Optimization of FUZZY controller by JAYA algorithm

Mr.Jadhav Vaibhav Hari[1], Mr.A.B.Patil[2]

PG Student Dept. of Electrical Engineering WCE Sangli Maharashtra, India
Professor, Dept. of Electrical Engineering WCE Sangli, Maharashtra, India

---

Abstract- *This paper describes a new approach of designing an optimized fuzzy controller, Optimized by Jaya algorithm, In view of this discussion for optimization of Fuzzy controller, we should design a controller for nonlinear time-delay systems so that the performance of the dynamical system can be improved. We aim to use a set of fuzzy rules to describe a global nonlinear system into a set of local time-delay systems with uncertain nonlinear functions and optimization is done by JAYA algorithm*.

Keywords- Nonlinear system Continuous stirrer tank reactor (CSTR), T-S Fuzzy system Type-1 Fuzzy controller, JAYA optimization algorithm

1. INTRODUCTION

Many available system contain nonlinearity characteristics, such as microwave oscillation, chemical process, hydraulic system, etc. It important to study behaviour of nonlinear system. The most important part in nonlinear system is Optimization. The dynamic of non-linear system can be strongly depends on either one or more parameter since their operative condition remain stable only if the value of parameters are must be in specific limit. If these parameter gone out of range then the equilibrium point become unstable. Because of this reason, nonlinear controllers like Fuzzy logic controller are used to control such system because they are more robust than other controllers. [2]Fuzzy technique has been widely and effectively used now a days in nonlinear system modelling and control for more than two decades. In many of the model based fuzzy control approaches, the famous t-s fuzzy model is a popular and convenient tool in functional approximation. [3, 4] CSTR is one of the most commonly used non-linear system, which is mostly used in chemical industries, it offers a verity of researches in the area of chemical and control engineering. Due to non-linearity presents in the system, performance of the conventional controller may not be proper. Hence complexity of the system analysis increases. [6] It becomes difficult to have results under certain conditions. So here Type-1 fuzzy controller is used and optimized, for optimization, Jaya algorithm is used, because it is one of the best optimization algorithm which gives good performance and results. [1] Traditional fuzzy logic controller design is based upon a human operator's experience or control engineer's knowledge. Since this method uses trial and error to find better fuzzy rules and membership function, it is very time consuming method. Also this method doesn't guarantee to have optimum solution or near optimal fuzzy rule and membership function. However by applying JAYA Algorithm in the design process, optimal or near optimal fuzzy rules and membership function can be found without any priory knowledge. The designing of Fuzzy JAYA controller is different than designing of traditional fuzzy control system. In traditional fuzzy control system, the FAM rules and membership functions are determined by operator's knowledge and after that parameters are fixed. But in case of Fuzzy JAYA controller the FAM rules and membership functions are adapted by JAYA algorithm that searches the parameter space for an optimal set of parameters based upon a specific parameters and population size. [4]

The latter part of the paper is arranged in the following sequence. Section 2 presents type 1 fuzzy controller design and optimization for triangular membership functions and optimization by JAYA algorithm is presented in section 3. The results of hardware implementation of controllers are presented in section 4. The main conclusions are reached through analysis of results.

2. FUZZY CONTROLLER DESIGN

Most of the available physical dynamical systems in real life, which are not possible to be represented by linear differential equations and have a nonlinear nature. On another side, linear control methods depends on the key assumption of small range of operation for the linear model and, acquired from linearizing the nonlinear system, to be valid. When the required operation range is large, a linear controller is unstable, because the nonlinearities in the plant cannot be properly dealt with the controller. One more assumption of the linear control is that the system model is definitely linearizable and the linear model is much accurate enough for building up the controller. However, the highly nonlinear and discontinuous nature of many system for example, mechanical and electrical systems does not allow linear approximation practically. As In the process of designing controllers, it is also necessary that the system model is well achievable through a mathematical model and the parameters of the system model are reasonably well-known for controller design. For many practically available nonlinear plants i.e. chemical processes, building a mathematical model with mathematical equation is very difficult and only the input-output data yielded from running the process is accessible for the estimation. Many control problems involve uncertainties like parametric, dynamic etc.

in the model parameters. A controller based on inaccurate or absolute values of the model parameters may show

Significant performance degradation or even instability. The fuzzy model was proposed by Takagi and Sugeno [2] and it is described by fuzzy IF-THEN rules which represent local input-output relations of a nonlinear system. The main feature of a Takagi-Sugeno fuzzy model is to express the local dynamics of each fuzzy implication (rule) by a linear system model. The overall fuzzy model of the system is achieved by fuzzy "blending" of the linear system models. Almost all nonlinear dynamical systems can be represented by Takagi-Sugeno fuzzy models to high degree of precision. In fact, it is proved that Takagi-Sugeno fuzzy models are universal approximation of any smooth nonlinear system.

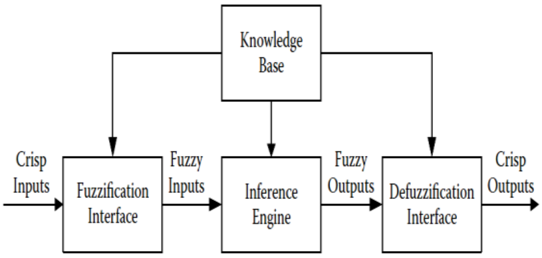


Fig-1: Graphical interpretation of fuzzy

2.1 Takagi-Sugeno fuzzy systems

An alternative type of fuzzy system, known as Takagi–Sugeno (T–S) fuzzy systems, was proposed in Takagi and Sugeno (1985), in an effort to develop a good and easy approach to approximating a nonlinear function. A typical fuzzy rule in a T–S fuzzy system has the form

IF x_1 is A_1 ...., and x_k is A_k,

THEN $y = f(x_1, x_2, \dots . x_k)$

where $x = (x_1, x_2, \dots . x_k)$ and y are linguistic variables, $A_1, \dots, A_k$ are fuzzy sets in the antecedent, and $y = f(x_1, x_2, \dots . x_k)$ is a polynomial in the input variable x, but can be any function as long as it can appropriately describe the output of the system within the region specified by the antecedent of the rule. When $f(x_1, x_2, \dots . x_k)$ is a first-order polynomial, the resulting fuzzy model is called the first-order T–S fuzzy system, which was originally proposed in Takagi and Sugeno (1985) and Sugeno and Kang (1988). If f is a constant then one then has the zero-order T–S fuzzy system, which can be viewed as a special case of the Mamdani type fuzzy system, where each rule's consequence is specified by a fuzzy singleton. The output of a T–S fuzzy system is obtained by the weighted average of the crisp outputs of fuzzy rules. This can avoid the time-consuming procedure of defuzzification. Fig.1.0 gives an example of graphical interpretation of fuzzy reasoning for a T–S fuzzy system with two rules.

2.2 Optimization for triangular membership functions

Here we have to optimize three membership functions one by one using the JAYA optimization technic using MATLAB Code. For that purpose we have to write m files for fitness function of each membership function and decide the bounds of the base for each membership function. This step is important because there are two types of bad membership function. First one type is too redundant and second type is too separated. Due to such membership function we cannot get the desired response or output as we want. Too redundant and too separated membership functions are shown in below Figure,

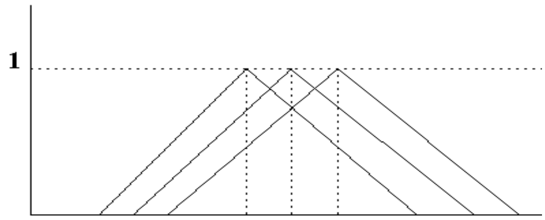


Fig-2: Too redundant membership function

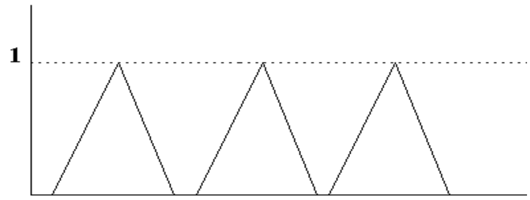


Fig-3: Too separated membership function

For reducing the probability of getting such membership functions the bounds of base are chosen very correctly for each function. Selection of membership function and respective bounds can be done as follow. Triangular membership function can be defined as

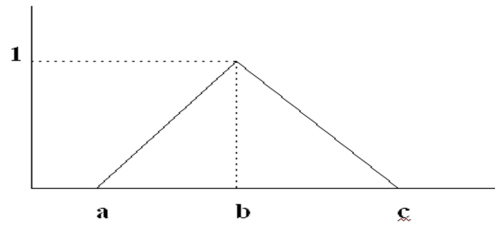


Fig-4: Triangular membership function

$$f(x;a,b,c) = \begin{cases} 0 & x \le a \\ \dfrac{x-a}{b-a} & a \le x \le b \\ \dfrac{c-x}{c-b} & b \le x \le c \\ 0 & c \le x \end{cases}$$

Where a, b, c are the base value of membership function which we are optimizing to get good membership Function. So based on our requirement we choose different fitness function value for each membership function

3. OPTIMIZATION BY JAYA ALGORITHM

A simple, easy and powerful optimization algorithm until now is proposed for solving the constrained and unconstrained both optimization problems. And this algorithm is based on the concept that the solution obtained for a given problem should move towards the best solution if it is profit function and should avoid the Worst solution if it is loss function. This algorithm requires only the common control parameters and does not require any algorithm-specific control parameters unlike other Dr.Rao et al.

The results have proved the better effectiveness of the proposed algorithm as compared with other algorithms. Furthermore, the statistical analysis of the experimental work has been carried out by conducting the Friedman's rank unconstrained and constrained test and Holm-Sidak test. The proposed algorithm is found to secure first rank up till now for the 'best' and 'mean' solutions in the Friedman's rank test for all the 24 constrained and unconstrained benchmark problems. In addition to solving the constrained benchmark problems, the algorithm is also investigated on 30 unconstrained benchmark problems taken from the literature and the performance of the algorithm is found better. [4]

3.1 Proposed algorithm

Let *f*(*x*) is the objective function to be minimized (or maximized). At any iteration *i*, assume that there are 'm' number of design variables (i.e. *j*=1, 2... m), 'n' number of candidate solutions (i.e. population size, *k*=1, 2... n). Let the best candidate *best* obtains the best value of *f* (*x*) (i.e. *f* (*x*) best) in the entire candidate solutions and the worst candidate *worst* obtains the worst value of *f* (*x*) (i.e. *f* (*x*) worst) in the entire candidate solutions. If x_j, *k, i is* the value of the j^{th} variable for the k^{th} candidate during the i^{th}iteration, then this value is modified as per the following Eq. (1).

$$X'_{j,k,i} = X_{j,k,i} + R_{1,k,i}\ (X_{j,best,i} - |X_{j,k,i}|) - R_{2,k,i}\ (X_{j,worst,i} - |X_{j,k,i}|) \quad (1)$$

Where,

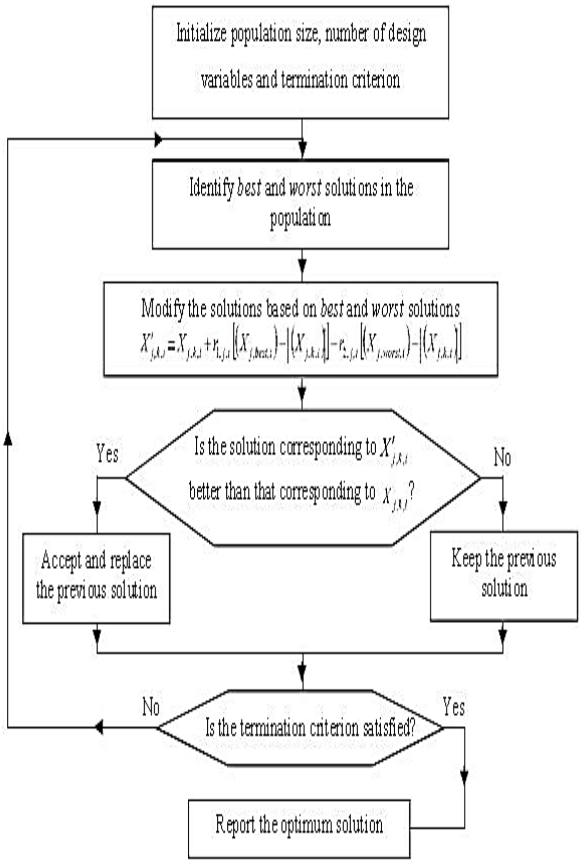
$X_{j,best,i}$ Is the value of the variable *j* for the *best* candidate and $X_{j,worst,i}$ is the value of the variable *j* for the *worst* candidate. $X_{j,k,i}$ Is the updated value of $X_{j,k,i}$ and $R_{1,k,i}$and $R_{2,k,i}$are the two random numbers for the j^{th} variable during the i^{th} iteration in the range [0, 1].

The term "$R_{1,k,i}$ ($X_{j,best,i}$ - $|X_{j,k,i}|$)" Indicates the tendency of the solution to move closer to the best solution and, The term "$R_{2,k,i}$ ($X_{j,worst,i}$- $|X_{j,k,i}|$)"Indicates the tendency of the solution to avoid the worst solution.

$X'_{j,k,i}$ Is accepted if it gives better function value. All the accepted function values at the end of iteration are maintained and these values become the input to the next iteration Dr.Rao et al. Fig.1 shows the flowchart of the proposed algorithm. The algorithm always tries to get closer to success (i.e. reaching the best solution) and tries to avoid failure (i.e. moving away from the worst solution).

The algorithm strives to become victorious by reaching the best solution and hence it is named as **Jaya** (a Sanskrit word meaning **victory**). The proposed method is illustrated by means of an unconstrained benchmark function known as Sphere function in the next section,

3.2 Flowchart of the Jaya algorithm



To demonstrate the working of Jaya algorithm, an unconstrained benchmark function of Sphere is considered. The objective function is to find out the values of *xi* that minimize the value of the Sphere function.

$$\min f(x) = \sum_{i=1}^{n} x_i^2 \qquad (2)$$

Where,

$f(x)$- Objective function to be minimized

4. SIMULATION AND RESULTS

Below table shows that the base value of input membership function for input 1 and input 2

Input 1 (range= -10 to 50)

Table-1: The base values of FUZZY Controller membership functions for input 1

1st Membership function	41.2	-14.8	10.83
2nd Membership function	-10.8	23.6	49.0
3rd Membership function	19.6	50.32	79.40

Input 2(range= 0 to 1.50)

Table-2: The base values of FUZZY Controller membership functions for input 1

1st Membership function	-0.7276	0.006	0.7176
2nd Membership function	0	0.7216	1.447
3rd Membership function	0.7276	1.451	2.173

Below table shows that the Optimized base value of input membership function for input 1 and input 2 by using below equation 1 and random numbers R_1 *and* R_2 .

$$X'_{j,k,i} = X_{j,k,i} + R_{1,k,i}\ (X_{j,best,i} - |X_{j,k,i}|) - R_{2,k,i}\ (X_{j,worst,i} - |X_{j,k,i}|) \qquad (3)$$

$R_1 = 0.49 \qquad R_2 = 0.43$

Input 1 (range= -10 to 50)

Table-3: The Optimized base values of FUZZY JAYA Controller membership functions for input 1

1st Membership function	-41.2	-14.8	25.31
2nd Membership function	-12.3	15.98	49.0
3rd Membership function	14.86	50.03	79.40

Input 2(range= 0 to 1.50)

Table-4: the Optimized base values of FUZZY JAYA Controller membership functions for input 2

1st Membership function	-0.728	0.006	1.24
2nd Membership function	0	0.8103	1.45
3rd Membership function	0.7276	1.450	2.173

As shown above numerically one iteration is completed and optimized base value are shown in above table with the help of MATLAB program of JAYA algorithm with fitness function we optimized the base values of triangular membership function up to 500 iteration and then it gives the optimised base values of all three triangular membership function of Fuzzy JAYA controller as shown in table no. 5.

4.3 Simulation

With step input initial value 50, and the transfer function of CSTR $\frac{0.12}{3s+1}$ as shown in below figure 6 of simulation and optimized fuzzy controllers are added in it.

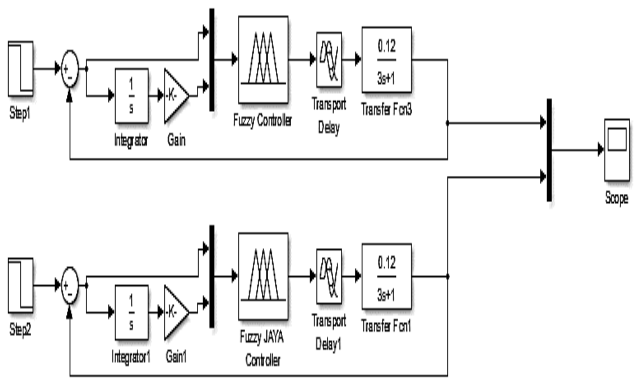


Fig-5: Simulation Block diagram of Fuzzy, Fuzzy JAYA controller

4.4 RESULT

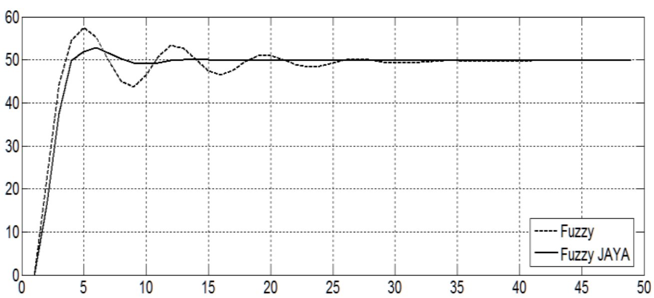


Fig-6: Comparative Step Response of Fuzzy, Fuzzy JAYA controller

4.5 Comparing Parameters

Table-5: Comparing Parameters of Step response

Parameters	FUZZY	JAYA
Rise Time(sec)	2.7673	**3.4764**
Settling Time(sec)	35.37	**19.145**
%Overshoot	15.25	**3.13**
Peak	57.47	**51.433**

5 CONCLUSION

Response of optimized fuzzy controller is smoother than normal fuzzy controller. By response of Optimized JAYA Fuzzy controller Rise time, Settling time and Overshoot is better than normal fuzzy controller.

Table-6: Optimized base values of Fuzzy JAYA controller
NOTE- All values of S.D < 1

MF .1	X1	X2	MF .2	X1	X2	MF .3	X1	X2
1	1.016	0.9	**1**	0.31	1.17	**1**	0.79	1.9
2	1.046	0.99	**2**	0.31	1.17	**2**	0.75	1.91
3	1.046	0.95	**3**	0.311	1.18	**3**	0.77	1.89
4	1.046	0.92	**4**	0.31	1.18	**4**	0.79	1.9
5	1.046	0.95	**5**	0.33	1.18	**5**	0.79	1.9
6	1.016	0.95	**6**	0.31	1.18	**6**	0.79	1.9
7	1.046	0.95	**7**	0.31	1.17	**7**	0.75	1.91
8	1.046	0.93	**8**	0.311	1.18	**8**	0.77	1.89
9	1.046	0.95	**9**	0.31	1.18	**9**	0.79	1.9
10	1.016	0.99	**10**	0.33	1.18	**10**	0.79	1.9
11	1.046	0.95	**11**	0.31	1.18	**11**	0.79	1.9
12	1.046	0.95	**12**	0.31	1.17	**12**	0.75	1.91
13	1.046	0.95	**13**	0.311	1.18	**13**	0.77	1.89
14	1.046	0.96	**14**	0.31	1.18	**14**	0.79	1.9
15	1.046	0.95	**15**	0.33	1.18	**15**	0.79	1.9
16	1.046	0.99	**16**	0.31	1.18	**16**	0.79	1.9
17	1.046	0.95	**17**	0.31	1.17	**17**	0.75	1.91
18	1.046	0.95	**18**	0.311	1.18	**18**	0.77	1.89
19	1.046	0.95	**19**	0.31	1.18	**19**	0.79	1.9
20	1.046	0.95	**20**	0.33	1.18	**20**	0.79	1.9
21	1.046	0.95	**F21**	0.31	1.18	F21	0.79	1.9
22	1.046	0.95	**22**	0.31	1.17	**22**	0.75	1.91
23	1.046	0.95	**23**	0.311	1.18	**23**	0.77	1.89
24	1.046	0.95	**24**	0.31	1.18	**24**	0.79	1.9
25	1.046	0.95	**25**	0.33	1.18	**25**	0.79	1.9
26	1.046	0.95	**26**	0.31	1.18	**26**	0.79	1.9
27	1.046	0.95	**27**	0.31	1.17	27	0.75	1.91
28	1.046	0.95	**28**	0.31	1.18	28	0.77	1.89
Total	**31.23**	**28.53**		**9.426**	**35.3**		**23.3**	**57**
Avg.	**2.018**	**1.840**		**0.608**	**0.27**		**1.50**	**3.67**
S.D	**0.009**	**0.017**		**0.004**	**0.04**		**0.01**	**0.00**

REFERENCES

[1] J. P. Richard, "Time-delay systems: An overview of some recent advances and open problems," Automatica, vol. 39, no. 10, pp. 1667–1694, 2003.

[2] Abhishek Singh, Dr.Veena Sharma "concentration control of CSTR Through Fractional Order PID Controller By using Soft Techniques ",4th ICCNT 2013 July 4-6,2013.

[3] R. Venkata Rao, "Review of applications of TLBO algorithm and a tutorial for beginners to solve the unconstrained and constrained optimization problems." International Journal of Industrial Engineering Computations 7 (2016) 19–34.

[4] Cheng, Y-H. (2015b). A novel teaching-learning based optimization for improved mutagenic primer design in mismatch PCR-RFLP SNP genotyping. IEEE/ACM Transactions on Computational Biology and Bioinformatics, doi:10.1109/tcbb.2015.2430354.

[4] R. Venkata Rao, "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems." International Journal of Industrial Engineering Computations 7 (2015).

[5] R. Venkata Rao, "Decision making in the manufacturing environment using Graph theory and Fuzzy multiple attribute decision making Methods". 2016(Book).

[6] Rudolf Seising, "When Computer Science Emerged and Fuzzy Sets Appeared The Contributions of Lotfi A. Zadeh and Other Pioneers", 2016(Book).

[7] Yun Li, Kiam Heong Ang and Gregory C Y Chong, "PID control system analysis and design," IEEE Control Systems Magazine, pp. 32-41, 2006.