (i.e., $x_2(t)$ is about 2.7520)
THEN

$$\delta \dot{x}(t) = A_1^2 \delta x(t) + A_2^2 \delta x(t - \tau) + B^2 \delta u(t)$$

$$\delta u(t) = -F_2 \delta x(t)$$

Rule 3: If $x_2(t)$ is High (i.e., $x_2(t)$ is about 4.7052)
THEN

$$\delta \dot{x}(t) = A_1^3 \delta x(t) + A_2^3 \delta x(t - \tau) + B^3 \delta u(t)$$

$$\delta u(t) = -F_3 \delta x(t)$$

Where,

$$\delta x(t) = x(t) - x_d, \delta x(t - \tau) = x(t - \tau) - x_d,$$

$$\delta u(t) = u(t) - u_d$$

and F_1, F_2, F_3 are to be designed.

Where,

$$A_1^1 = \begin{bmatrix} -1.4274 & 0.0757 \\ -1.4189 & -0.9442 \end{bmatrix} \quad A_1^2 = \begin{bmatrix} -2.0508 & 0.3958 \\ -6.4066 & 1.6168 \end{bmatrix}$$

$$A_1^3 = \begin{bmatrix} -4.5279 & 0.3167 \\ -26.2228 & 0.9837 \end{bmatrix}$$

$$A_2^1 = A_2^2 = A_2^3 = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix}$$

$$B^1 = B^2 = B^3 = \begin{bmatrix} 0 \\ 0.3 \end{bmatrix} \quad F_1 = [-8.4129 \quad 2.3021]$$

$$F_2 = [-3.5455 \quad 2.9766] \quad F_3 = [-18.2132 \quad 3.6272]$$

IV. OPTIMIZATION OF FUZZY RULE USING GENETIC ALGORITHM

Genetic learning and turning of membership functions based on real-coded genetic algorithm (RGA) assume the fuzzy control rules are predefined, which consist of two processes with different goals.

1) The number of labels for each linguistic variable is obtained through genetic learning of predefined rules, and a set of crude fuzzy membership functions are generated for genetic turning.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

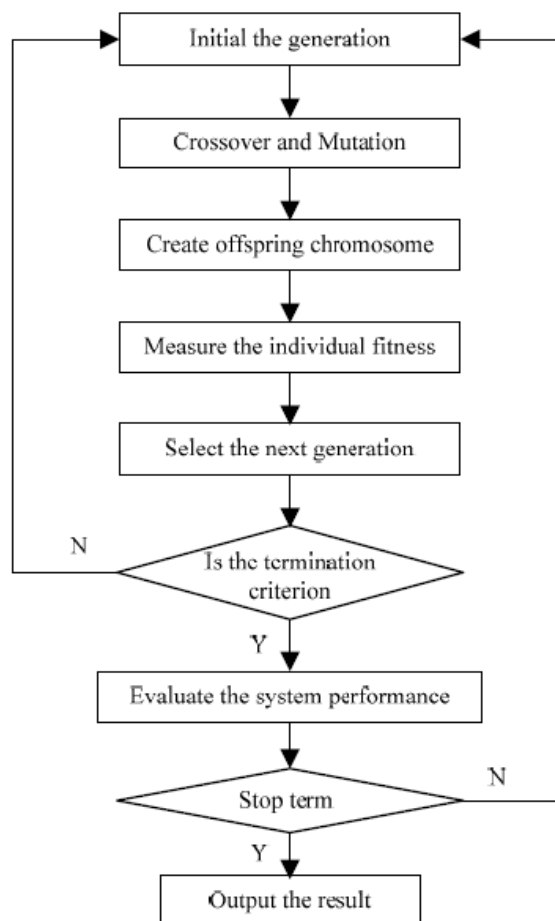
Vol. 5, Issue 7, July 2016

2) The genetic turning process will be apply to adjust the base lengths of the membership functions and the location of its peaks, which are obtained from former process, using evaluation unit to obtain a set of high-performance fuzzy membership functions to the association rules.

Population options specify options for the population of the genetic algorithm.

Population type specifies the type of the input to the fitness function. Types and their restrictions:

- Double vector
- Bit string — For Creation function and Mutation function, use Uniform or Custom. For Crossover function, use Scattered, Single point, Two point, or Custom. You cannot use a Hybrid function or Nonlinear constraint function.
- Custom — For Crossover function and Mutation function, use Custom. For Creation function, either use Custom, or provide an Initial population. You cannot use a Hybrid function or Nonlinear constraint function.



Population size specifies how many individuals there are in each generation. If you set **Population size** to be a vector of length greater than 1, the algorithm creates multiple subpopulations. Each entry of the vector specifies the size of a subpopulation.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

Scaling function specifies the function that performs the scaling. You can choose from the following functions:

- Rank scales the raw scores based on the rank of each individual, rather than its score. The rank of an individual is its position in the sorted scores. The rank of the fittest individual is 1, the next fittest is 2, and so on. Rank fitness scaling removes the effect of the spread of the raw scores.
- reproduction options determine how the genetic algorithm creates children at each new generation.
- Elite count specifies the number of individuals that are guaranteed to survive to the next generation. Set Elite count to be a positive integer less than or equal to Population size.
- Crossover fraction specifies the fraction of the next generation that crossover produces. Mutation produces the remaining individuals in the next generation. Set Crossover fraction to be a fraction between 0 and 1, either by entering the fraction in the text box, or by moving the slider.

Mutation functions make small random changes in the individuals in the population, which provide genetic diversity and enable the genetic algorithm to search a broader space.

Specify the function that performs the mutation in the Mutation function field. You can choose from the following functions:

- Use constraint dependent default chooses:
 - Gaussian if there are no constraints
 - Adaptive feasible otherwise

crossover combines two individuals, or parents, to form a new individual, or child, for the next generation.

Specify the function that performs the crossover in the Crossover function field. You can choose from the following functions:

- Scattered creates a random binary vector. It then selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second parent, and combines the genes to form the child. For example:

- $p1 = [a\ b\ c\ d\ e\ f\ g\ h]$
- $p2 = [1\ 2\ 3\ 4\ 5\ 6\ 7\ 8]$
- random crossover vector =
- $[1\ 1\ 0\ 0\ 1\ 0\ 0\ 0]$

child = [a b 3 4 e 6 7 8]

- Single point chooses a random integer n between 1 and Number of variables, and selects the vector entries numbered less than or equal to n from the first parent, selects genes numbered greater than n from the second parent, and concatenates these entries to form the child. For example:

- $p1 = [a\ b\ c\ d\ e\ f\ g\ h]$
- $p2 = [1\ 2\ 3\ 4\ 5\ 6\ 7\ 8]$
- random crossover point = 3

child = [a b c 4 5 6 7 8]

- Two point selects two random integers m and n between 1 and Number of variables. The algorithm selects genes numbered less than or equal to m from the first parent, selects genes numbered from $m+1$ to n from the second parent, and selects genes numbered greater than n from the first parent. The algorithm then concatenates these genes to form a single gene. For example:

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

- $p1 = [a\ b\ c\ d\ e\ f\ g\ h]$
 - $p2 = [1\ 2\ 3\ 4\ 5\ 6\ 7\ 8]$
 - random crossover points = 3,6
- child = [a b c 4 5 6 g h]

- Intermediate creates children by a random weighted average of the parents. Intermediate crossover is controlled by a single parameter Ratio:
- $child1 = parent1 + \dots$
- $rand * Ratio * (parent2 - parent1)$

If Ratio is in the range [0,1], the children produced are within the hypercube defined by the parents locations at opposite vertices. If Ratio is in a larger range, say 1.1, children can be generated outside the hypercube. Ratio can be a scalar or a vector of length Number of variables. If Ratio is a scalar, all the children lie on the line between the parents. If Ratio is a vector, children can be any point within the hypercube.

Migration is the movement of individuals between subpopulations, which the algorithm creates if you set Population size to be a vector of length greater than 1. Every so often, the best individuals from one subpopulation replace the worst individuals in another subpopulation. You can control how migration occurs by the following three parameters.

Direction specifies the direction in which migration can take place:

- If you set Direction to Forward, migration takes place toward the last subpopulation. That is the nth subpopulation migrates into the (n+1)th subpopulation.
- If you set Direction to Both, the nth subpopulation migrates into both the (n-1)th and the (n+1)th subpopulation.

(MatLab help)

For work we have considered membership function as

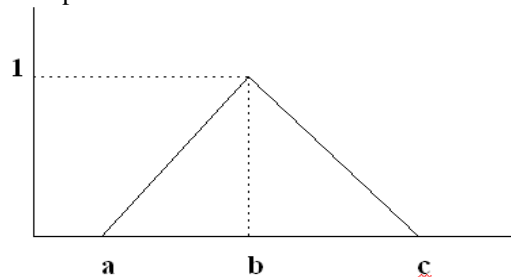


Figure 3. General membership function

For membership function one bounds are as follow,
Lower bound for x (1) and x (2) = [1.8192, 0.8862]
Upper bound for x (1) and x (2) = [2.7520, 1.8192]

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

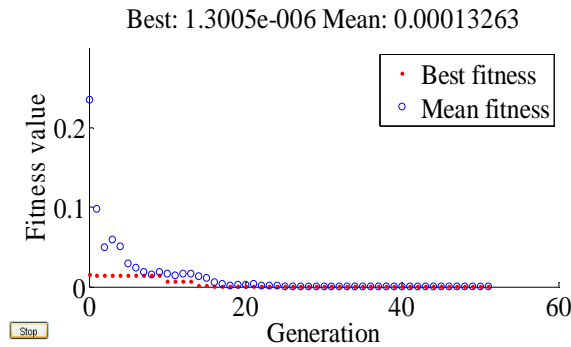


Figure 4. Best Fitness And Mean Fitness Values

From the genetic algorithm optimization tool box we get $x(1) = 2.752$ and $x(2) = 0.886$ also we get the result of best fitness and mean fitness as shown in figure 4

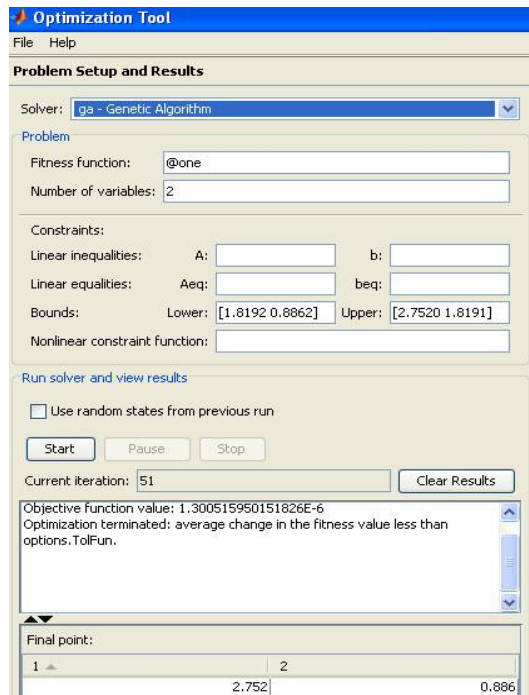


Figure 5. Optimization tool box

So here we get for first rule triangular membership function as

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

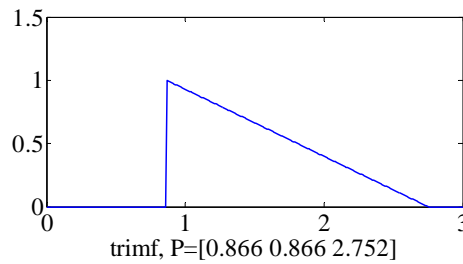


Figure 6. First membership function

Second membership function we can get as

Range for $b = x(2) = [0.8862 \ 1.8191]$

And range for $x = x(1) = [1.8192 \ 2.7520]$

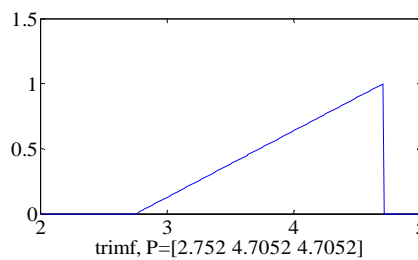


Figure 7. Second membership function

And third membership function as

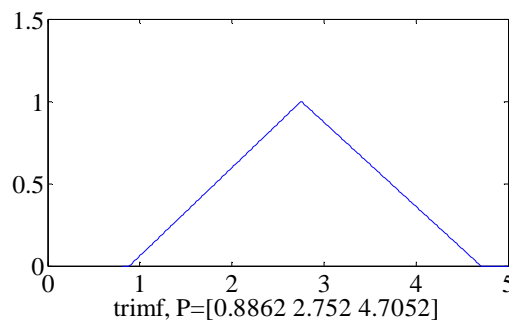


Figure 8. Third membership function

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

This result is for first run of the simulation so we run the simulation for 25 times and check the results for membership functions.

x(1)	x(2)	Objective Function Value	Iteration
2.752	0.886	1.92E-07	51
2.752	0.888	1.92E-07	51
2.752	0.887	5.56E-08	51
2.752	0.887	1.47E-08	51
2.752	0.887	4.87E-07	51
2.752	0.89	1.93E-08	51
2.752	0.888	1.70E-08	51
2.752	0.889	5.82E-09	51
2.752	0.887	4.39E-08	51
2.752	0.886	1.90E-08	51
2.752	0.887	1.96E-07	51
2.752	0.932	5.20E-08	51
2.752	0.886	9.10E-08	51
2.752	0.886	7.19E-07	51
2.752	0.886	7.95E-09	51
2.752	0.887	6.21E-07	51
2.752	0.892	4.17E-08	51
2.752	0.886	3.33E-06	51
2.752	0.886	5.33E-06	51
2.752	0.886	1.71E-08	51
2.752	0.886	1.42E-07	51
2.752	0.886	2.30E-07	51
2.752	0.886	3.40E-06	51
2.752	0.887	1.35E-06	51
2.752	0.886	1.45E-07	51

Figure 9. Base values of first membership function with consecutive 25 run

Below we show bar plot of 25 consecutive runs in which we get 0.886 repeating more time or maximum hits occurs at 0.886 so we consider that point as our base value.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

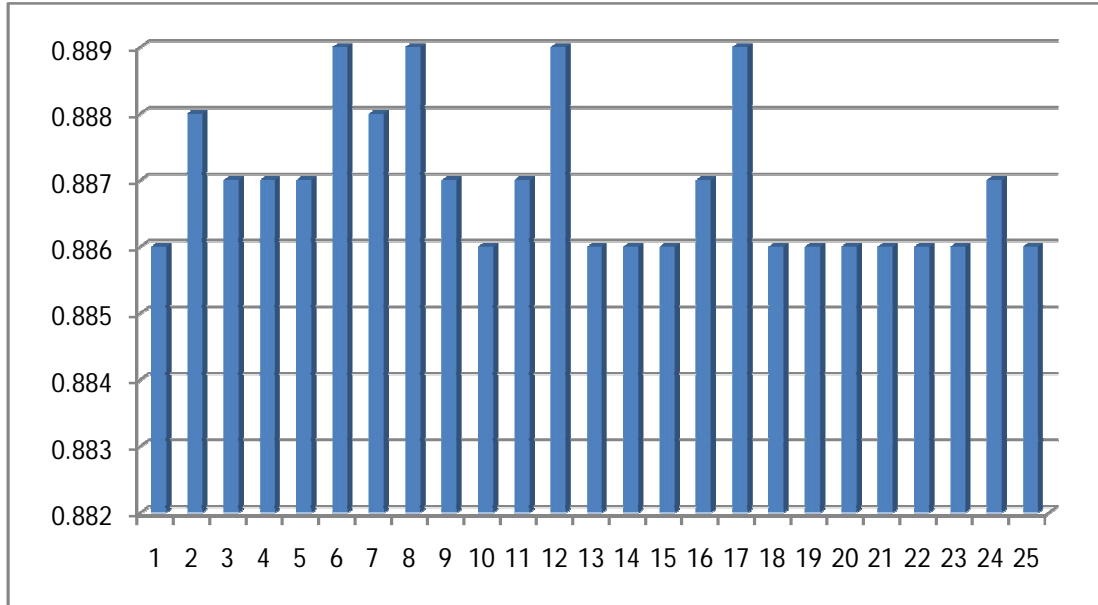


Figure 10. Base values bar chart of first membership function with consecutive 25 run

Base values of second membership function with consecutive 25 run is shown in fig.6.10

x(1)	x(2)	Objective Function Value	Iteration
2.752	4.705	5.12E-05	51
2.752	4.705	5.12E-05	51
2.752	4.611	5.38E-05	51
2.752	4.62	5.35E-05	51
2.752	4.698	5.14E-05	51
2.752	4.699	5.14E-05	51
2.752	4.578	5.48E-05	51
2.752	4.705	5.12E-05	51
2.752	4.661	5.24E-05	51
2.752	4.699	5.14E-05	51
2.752	4.315	6.40E-05	51
2.752	4.705	5.12E-05	51
2.752	4.605	5.40E-05	51
2.752	4.27	6.59E-05	51
2.752	4.703	5.13E-05	51
2.752	4.679	5.19E-05	51
2.752	4.705	5.12E-05	51
2.752	4.663	5.23E-05	51
2.752	4.705	5.12E-05	51
2.752	4.694	5.15E-05	51

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

2.752	4.642	5.29E-05	51
2.752	4.661	5.24E-05	51
2.752	4.565	5.52E-05	51
2.752	4.7	5.13E-05	51
2.752	4.668	5.22E-05	51

Figure 11. Base values of second membership function with consecutive 25 run

Below we show bar plot of 25 consecutive runs in which we get 4.70 repeating more time or maximum hits occurs at 4.70 so we consider that point as our base value.

For first membership function we get base values as follow

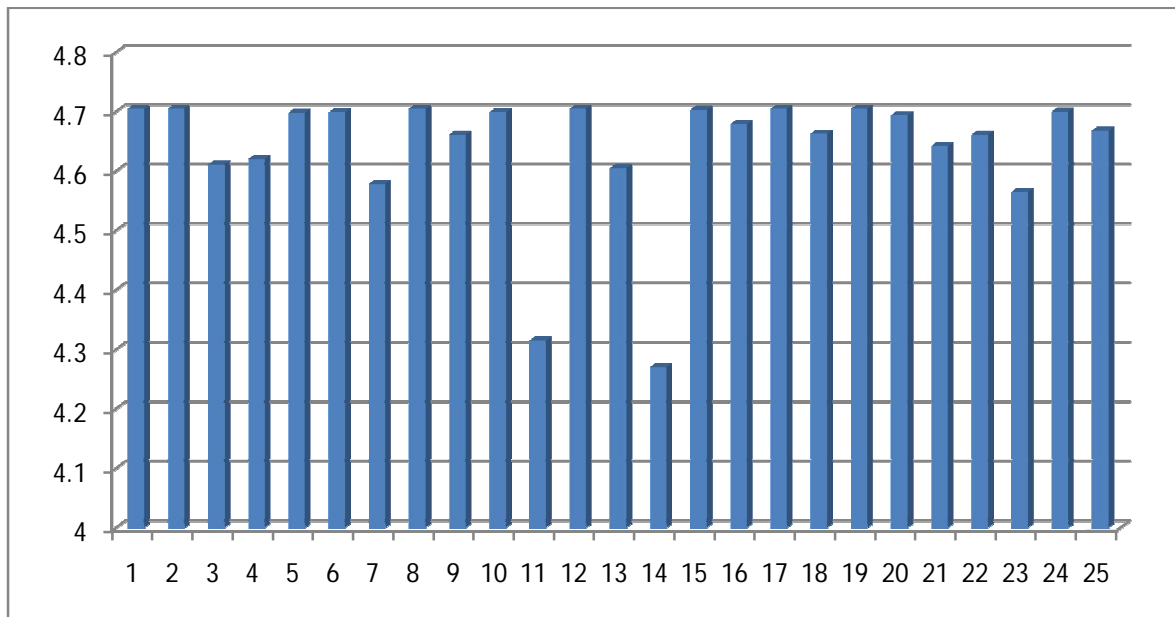
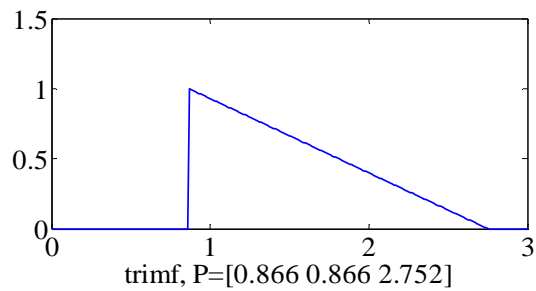


Figure 12. Base values bar chart of second membership function with consecutive 25 run

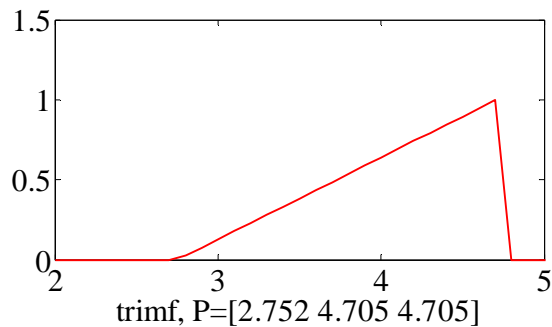
International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

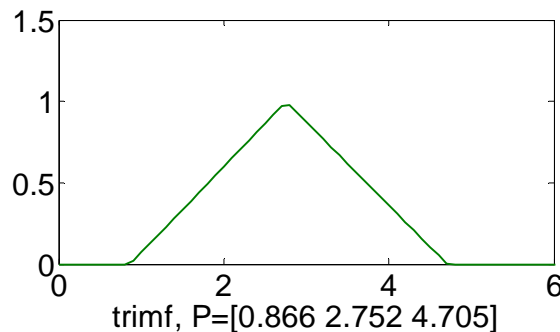
Vol. 5, Issue 7, July 2016

With this run we get one membership function which is repeated many times.

For second membership function we get base values as follow



And third membership function base values are as follow



So overall membership function we will get as follow

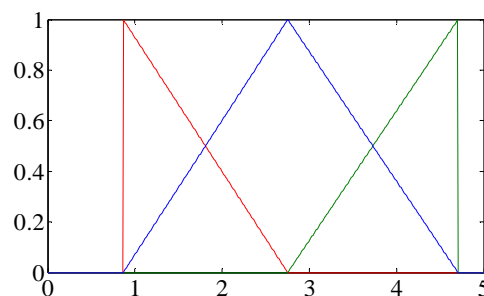


Figure 13. Overall membership function

We choose the overallMembership Function as shown in F

A. Simulation And Result

The SIMULINK model of the Continuously Stirred Tank Reactor (CSTR) is simulated in MATLAB®/SIMULINK environment. The simulation results for different cases using SIMULINK block diagram are given below.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

1) Case-1: When Time Delay is zero

Time Delay = 0, Desired operating Point $x_d(t) = (0.4472, 2.7520)$, Initial Conditions $x_0(t) = (0.4, 2.5)$.

Here in the result open loop response with time delay=0 (fig. 11a) system response is not reaching to the desired operating point (0.4472, 2.7520). Working in closed loop manner (fig.11 b) it reaches to the On the other hand when system is desired operating point (0.4472,2.7520).

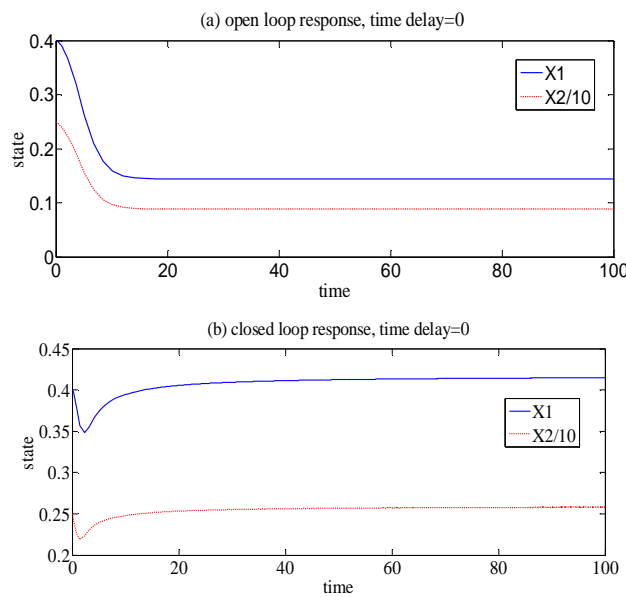
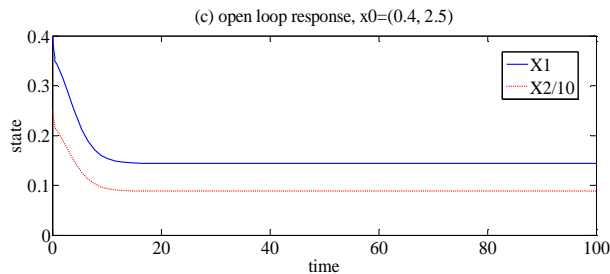


Figure 14. Simulation results CSTR when Time Delay is zero

2) Case-2: When Time Delay is constant

Time Delay = 0.5, Desired operating Point $x_d(t) = (0.4472, 2.7520)$, Initial Conditions $x_0(t) = (0.4, 2.5)$.



When system is having time delay but operates in open loopmanner system response (fig.12 c) is not reaching to thedesired operating point (0.4472, 2.7520). On the other hand when system is working in closed loop manner (fig.12 d) it reaches to the desired operating point (0.4472, 2.7520)

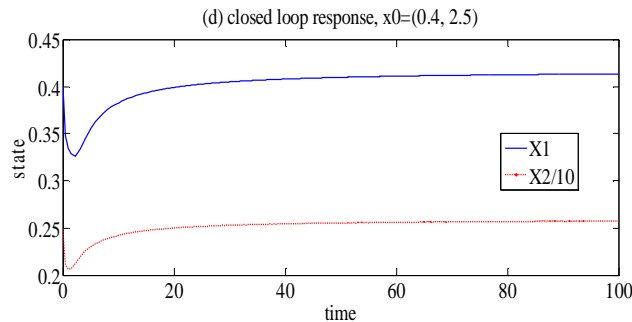


Figure 15. Simulation results CSTR when Time Delay is constant

V. CONCLUSION

Stability analysis and design method for a class of nonlinear Time-Delay systems based on T-S fuzzy modeling and control approach is simulated. First, TS fuzzy models with time delay are extended to describe the nonlinear Time-delay system uncertain nonlinear function. The design methodology is illustrated by application to the stabilization problem of a CSTR example, which is a classical nonlinear Time-Delay system. In this paper, T-S fuzzy approach is used to control nonlinear time delayed system and rules are optimized with the help of genetic algorithm approach. The system is operated in this system in open loop and close loop mode. The results are shown for different initial conditions.

REFERENCES

- [1] Y. Cao and P. Frank, "Analysis and synthesis of nonlinear time-delay systems via fuzzy control approach," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 2, pp. 200–211, Apr.
- [2] J. P. Richard, "Time-delay systems: An overview of some recent advances and open problems," *Automatica*, vol. 39, no. 10, pp. 1667–1694, 2003.
- [3] Y. He, M. Wu, J. She, and G. Liu, "Parameter-dependent Lyapunov functional for stability of time-delay systems with polytopic-type uncertainties," *IEEE Trans. Autom. Control*, vol. 49, no. 5, pp. 828–832, May 2004.
- [4] Slotine, Weiping Li, *Applied Nonlinear Control*. New Jersey, Prentice Hall, 1991.
- [5] W. Assawinchaichote, S. K. Nguang, and P. Shi, *Fuzzy Control and Filter Design for Uncertain Fuzzy Systems*. Berlin, Germany: Springer-Verlag, 2006.
- [6] M. S. Mahmoud and N. F. Al-Muthairi, "Design of robust controllers for time-delay systems," *IEEE Trans. Automat. Contr.*, vol. 39, pp. 995–999, Dec. 1994.
- [7] G. Feng, "A survey on analysis and design of model-based fuzzy control systems," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 5, pp. 676–697, Oct. 2006.
- [8] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan. 1985.
- [9] B. Kosko, "Fuzzy systems as universal approximators," *IEEE Trans. Comput.*, vol. 43, no. 11, pp. 1329–1333, Nov. 1994.
- [10] .2000K. Gu, V. L. Kharitonov, and J. Chen, *Stability of Time-Delay Systems*. Berlin, Germany: Birkhauser, 2003.
- [11] C. Hua, X. Guan, and P. Shi, "Robust stabilization of a class of nonlinear time-delay systems," *Appl. Math. Comput.*, vol. 155, no. 3, pp. 737–752, 2004.
- [12] C. Hua, X. Guan, and P. Shi, "Robust backstepping control for a class of time delayed systems," *IEEE Trans. Autom. Control*, vol. 50, no. 6, pp. 894–899, Jun. 2005.
- [13] X. Guan and C. Chen, "Delay-dependent guaranteed cost control for T-S fuzzy systems with time delays," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 2, pp. 236–249, Apr. 2004.
- [14] X. Guan and C. Chen, "Delay-dependent guaranteed cost control for T-S fuzzy systems with time delays," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 2, pp. 236–249, Apr. 2004.
- [15] C. Lin, Q. Wang, and T. Lee, "Stabilization of uncertain fuzzy time-delay systems via variable structure control approach," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 6, pp. 787–798, Dec. 2005.
- [16] F. Gouaibaut, Y. Blanco, and J. P. Richard, "Robust sliding mode control of non-linear systems with delay: A design via polytopic formulation," *Int. J. Control*, vol. 77, no. 2, pp. 206–215, 2004.
- [17] M. S. Mahmoud and P. Shi, *Methodologies for Control of Jump Time-Delay Systems*. Boston, MA: Kluwer, 2003.
- [18] W. Assawinchaichote, S. K. Nguang, and P. Shi, *Fuzzy Control and Filter Design for Uncertain Fuzzy Systems*. Berlin, Germany: Springer-Verlag, 2006.
- [19] Huai-Xiang Zhang, Feng Wang, Bo Zhang "Genetic Optimization Of Fuzzy Membership Functions", Proceedings of the 2009 International Conference on Wavelet Analysis and Pattern Recognition, Baoding, 12-15 July 2009