Non-Interactive Fuzzy Rule-Based Systems

A. Lotfi, M. Howarth

Manufacturing Automation Research Group

Department of Manufacturing Engineering

The Nottingham Trent University

Burton Street, Nottingham, NG1 4BU

England

Email: lotfia@marg.ntu.ac.uk

Abstract

In this paper, we have introduced a non-interactive model for fuzzy rule-based systems. A critical aspects of this non-interactive model is the introduction of a new set of rules with fewer parameters and without considering the interaction between the functionality of inputs. The new non-interactive model of the fuzzy rule-based system represents the output as a linear combination of the non-linear function of individual inputs.

1 Introduction

A fuzzy rule based system (FRBS) normally consists of the following components:

- A fuzzification stage: this stage outputs a fuzzy value when the input is a crisp value. This is performed by passing the crisp input into a membership function (MF). The fuzzy or linguistic values are defined in an appropriate universe of discourse.
- An inference stage: the major function of this stage is to combine the output of the fuzzification stage together. There are a number of methods for combining the variables, e.g. multiplication.
- A defuzzification stage: the output of the inference stage is converted back into crisp value.

When the FRBS contains multi-inputs for each rule, the MFs of individual inputs must be combined together. This combination is usually performed using the conjunctive operator "*and*" [5]. In this paper, we propose a non-interactive model for FRBSs; a method whereby the "*and*" function (usually multiplication) can be removed completely. The power of our new approach becomes apparent when it is used in a situation where the FRBS with multi-inputs is not very interactive or when there are too many rules in the FRBS. The new non-interactive model of the FRBS represents the output as a linear combination of the non-linear function of individual inputs. At the same time the number of consequent parameters will decrease.

The organization of the rest of the paper is as follows: fuzzy if-then rules and related inference mechanisms will be briefly described in Section 2. This is followed by a non-interactive model for FRBS for a simple two-input one-output system in Section 3 to set the background for the proposed method. An artificial neural network (ANN) structure of the FRBS, and subsequent explanation for different layers of the ANN will be introduced in Section 4. The general non-interactive model of FRBS is given in Section 5. Some numerical examples are presented in Section 6 and conclusions will be drawn in Section 7.

2 Fuzzy Rule-Based Systems

In this paper, fuzzy if-then rules of the following configuration are employed for the modeling of linguistic information.

R^i: If x_1 is $\tilde{A}_1^k$ and ... x_j is $\tilde{A}_j^k$ ... and x_p is $\tilde{A}_p^k$ then y is B^i $\star$

where R^i is the label of i^{th} rule, $x_j : \ j = 1, 2, \ldots, p$ is the j^{th} input variable, y is the output, $\tilde{A}_j^k$ ($j = 1, 2, \ldots, p$ and $k = 1, 2, \ldots, K_j$) is a fuzzy set, and B^i is a real number. n and p are the numbers of rules and individual inputs, respectively. The superscript $\sim$ represents the fuzzy values in contra-distinction to crisp values. The number of individual MFs for a specific input value x_j $(\tilde{A}_j^1, \tilde{A}_j^2, ...\tilde{A}_j^{K_j})$ is K_j. Note that $K_j \leq n$. In this paper, the MF for the fuzzy values, $\tilde{A}_j^k$, is defined by a Gaussian function as follows:

$$\tilde{A}_j^k = \exp\left(-\left(\frac{x_j - \rho_j^k}{\sigma_j^k}\right)^2\right) \quad j = 1, \ldots, p; \quad k = 1, \ldots, K_j \tag{1}$$

where σ_j^k and ρ_j^k are unknown constant parameters. These parameters can be adjusted on-line using a gradient descent algorithm. We further assume that the universe of antecedent is limited to a specific domain interval, i.e. $x_j \in [U_j^-, \ \ U_j^+], \ \ j = 1, \ldots, p$.

The decision, $y(t)$, at the t^{th} instant, as a function of inputs $x_j(t) : \ j = 1, 2, \ldots, p$, is given in the following equation [8]:

$$y(t) = \frac{\sum_{i=1}^{n} B^i w^i}{\sum_{i=1}^{n} w^i} \tag{2}$$

where B^i is the consequent parameters and w^i is the rule firing strength given by:

$$w^i = \prod_{j=1}^{p} \tilde{A}_j^k(x_j(t)) \quad k = 1, 2, \ldots, K_j \tag{3}$$

3 Non-Interactive Model For Fuzzy Rule-Based Systems

In order to emphasis and to clarify the basic idea of our new concept of non-interactive model for FRBSs, a simple two-input one-output FRBS is employed. Extension to multi-input one-output case is straightforward and will be explained in the pertinent sections.

Suppose we are given a FRBS with four rules, $n = 4$, two individual MFs for the first input, x_1, and two individual MFs for the second input, x_2, i.e. $K_1 = K_2 = 2$. The rules are of the following form:

R^1: If x_1 is $\tilde{A}_1^1$ and x_2 is $\tilde{A}_2^1$ then y is B^1

R^2: If x_1 is $\tilde{A}_1^2$ and x_2 is $\tilde{A}_2^1$ then y is B^2

R^3: If x_1 is $\tilde{A}_1^1$ and x_2 is $\tilde{A}_2^2$ then y is B^3

R^4: If x_1 is $\tilde{A}_1^2$ and x_2 is $\tilde{A}_2^2$ then y is B^4

Using the expression (2) for calculating the output $y(t)$ we have:

$$y(t) = \frac{B^1 w^1 + B^2 w^2 + B^3 w^3 + B^4 w^4}{w^1 + w^2 + w^3 + w^4} \tag{4}$$

where $w^1 = \tilde{A}_1^1(x_1)\tilde{A}_2^1(x_2)$, $w^2 = \tilde{A}_1^2(x_1)\tilde{A}_2^1(x_2)$, $w^3 = \tilde{A}_1^1(x_1)\tilde{A}_2^2(x_2)$ and $w^4 = \tilde{A}_1^2(x_1)\tilde{A}_2^2(x_2)$.

We substitute the consequent parameters B^i $(i = 1, 2, 3, 4)$ with four variables C_1^1, C_1^2, C_2^1 and C_2^2 defined as follows:

$$\begin{cases} C_1^1 + C_2^1 &= B^1 \\ C_1^2 + C_2^1 &= B^2 \\ C_1^1 + C_2^2 &= B^3 \\ C_1^2 + C_2^2 &= B^4 \end{cases} \tag{5}$$

It is to be noted that the new set of variables are defined in correlation with the firing strength, w^i, for each rule. For instance, the first firing strength is $w^1 = \tilde{A}_1^1(x_1)\tilde{A}_2^1(x_2)$ and the consequent variable B^1 is replaced with $C_1^1 + C_2^1$.

The output Equation (4) can be rewritten in the following form.

$$\begin{aligned} y(t) &= \frac{(C_1^1 + C_2^1)\tilde{A}_1^1(x_1)\tilde{A}_2^1(x_2) + (C_1^2 + C_2^1)\tilde{A}_1^2(x_1)\tilde{A}_2^1(x_2)}{\tilde{A}_1^1(x_1)\tilde{A}_2^1(x_2) + \tilde{A}_1^2(x_1)\tilde{A}_2^1(x_2) + \tilde{A}_1^1(x_1)\tilde{A}_2^2(x_2) + \tilde{A}_1^2(x_1)\tilde{A}_2^2(x_2)} + \\ &\quad \frac{(C_1^1 + C_2^2)\tilde{A}_1^1(x_1)\tilde{A}_2^2(x_2) + (C_1^2 + C_2^2)\tilde{A}_1^2(x_1)\tilde{A}_2^2(x_2)}{\tilde{A}_1^1(x_1)\tilde{A}_2^1(x_2) + \tilde{A}_1^2(x_1)\tilde{A}_2^1(x_2) + \tilde{A}_1^1(x_1)\tilde{A}_2^2(x_2) + \tilde{A}_1^2(x_1)\tilde{A}_2^2(x_2)} \\ &= \frac{(C_1^1\tilde{A}_1^1(x_1) + C_1^2\tilde{A}_1^2(x_1))(\tilde{A}_2^1(x_2) + \tilde{A}_2^2(x_2))}{(\tilde{A}_2^1(x_2) + \tilde{A}_2^2(x_2))(\tilde{A}_1^1(x_1) + \tilde{A}_1^2(x_1))} + \\ &\quad \frac{(C_2^1\tilde{A}_2^1(x_2) + C_2^2\tilde{A}_2^2(x_2))(\tilde{A}_1^1(x_1) + \tilde{A}_1^2(x_1))}{(\tilde{A}_2^1(x_2) + \tilde{A}_2^2(x_2))(\tilde{A}_1^1(x_1) + \tilde{A}_1^2(x_1))} \\ &= \frac{C_1^1\tilde{A}_1^1(x_1) + C_1^2\tilde{A}_1^2(x_1)}{\tilde{A}_1^1(x_1) + \tilde{A}_1^2(x_1)} + \frac{C_2^1\tilde{A}_2^1(x_2) + C_2^2\tilde{A}_2^2(x_2)}{\tilde{A}_2^1(x_2) + \tilde{A}_2^2(x_2)} \end{aligned} \tag{6}$$

It is clear that the first part of the above statements is only a function of the first input, x_1, and the second part is only function of the second input, x_2. Therefore the output of FRBS, $y(t)$, which is a non-linear function of $x_1(t)$ and $x_2(t)$, i.e.

$$y(t) = f(x_1(t), x_2(t)) \tag{7}$$

can be expressed as a linear combination of non-linear functions of individual inputs.

$$y(t) = y_1(t) + y_2(t) = f_1(x_1(t)) + f_2(x_2(t)) \tag{8}$$

where

$$y_1 = \frac{C_1^1 \tilde{A}_1^1(x_1) + C_1^2 \tilde{A}_1^2(x_1)}{\tilde{A}_1^1(x_1) + \tilde{A}_1^2(x_1)} \tag{9}$$

and

$$y_2 = \frac{C_2^1 \tilde{A}_2^1(x_2) + C_2^2 \tilde{A}_2^2(x_2)}{\tilde{A}_2^1(x_2) + \tilde{A}_2^2(x_2)} \tag{10}$$

The FRBS with 4 rules can be rewritten in the non-interactive model if and only if there is a solution (exact or approximate) for the set of Equations (5). The output y_1 given in Equation (9) represents a fuzzy rule-based system with 2 rules and one input, x_1. The rules are as follows:

$\hat{R}^1$: If x_1 is $\tilde{A}_1^1$ then y_1 is C_1^1

$\hat{R}^2$: If x_1 is $\tilde{A}_1^2$ then y_1 is C_1^2

Similarly, the output y_2 given in Equation (10) represents a fuzzy rule-based system with 2 rules and one input, x_2. Rules are given below:

$\hat{R}^3$: If x_2 is $\tilde{A}_2^1$ then y_2 is C_2^1

$\hat{R}^4$: If x_2 is $\tilde{A}_2^2$ then y_2 is C_2^2

To extend the concept of non-interactive model for FRBS introduced in this section for multi-input systems, a NN structure for FRBS is introduced. This NN model simplifies the formulation of FRBS.

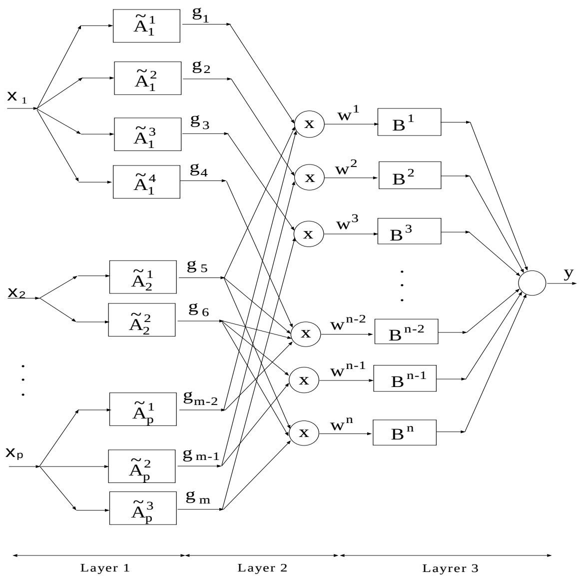

Figure 1: Artificial neural network model for FRBSs.

4 Artificial Neural Network Model of Fuzzy Rule-Based Systems

Figure (1) represents the proposed ANN structure for the FRBS given in statement ($\star$) [1, 3]. We can consider this ANN model for FRBSs in three individual layers. The first layer represents the MFs in the antecedent part of the rules, $\tilde{A}_j^k$, the second layer is a simple multiplication for combining different MFs and the third layer, contains the parameters of the consequent part of rules, B^i.

Each input $x_j, j = 1, 2, ..., p$ is connected to its related individual MFs. The output of this layer can be represented in a vector form $\mathbf{g} = [g_1, g_2, ..., g_r, ..., g_m]^T$ where $g_r, (r = 1, 2, ..., m)$ is the output of each individual MF and m is the total number of individual MFs for all inputs, i.e. $m = \sum_{j=1}^{p} K_j$. In the second layer a *connection matrix*, Φ, is introduced to show the relationship between each individual MF and the rule firing strength for each rule, w^i. The output of this layer can also be represented in a vector form $\mathbf{w} = [w^1, w^2, ..., w^i, ..., w^n]^T$. Connection matrix Φ is a matrix $n \times m$ whose elements are binary. An element '1' represents the existence of a

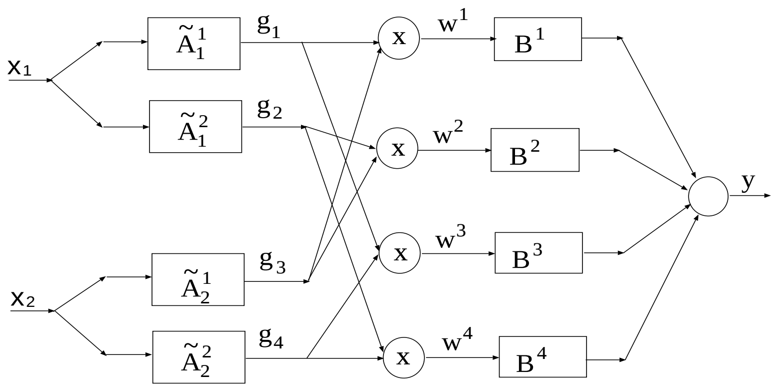

Figure 2: Artificial neural network model for the example fuzzy rule-based system.

connection between the output of first-layer and the i^{th} firing strength. Likewise a '0' represents the lack of a connection. The consequence parameters, B^i can also be expressed in a vector form $\mathbf{B} = [B^1, B^2, ..., B^i, ..., B^n]^T$.

Example: Consider the simple FRBS explained in the previous section. The ANN structure for these rules is depicted in Figure (2). There are four MFs in the first layer. The output of this layer is represented by vector $\mathbf{g} = [\tilde{A}_1^1(x_1), \tilde{A}_1^2(x_1), \tilde{A}_2^1(x_2), \tilde{A}_2^2(x_2)]^T$ and the connection matrix, Φ, is defined as follows:

$$\Phi = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{matrix} \leftarrow & w^1 \\ \leftarrow & w^2 \\ \leftarrow & w^3 \\ \leftarrow & w^4 \end{matrix} \qquad (11)$$

$$\begin{matrix} \uparrow & \uparrow & \uparrow & \uparrow \\ g_1 & g_2 & g_3 & g_4 \end{matrix}$$

The ANN model of the non-interactive model of FRBS given in this example is depicted in Figure (3). The output $y(t)$ is the summation of two independent outputs $y_1(t)$ and $y_2(t)$.

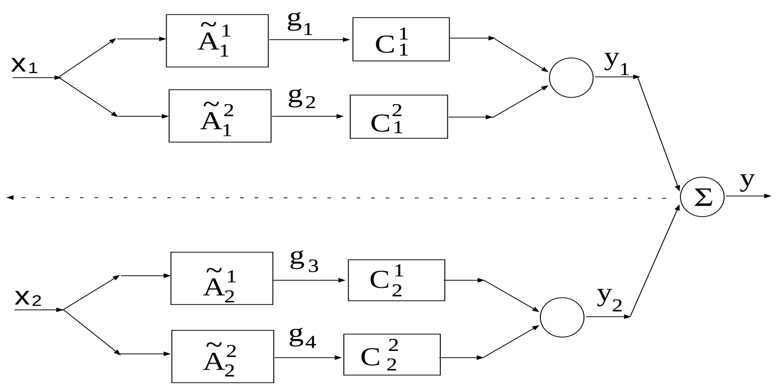

Figure 3: Artificial neural network model for the example non-interactive model of fuzzy rule-based system.

5 Non-Interactive Model Generalization

The procedure explained in Section 3 can easily be extended to a general multi-input FRBS with n rules. We define a new vector $\mathbf{C} = [C^1, C^2, ..., C^r, ..., C^m]^T$ which will be replaced by the consequent vector $\mathbf{B}$. From the set of Equations (5) it is evident to see that the relation between vector $\mathbf{B}$ and $\mathbf{C}$ is defined as follows:

$$\Phi_{n\times m}\mathbf{C}_{m\times 1} = \mathbf{B}_{n\times 1} \tag{12}$$

Therefore the new consequent vectors for non-interactive model, $\mathbf{C}$ is expressed below:

$$\mathbf{C} = \Phi^{\dagger}\mathbf{B} \tag{13}$$

where $\Phi^{\dagger}$ is the *pseudo inverse* of the connection matrix, Φ. We use the pseudo inverse of connection matrix Φ since it is not a square matrix. Even when Φ is a square matrix i.e. when the number of rules is equal to the total number of individual MFs, $n = m$, it may **not** be full rank i.e. $\rho(\Phi) \leq n$. The rank of matrix Φ is determined by the structure of the FRBS.

Upon calculation of the new consequent vector, $\mathbf{C}$, the non-interactive model for the FRBS with the following configuration will be used.

$\hat{R}^r$: If x_j is $\tilde{A}_j^k$ then y is C^r $\star\star$

The output of the FRBS is represented as a linear combination of independent non-linear function of inputs, i.e.

$$\begin{aligned} y &= f(x_1, x_2, ..., x_p) \\ &= f_1(x_1) + f_2(x_2) + ... + f_p(x_p) \\ &= y_1 + y_2 + ... + y_p \end{aligned} \tag{14}$$

It is to be noted that the number of consequent parameters has been reduced from n to m. For instance a FRBS with $p = 4$ inputs and $K_1 = K_2 = K_3 = K_4 = 3$ individual MFs for each input and $n = 25$ rules can be replaced with just $m = 12$ consequent parameters.

If we use the Takagi-Sugeno fuzzy if-then rule [6] instead of the fuzzy if-then rule explained in Section 2, the linear combination of inputs can be expressed in a bi-linear form. This is not within the scope of this paper and it will be explained in a separate paper.

6 Illustrative Example: Truck Backer-Upper Control

The truck backer-upper control system is taken as a test-bed to illustrate our proposed scheme. Backing a truck to a loading dock is a non-linear control problem which can involve extensive computation time to steer the truck to a prescribed loading zone. As linguistic fuzzy controller has been shown to be more advantageous over the traditional non-linear controller, it seems that the fuzzy if-then rule control can also work well in this specific problem. If we elicit the skilled driver experience in a fuzzy if-then rule format, we can be assured that the fuzzy controller is working with the same set of rules, and would obtain the same trajectory. For truck backer-upper control, Kong and Kosko [2] propose a fuzzy logic controller with 35 expert rules, and they compare their results obtained from a fuzzy controller with results achieve by using a neural network controller. Fuzzy controller has been shown to give more appropriate tracking results.

The truck in our simulation is the same truck used in [2] and [7] except for the size of the yard and the definition of steering and azimuth angle. Since our study is performed by simulation, the dynamics of the truck is required. We used the following approximate kinematics [7].

$$x(t+1) = x(t) + v\,(cos[\phi(t)+\theta(t)] + sin[\theta(t)]sin[\phi(t)]) \tag{15}$$

$$y(t+1) = y(t) + v\,(sin[\phi(t)+\theta(t)] - sin[\theta(t)]cos[\phi(t)]) \tag{16}$$

$$\phi(t+1) = \phi(t) - v\left(sin^{-1}\left[\frac{2sin[\theta(t)]}{\ell}\right]\right) \tag{17}$$

where x, y, and ϕ are rear, center of truck coordinate and azimuth angle of truck in yard respectively. They can be considered as state variables of the system which indicate position and direction of the truck in the yard at any instant of time. θ is the steering angle to direct the truck to the loading zone x_f and y_f. Constant parameters v and ℓ are truck speed and length of the truck respectively. The control goal is to steer the truck from any initial position to pre-specified loading dock with a right azimuth angle ($\phi_f = 90$) and coincided rear position. The steering angle θ is the control action which is provided by the designed fuzzy controller. Since we pre-suppose adequate clearance between the truck and the loading dock, state variable y can be abandoned for the reason that it becomes a dependent variable. Therefore the inputs to the controller are x and ϕ. The range of variables for simulated truck and controller are as follows;

$$x \in [x^- \quad x^+] = [0\ 100] \qquad \phi \in [\phi^- \quad \phi^+] = [-90\ 270] \qquad \theta \in [\theta^- \quad \theta^+] = [-30\ 30]$$

The truck speed $v = 5$ and the length of the truck $\ell = 4$. The maximum width of the yard is

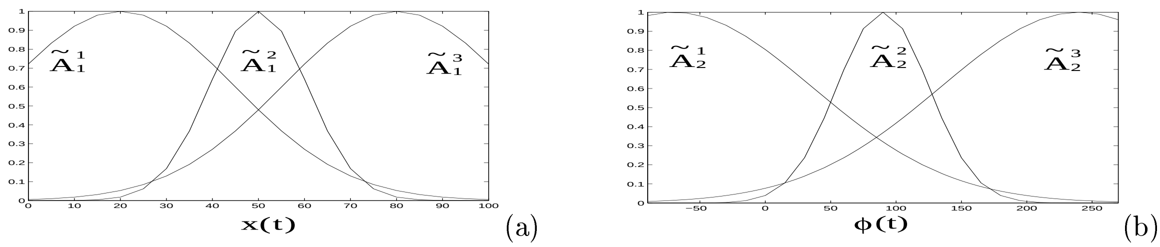

Figure 4: Membership functions of the fuzzy values a) for the first input i.e. $\tilde{A}_1^1, \tilde{A}_1^2$ and $\tilde{A}_1^3$ b) for the second input i.e. $\tilde{A}_2^1, \tilde{A}_2^2$ and $\tilde{A}_2^3$.

$y = 100$. Desired loading dock position is $x_f = 50$ and $y_f = 100$. Positive attitude of azimuth angle ϕ is clockwise with respect to the horizontal line. Steering angle θ is positive when the steering wheels rotate counterclockwise.

In our simulation, we consider a fuzzy controller with 9 rules with the following structure. It has been shown in [4] that these rules are able to control the truck towards the loading dock.

R^1: If x is $\tilde{A}_1^1$ and ϕ is $\tilde{A}_2^1$ then θ is B^1

R^2: If x is $\tilde{A}_1^2$ and ϕ is $\tilde{A}_2^1$ then θ is B^2

R^3: If x is $\tilde{A}_1^3$ and ϕ is $\tilde{A}_2^1$ then θ is B^3

R^4: If x is $\tilde{A}_1^1$ and ϕ is $\tilde{A}_2^2$ then θ is B^4

R^5: If x is $\tilde{A}_1^2$ and ϕ is $\tilde{A}_2^2$ then θ is B^5

R^6: If x is $\tilde{A}_1^3$ and ϕ is $\tilde{A}_2^2$ then θ is B^6

R^7: If x is $\tilde{A}_1^1$ and ϕ is $\tilde{A}_2^3$ then θ is B^7

R^8: If x is $\tilde{A}_1^2$ and ϕ is $\tilde{A}_2^3$ then θ is B^8

R^9: If x is $\tilde{A}_1^3$ and ϕ is $\tilde{A}_2^3$ then θ is B^9

The consequent parameters $\mathbf{B}$ is given in expression (18). The MFs of the fuzzy values for the first input, x, i.e. $\tilde{A}_1^1, \tilde{A}_1^2$ and $\tilde{A}_1^3$ and the MFs of the fuzzy values for the second input, ϕ, i.e. $\tilde{A}_2^1, \tilde{A}_2^2$ and

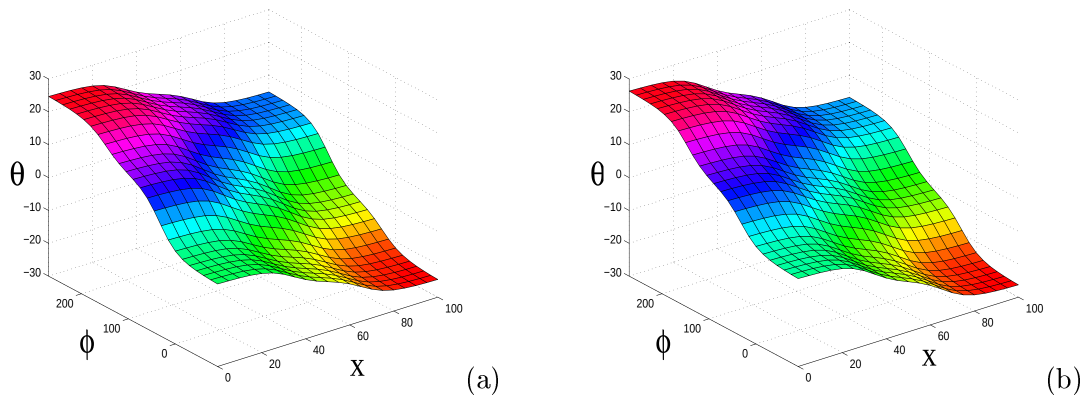

Figure 5: Control surface of fuzzy controller a) with 9 rules b) non-interactive model.

$\tilde{A}_2^3$ are depicted in Figures (4-a) and (4-b) respectively. The control surface (all possible control action) for the fuzzy control using 9 rules is depicted in Figure (5-a) in the range of $x \in [0\ 100]$ and $\phi \in [-90\ 270]$ for the first and second inputs respectively.

From the above rule-base the connection matrix can be easily formed. It is given below. The new consequent vector, **C**, for the non-interactive model is calculated using Equation (13).

$$\mathbf{B} = \begin{bmatrix} -5 \\ -15 \\ -25 \\ 15 \\ 0 \\ -15 \\ 25 \\ 15 \\ 5 \end{bmatrix} \qquad \Phi = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} 11.6667 \\ 0.0000 \\ -11.6667 \\ -15.0000 \\ 0.0000 \\ 15.0000 \end{bmatrix} \tag{18}$$

Using the new consequent vector, **C**, we can use a non-interactive model for the rule-base given in this section. The control surface generated from the non-interactive model is depicted in Figure

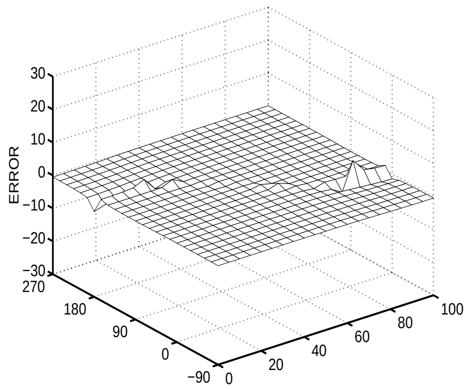

Figure 6: Error between interactive and non-interactive control surfaces.

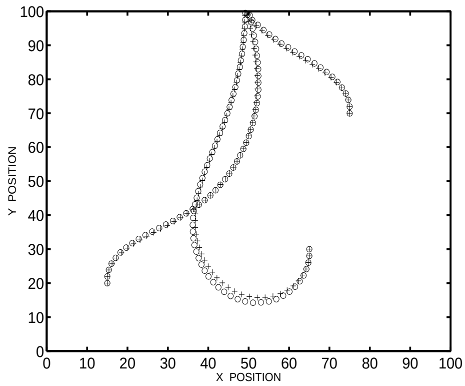

Figure 7: Truck trajectory tracking with interactive ('+') and non-interactive ('o') fuzzy controllers.

(5-b). The difference between two control surfaces is shown in Figure (6).

To compare the performance of both interactive and non-interactive fuzzy controllers, truck trajectory tracking from three initial positions, $[x, y, \phi] = [70,\ 30,\ 270]$, $[20,\ 20,\ 90]$ and $[80,\ 70,\ 90]$, are performed and they are depicted in Figure (7). The trajectories achieved by non-interactive model are shown by 'o' symbol and '+' symbol represents the trajectories obtained from the fuzzy controller with 9 rules.

Considering the error between two control surfaces shown in Figure (6) and the trajectory tracking shown in Figure (7) we can conclude that performance of the non-interactive model of FRBS is almost identical to the FRBS with 9 rules. The new model gives some advantages. There

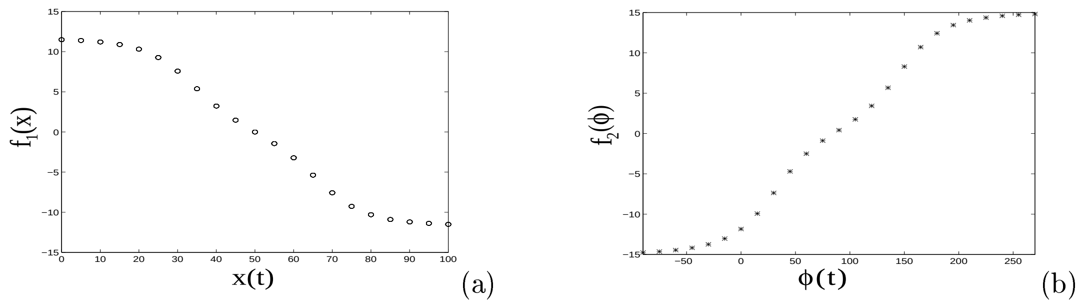

Figure 8: Independent function generated by non-interactive model a) $f_1(x)$ b) $f_2(\phi)$.

are 6 rules in the new model i.e. one third of rules in the original FRBS have been removed. Furthermore the output of new model can be expressed as a linear combination of two non-linear function of inputs, i.e.

$$\theta = f(x, \phi) = f_1(x) + f_2(\phi) \tag{19}$$

The non-linear functions $f_1(x)$ and $f_2(\phi)$ are depicted in Figures (8-a) and (8-b) respectively.

7 Conclusions

In this paper a non-interactive model for FRBS is introduced; a method whereby the output can be represented as a linear combination of non-linear function of individual inputs. If there is an inverse (exact or approximate) for the connection matrix, Φ, introduced in Section 4, then the rules can be written in a non-interactve format. The possiblity of determining a solution for the inverse of connection matrix depends on the configuration of original fuzzy rule-based system.

References

[1] J-S. R. Jang, " Neuro-fuzzy modeling and control", *Proceedings of IEEE*, vol. 38. no. 3, pp. 378-405, March 1995.

[2] Kong, Seong-Gon,& Kosko, Bart, "Comparison of fuzzy and neural track backer-upper control system". Proc. of Int. Joint Conf. on Neural Networks (IJCNN-90), vol.3, pp. 349-358, June 1990.

[3] C. T. Lin, and C. S. G. Lee, "Neural-network-based fuzzy logic control and decision system", *IEEE Trans. on Computers*, vol. 40, no. 12, pp. 1320-1336, Dec. 1991.

[4] A. Lotfi, and A. C. Tsoi, "Learning fuzzy inference systems using an adaptive membership function scheme", *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 26, no. 4, pp. 326-332, April 1996.

[5] A. Lotfi, H. C. Andersen and A. C. Tsoi, " Matrix formulation of fuzzy rule-based systems", *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 26, no. 4, pp. 332-340, April 1996.

[6] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control", *IEEE Trans. on Systems, Man, and Cybernetics*, vol. SMC-15, no. 1, pp. 116–132, Jan. 1985.

[7] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol.22, No.6, Nov./Dec. 1992.

[8] L. X. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning", *IEEE Trans. on Neural Networks*, vol. 3, no. 5, pp. 807–814, Sep. 1992.