

# Optimal control for a linear quadratic neuro Takagi–Sugeno fuzzy singular system using genetic programming

N. Kumaresan<sup>1</sup>

Kuru Ratnavelu<sup>2</sup>

(Received 17 December 2013; revised 16 January 2015)

## Abstract

Optimal control for a linear neuro Takag–Sugeno fuzzy singular system with quadratic performance is obtained using genetic programming (GP). To obtain the optimal control, the solution of a matrix Riccati differential equation is computed by solving a differential algebraic equation using the GP approach. The obtained solution is equivalent or very close to the exact solution of the problem. The accuracy of the solution computed by the GP approach is qualitatively better than the traditional Runge–Kutta method. An illustrative numerical example is presented for the proposed method.

---

<http://journal.austms.org.au/ojs/index.php/ANZIAMJ/article/view/7806> gives this article, © Austral. Mathematical Soc. 2015. Published February 13, 2015, as part of the Proceedings of the 11th Biennial Engineering Mathematics and Applications Conference. ISSN 1446-8735. (Print two pages per sheet of paper.) Copies of this article must not be made otherwise available on the internet; instead link directly to this URL for this article.

# Contents

<b>1</b>	<b>Introduction</b>	<b>C514</b>
<b>2</b>	<b>Statement of the problem</b>	<b>C515</b>
2.1	Neuro fuzzy optimal control system . . . . .	C517
<b>3</b>	<b>Genetic programming method</b>	<b>C518</b>
3.1	Initialization of the population . . . . .	C518
3.2	Grammatical evolution . . . . .	C518
3.3	Genetic operators . . . . .	C520
3.4	Termination control . . . . .	C521
3.5	Genetic programming algorithm . . . . .	C521
<b>4</b>	<b>Numerical example</b>	<b>C522</b>
<b>5</b>	<b>Conclusion</b>	<b>C522</b>
	<b>References</b>	<b>C523</b>

## 1 Introduction

A neuro fuzzy system is a hybrid approach in which a fuzzy system is trained using techniques similar to those applied to neural networks. One of the first neuro fuzzy systems was the adaptive network-based fuzzy inference system (ANFIS) [4]. The ANFIS represents a Takagi–Sugeno (TS) fuzzy system as a multilayer feed-forward network which is trained via a back-propagation algorithm. Neural networks and fuzzy systems are combined to access the advantages of both systems and to avoid their individual weaknesses. The neuro fuzzy system contains a mixture of algebraic and differential equations. The algebraic equations represent the constraints on the solution of the differential part.

As the theory of optimal control of linear systems with quadratic performance criteria is well developed, it is used in many practical design problems [1]. The quadratic cost control problem is a more interesting problem since stability is guaranteed when the system has all of its states available for feedback. Da Prato and Ichikawa [2] showed that the optimal feedback control and the minimum cost are characterized by the solution of a Riccati equation. Solving the matrix Riccati differential equation (MRDE) is the central issue in optimal control theory. The problem is stated in Section 2.

Genetic programming (GP) is an evolutionary algorithm that attempts to evolve solutions of the given problem by using concepts taken from naturally occurring evolving processes. The technique uses the evolution of a large number of candidate solutions through genetic operations such as reproduction, crossover and mutation. It is based upon the genetic algorithm (GA) [3], which exploits natural selection based on a fitness measure to breed a population of trial solutions that improve over time. Although parallel algorithms compute solutions faster than sequential algorithms, there are no reports on GP solutions for MRDE [7]. This article focuses on the implementation of a GP approach for solving MRDE and obtaining the optimal solution. Section 3 presents the solution of the MRDE. In Section 4, an example is given to illustrate the advantage and accuracy of the GP solution by comparing the Runge–Kutta (RK) solution.

## 2 Statement of the problem

In this section an ANFIS class of adaptive networks is proposed. For simplicity, the fuzzy inference system has two inputs,  $x_1$  and  $x_2$ , and output  $[f_1, f_2]$ . The rule base contains two fuzzy if-then TS-type rules [8]:

Rule 1 : if  $x_1$  is  $A_1$  and  $x_2$  is  $B_1$ , then  $f_1 = p_1x_1 + q_1x_2 + r_1$  ;

Rule 2 : if  $x_1$  is  $A_2$  and  $x_2$  is  $B_2$ , then  $f_2 = p_2x_1 + q_2x_2 + r_2$  .

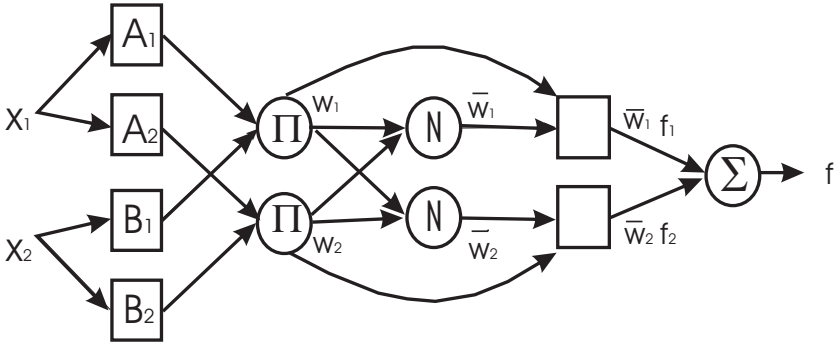


Figure 1: Two rule fuzzy reasoning.

Figure 1 shows the architecture of fuzzy reasoning ANFIS with two rules. The node functions in the same layer are of the same function family, as described by Kumaresan [7]. The input variables  $x_j$  are fuzzified as fuzzy variables with Gaussian membership function.

The corresponding output for the neural network [10] is, for  $r$  number of rules,

$$D\mathbf{x}(t) = A_i\mathbf{x}(t) + B_i\mathbf{u}(t), \quad i = 1, \dots, r. \quad (1)$$

Here,  $\mathbf{x}(t) \in \mathbb{R}^n$  is a generalized state vector with components  $x_j$ ,  $n$  is the number of state variables,  $\mathbf{u}(t) \in \mathbb{R}^m$  is a control variable,  $A_i \in \mathbb{R}^{n \times n}$  and  $B_i \in \mathbb{R}^{n \times m}$  are coefficient matrices associated with  $\mathbf{x}(t)$  and  $\mathbf{u}(t)$ , respectively, and  $D\mathbf{x}(t)$  denotes  $E\dot{\mathbf{x}}(t)$  for the continuous case and  $E\mathbf{x}(t+1)$  for the discrete case.

The proposed neuro fuzzy network has a neural-based linear TS fuzzy modelling structure. Using the neural learning technique, this TS structure will proceed the structure and parameter learning and generate the following linear TS fuzzy system with

Rule  $i$ : if  $x_j$  is  $T_{ji}(m_{ji}, \sigma_{ji})$  for  $i = 1, \dots, r$  and  $j = 1, \dots, n$ , then  $D\mathbf{x}(t) = A_i\mathbf{x}(t) + B_i\mathbf{u}(t)$ .

Here,  $T_{ji}$  is a fuzzy variable with Gaussian membership function,  $m_{ji}$  and  $\sigma_{ji}$  are the mean and standard deviation of the Gaussian membership function, respectively.

## 2.1 Neuro fuzzy optimal control system

Consider the linear dynamical neuro TS fuzzy singular system with

$$\begin{aligned} \text{Rule } i: & \text{ if } x_j \text{ is } T_{ji}(m_{ji}, \sigma_{ji}) \text{ for } i = 1, \dots, r \text{ and } j = 1, \dots, n, \text{ then} \\ E_i \dot{\mathbf{x}}(t) &= A_i \mathbf{x}(t) + B_i \mathbf{u}(t), \quad \mathbf{x}(0) = \mathbf{x}_0, \quad t \in [0, t_f], \end{aligned} \quad (2)$$

where the matrix  $E_i$  is singular,  $\mathbf{x}_0$  is a given initial state vector and  $m \leq n$ .

To minimize both state and control signals of the feedback control system, a quadratic performance index is minimized:

$$J = \frac{1}{2} \mathbf{x}^T(t_f) E_i F E_i \mathbf{x}(t_f) + \frac{1}{2} \int_0^{t_f} [\mathbf{x}^T(t) Q \mathbf{x}(t) + \mathbf{u}^T(t) R \mathbf{u}(t)] dt, \quad (3)$$

where the superscript  $T$  denotes the transpose operator,  $F, Q \in \mathbb{R}^{n \times n}$  are symmetric and positive definite (or semidefinite) weighting matrices for  $\mathbf{x}(t)$ , and  $R \in \mathbb{R}^{m \times m}$  is a positive definite weighting matrix for  $\mathbf{u}(t)$ . It is assumed that  $|sE_i - A| \neq 0$  for some  $s$ . This assumption guarantees that any input  $\mathbf{u}(t)$  will generate one and only one state trajectory  $\mathbf{x}(t)$ .

If all state variables are measurable, then a linear state feedback control law

$$\mathbf{u}(t) = -R^{-1} B_i^T K_i(t) E_i \mathbf{x}(t),$$

is obtained for the system described by equation (2), where  $K(t) \in \mathbb{R}^{n \times n}$  is a symmetric matrix and the solution of MRDE. The MRDE for the linear neuro fuzzy system (2) is

$$E_i^T \dot{K}_i(t) E_i + E_i^T K_i(t) A_i + A_i^T K_i(t) E_i + Q - E_i^T K_i(t) B_i R^{-1} B_i^T K_i(t) E_i = 0, \quad (4)$$

with terminal condition  $K_i(t_f) = E_i^T F E_i$ .

## 3 Genetic programming method

GP is used to obtain a set of expressions. If the required number of expressions satisfy the fitness function, then they will be the optimal solution of MRDE.

### 3.1 Initialization of the population

The first step is to initialize the population. An initial population of the desired size is generated randomly. The length of each chromosome  $\mathbf{x}(t)$  in the population is set according to the nature of the problem.

### 3.2 Grammatical evolution

Grammatical evolution is an evolutionary algorithm that produces code in any programming language. The algorithm starts from the start symbol of the grammar and gradually creates the program string by replacing non terminal symbols with the right hand side of the selected production rule. First read an element from the chromosome and compute  $\text{Rule} = V \bmod N_R$ , where  $V$  is the value of the chromosome and  $N_R$  is the number of rules for the specific non terminal symbol which is shown in Table 1. The symbol  $S$  in the grammar denotes the start symbol of the grammar.

For example, consider the chromosome  $\mathbf{x} = \{7, 2, 10, 4, 4, 2, 11, 16, 30, 5\}$ . The method of producing a valid expression is shown in Table 2. The  $\text{Rule} = V \bmod N_R$  is applied in each row of Table 2. In first line  $V = 7$  and  $N_R = 7$  (number of elements in  $\langle \text{exp} \rangle$  group in Table 1). In the next line is  $\langle \text{exp} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle$  which, from Table 1, is 0. Similarly the remaining rows of Table 2 are found. The function which corresponds to the chromosome is  $\text{exp}(x) + \log[\text{exp}(y)]$ . O'Neil and Ryan [6] and Tsoulos and Lagaris [9] provide further details.

Table 1: The grammar of the proposed method.

$S :=$	$\langle \text{exp} \rangle$	(0)
$\langle \text{exp} \rangle :=$	$\langle \text{exp} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle$	(0)
	$(\langle \text{exp} \rangle)$	(1)
	$\langle \text{fun} \rangle (\langle \text{exp} \rangle)$	(2)
	$\langle \text{pre-op} \rangle (\langle \text{exp} \rangle)$	(3)
	$\langle \text{digit} \rangle$	(4)
	$x$	(5)
	$y$	(6)
	$z$	(7)
$\langle \text{op} \rangle :=$	$+$	(0)
	$-$	(1)
	$*$	(2)
	$/$	(3)
$\langle \text{fun} \rangle :=$	$\sin$	(0)
	$\cos$	(1)
	$\exp$	(2)
	$\log$	(3)
$\langle \text{digit} \rangle :=$	$0$	(0)
	$1$	(1)
	$2$	(2)
	$3$	(3)
	$4$	(4)
	$5$	(5)
	$6$	(6)
	$7$	(7)
	$8$	(8)
	$9$	(9)

Table 2: Example of construction of expression using BNF grammar.

String	Chromosome	Operation
$\langle \text{exp} \rangle$	7, 2, 10, 4, 4, 2, 11, 16, 30, 5	$7 \bmod 7 = 0$
$\langle \text{exp} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle$	2, 10, 4, 4, 2, 11, 16, 30, 5	$2 \bmod 7 = 2$
$\text{fun}(\langle \text{exp} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle)$	10, 4, 4, 2, 11, 16, 30, 5	$10 \bmod 4 = 2$
$\text{exp} \langle \text{exp} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle$	4, 4, 2, 11, 16, 30, 5	$4 \bmod 7 = 4$
$\text{exp} \langle \mathbf{x} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle$	4, 2, 11, 16, 30, 5	$4 \bmod 4 = 0$
$\text{exp} \langle \mathbf{x} \rangle + \langle \text{exp} \rangle$	2, 11, 16, 30, 5	$2 \bmod 7 = 2$
$\text{exp} \langle \mathbf{x} \rangle + \text{fun} \langle \text{exp} \rangle$	11, 16, 30, 5	$11 \bmod 4 = 3$
$\text{exp} \langle \mathbf{x} \rangle + (\log)(\langle \text{exp} \rangle)$	16, 30, 5	$16 \bmod 7 = 2$
$\text{exp} \langle \mathbf{x} \rangle + \log(\text{fun} \langle \text{exp} \rangle)$	30, 5	$30 \bmod 4 = 2$
$\text{exp} \langle \mathbf{x} \rangle + \log(\text{exp} \langle \text{exp} \rangle)$	5	$5 \bmod 7 = 5$
$\text{exp} \langle \mathbf{x} \rangle + \log(\text{exp} \langle \mathbf{y} \rangle)$		

The *fitness function* provides a basis for competition among available solutions and to obtain the optimal solution. Hence, the fitness function for MRDE is defined as

$$E_r = \{k_{1n}(t_l) - \varphi[k_{ij}(t_l)]\}^2 + \sum_{i,j=1}^{n-1} \left\{ \dot{k}_{ij}(t_l) - \Phi_{ij}[k_{ij}(t_l)] \right\}^2, \quad (5)$$

where  $l = 0, 1, 2, \dots, t_f$  represents the equidistance points in the relevant range  $[0, t_f]$  and the  $k_{ij}$  are elements of matrix  $K$ .

### 3.3 Genetic operators

The genetic operators reproduction, crossover and mutation are now defined. In a *reproduction* process, the best chromosomes in a population, according to their fitness values, are probabilistically assigned a large number of copies. No new strings are formed in the reproduction phase. Koza [5] allowed ten percent of the population to reproduce. The *crossover* is applied every



generation to create new chromosomes which replace the worst individuals in the population. There are many types of crossover operators. We use a single point crossover technique. In the *mutation* process, for every element in a chromosome, a random number in the range  $[0, 1]$  is chosen. If the number is less than or equal to the mutation rate, then the corresponding element is changed randomly, otherwise it remains unchanged.

### 3.4 Termination control

In each generation, a set of expressions are generated by the chromosomes. If an expression minimizes the fitness function  $E_r$  (5) to zero or very close to zero and satisfies the initial condition, then the process is stopped; otherwise, the GP approach continues.

### 3.5 Genetic programming algorithm

The algorithm has the following steps.

1. Initialize the random population.
2. Create a valid function using the grammar.
3. Evaluate the fitness value of the chromosome.
4. If the fitness tends to zero, then stop the procedure, otherwise proceed to next step.
5. Generate a new population using genetic operations and then go to Step 2.

## 4 Numerical example

Consider the optimal control problem: minimize  $J$ , defined in equation (3), subject to the neuro TS fuzzy singular system Rule  $i$  with  $r = 2$  and  $n = 1$  so that  $j = 1$  and  $i = 1, 2$ . Also,

$$E = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad S = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}, \quad A_1 = \begin{bmatrix} -1 & -1 \\ 0 & 1 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} -2 & -2 \\ 0 & 2 \end{bmatrix}, \quad B_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad R = 1, \quad Q = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

The numerical implementation proceeds with  $t_f = 2$ . Then the optimal control of the fuzzy system is found from the solution of MRDE.

To solve the MRDE, each chromosome is split uniformly into 30 parts and the population size is 100. In the computation process, the replication rate is 0.01, the crossover probability is 0.9 and the mutation rate is 0.05. After 150 generations,

$$k_{11} = \frac{4}{-1 + 5e^{-4t+8}}, \quad k_{12} = \frac{4}{-1 - 3e^{-4t+8}},$$

which satisfy the fitness function  $E_r$  (5), and the value of the fitness function tends to zero, thus satisfying the termination condition. Hence the solution of the MRDE is obtained.

Table 3 shows the numerical solutions of the MRDE, calculated using the GP and RK methods. Since this problem has an explicit solution, the GP solution is equivalent to the exact solution.

## 5 Conclusion

The optimal control for the linear neuro TS fuzzy singular system is obtainable with the GP approach. To obtain the optimal control, the MRDE solution is

Table 3: Solutions of MRDE.

t	RK solution		GP solution	
	k <sub>11</sub>	k <sub>12</sub>	k <sub>11</sub>	k <sub>12</sub>
0.0	0.0002	-0.0004	0.0002	-0.0004
0.4	0.0014	-0.0022	0.0013	-0.0022
0.8	0.0068	-0.0112	0.0066	-0.0109
1.2	0.0339	-0.0543	0.0329	-0.0536
1.6	0.1715	-0.2532	0.1683	-0.2522
2.0	1.0000	-1.0000	1.0000	-1.0000

computed using a novel and nontraditional GP approach. For a numerical example, the accuracy of the solution computed with the GP approach is qualitatively better than the RK solution. In this example, the obtained GP solution is equivalent to the exact solution.

**Acknowledgements** The authors are very thankful to the referees for their valuable comments and suggestions for improving this manuscript. The funding of this work by the HIR grant (account no:UM.C/625/1/HIR/MOHE/SC/13) and UMRG grant (account no:RG099/10AFR) is gratefully acknowledged.

## References

- [1] P. Balasubramaniam, J. Abdul Samath, N. Kumaresan and A. Vincent Antony Kumar, Solution of matrix Riccati differential equation for the linear quadratic singular system using neural networks, *Appl. Math. Comput.* 182(2):1832–1839, 2006. doi:[10.1016/j.amc.2006.06.020](https://doi.org/10.1016/j.amc.2006.06.020) C515

- [2] G. Da Prato and A. Ichikawa, Quadratic control for linear periodic systems, *Appl. Math. Opt.* 18:39–66, 1988. doi:10.1007%2FBF01443614 C515
- [3] D. E. Goldberg, *Genetic algorithms in search, optimization and machine learning*, Addison Wesley, 1989.  
<http://dl.acm.org/citation.cfm?id=534133> C515
- [4] J. Jang, ANFIS: adaptive-network-based fuzzy inference systems, *IEEE T. Syst. Man. Cyb.* 23(3):665–685, 1993. doi:10.1109/21.256541 C514
- [5] J. R. Koza, *Genetic programming: on the programming of computers by means of natural selection*. MIT Press, 1992.  
<http://mitpress.mit.edu/books/genetic-programming> C520
- [6] M. O’Neill and C. Ryan, Evolutionary automatic programming in an arbitrary language, *Genetic Programming, Vol. 4*, Kluwer Academic Publishers, 2003.  
<http://www.springer.com/computer/ai/book/978-1-4020-7444-8> C518
- [7] N. Kumaresan, Optimal control for stochastic linear quadratic singular periodic neuro Takagi–Sugeno fuzzy system with singular cost using ant colony programming, *Appl. Math. Model.*, 35:3797–3808, 2011. doi:10.1016/j.apm.2011.02.017 C515, C516
- [8] T. Takagi and M. Sugeno, Derivation of fuzzy control rules from human operator’s actions, *IFAC-IFIP-IFORS Symp., Fuzzy information, knowledge representation and decision analysis*, 55–60, 1983.  
<http://dl.acm.org/citation.cfm?id=577582> C515
- [9] I. G. Tsoulas and I. E. Lagaris, Solving differential equations with genetic programming, *Genet. Program. Evol. M.*, 7:33–54, 2006. doi:10.1007/s10710-006-7009-y C518

- [10] S.-J. Wu, H.-H. Chiang, H.-T. Lin and T.-T. Lee, Neural-network-based optimal fuzzy controller design for nonlinear systems, *Fuzzy Set. Syst.*, 154:182–207, 2005. doi:[10.1016/j.fss.2005.03.011](https://doi.org/10.1016/j.fss.2005.03.011) C516

## Author addresses

1. **N. Kumaresan**, Institute of Mathematical Sciences, Faculty of Science, University of Malaya, Kuala Lumpur 50603, Malaysia.  
<mailto:drnk2008@gmail.com>, [drnk2008@um.edu.my](mailto:drnk2008@um.edu.my)
2. **Kuru Ratnavelu**, Institute of Mathematical Sciences, Faculty of Science, University of Malaya, Kuala Lumpur 50603, Malaysia.