Performance Evaluation of a Systematic Fuzzy Logic Control System

David Fugate A.T. Alouani
Department of ECE Advanced Systems Lab
Tennessee Technological University
Cookeville, TN. 38505

ABSTRACT

Fuzzy logic control (FLC) has been successfully used in many applications. The design of a FLC has mostly relied on an expert or used ad hoc techniques. For this reason the robustness of FLC has only been shown via testing or simulation. Recently, work developed at Tennessee Technological University (TTU) proved that it is possible to design a FLC that guarantees the system robustness to parameter uncertainties. In addition, this design is systematic and does not require an expert. It rather uses available information contained in the system model and interprets the uncertainties in the system as a source of fuzziness. The objective of this paper is to implement this new design methodology to provide real time control and use it to control one joint of an IBM 7540 robot arm. Experimental results of the performance of this controller will be presented in this paper.

1. Introduction.

The control of nonlinear uncertain dynamical system has been an active research area for many years. The different approaches to this problem can be summarized into two classes: The linear and nonlinear design methodologies. The linear philosophy consists of basically linearizing the system and providing a norm bound of the model uncertainty to design robust linear controllers. Well known linear robust designs are the linear quadratic regulators with loop transfer recovery (LQG/LTR) and the H_∞ control [1], among others. The assumption on norm bound may lead to unnecessary conservatism of the controller. In addition, the controller robustness remains valid only around an operating point where the linearization took place. For the nonlinear philosophy, the two main design approaches are the variable structure control (VSC) and Lyapunov based control [2]. One of the limitations of the current nonlinear design is that the stability of the controlled system can not be guaranteed unless the structure of the uncertainties satisfy what is called matching conditions [3].

A new control strategy, Fuzzy Logic Control, that uses multilevel logic has been recently introduced to deal with the control of uncertain and complex systems [4]. Fuzzy logic control has been successfully applied to many practical systems [5]. Previously design methodologies consisted of model-free designs that required either an expert or an add hoc technique. In either case the robustness of the model-free based fuzzy logic control could only be shown via testing and/or simulations. These designs methods do not have robustness considerations in the design.

Recent work at The Advanced Systems Lab (ASL) at TTU has shown that it is possible to design fuzzy logic control systems that guarantee robustness to system uncertainties [6]. The idea is to first interpret a class of uncertain dynamical systems as a single fuzzy dynamical system and provide a new model, called a fuzzy model, that captures all the features of the uncertain system for the ranges of the uncertainties of the system parameters. This new model is called a fuzzy model and is represented by a fuzzy differential equation. Then an extension of classical variable structure control design is made using the fuzzy model to include fuzzy dynamical systems.

The early development of this design could not be implemented in real time. The objective of this paper is two fold. First, provide a real time implementation of the new controller. Second, test the performance of this controller on a physical system; the control of joint 2 of an IBM 7540, Figure 1. This robot has two joints and four degrees of freedom.

2. System Model.

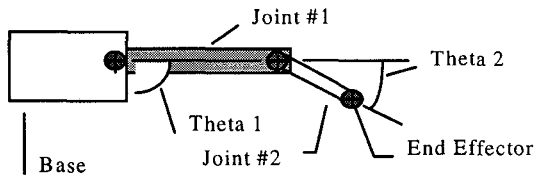

Figure 1: Diagram of IBM 7540 Robot

The system consists of the Joint #2 of an IBM 7540 robot arm, with a dynamic equation:

0-8186-7873-9/97 $10.00 © 1997 IEEE

$$ml^2\ddot{\theta}(t) + mgl\sin(\theta(t)) + Fr = u(t) \quad (1).$$

Where $\theta(t)$, $u(t)$, m, l, g, and Fr are the angular position (Theta 2) to be controlled, control torque, mass of Joint #2 and payload, length of Joint #2, gravity, and a friction term. The model described in Equation (1) is called a crisp (non-fuzzy) model [6].

Eq. (1) can be written in state equation form as

$$\dot{\underline{\chi}}(t) = f(\underline{\chi}(t), \eta(t)) + g(\underline{\chi}(t), \eta(t))u(t) \quad (2),$$

where $\underline{\chi}(t) = \left[\theta(t) \quad \dot{\theta}(t)\right]'$ is the state vector, and

$\eta = [m \quad l \quad F_r]'$,

$\underline{f}(\cdot) = [f_1 = \chi_2(t), f_2 = -g_r l \sin(\chi_1(t)) - (1/ml)^2 Fr]'$,

and $\underline{g}(\cdot) = [g_1 = 0, g_2 = -(1/ml)^2]'$ (3).

The parameter η is made up of the uncertain parameters of the system. In this study it is assumed that the ranges of the uncertainties are: $m \in [20, 30]$ kg, $l \in [0.35, 0.45]$m, and $Fr \in [-0.45, 0.45]$.

Because of the uncertainties in the system parameters, this system is nonlinear and uncertain. Given a fixed initial condition, for each value of the parameters, the above differential equation will have a solution. Characterizing the above system in a crisp way is not practical because of the large range of possible values of the uncertain parameters. These uncertainties in the system parameters are used as a source of fuzziness. The solution of the above differential equation, for all possible system parameters, will be seen as a fuzzy state vector $\underline{\tilde{\chi}}(t)$ [6] that is a solution of the fuzzy differential equation

$$\dot{\underline{\tilde{\chi}}}(t) = \tilde{f}(\underline{\tilde{\chi}}(t)) \tilde{+} \tilde{g}(\underline{\tilde{\chi}}(t))\tilde{u}(t) \quad (4).$$

Note that the $\tilde{f}$, $\tilde{+}$, $\tilde{=}$, $\tilde{\cdot}$, and $\tilde{/}$ denote a fuzzy set and fuzzy arithmetic operations.

To construct the fuzzy model, the universe of discourses, $[-\pi/2, \pi/2]$ for $x_1(t)$, and $[-\pi/4, \pi/4]$ for $x_2(t)$ are partitioned using 0.157rad and 0.1963rad/s, as the spacing of the partitions in each fuzzy set. For each given fuzzy state vector, the fuzzy maps $\tilde{f}(\underline{\tilde{\chi}}(t))$ *and* $\tilde{g}(\underline{\tilde{\chi}}(t))$ are computed for all possible values of the vector η. See [6] for details.

The result is a set of if then rules of the form

IF $x_1(t)$ is $F^{ki}{}_{x1}$ ***AND*** $x_2(t)$ is $F^{kj}{}_{x2}$, ***THEN*** $\tilde{f}(\underline{\tilde{\chi}}(t))$ is $F^{kj}{}_{f}$,

And ***IF*** $x_1(t)$ is $F^{ki}{}_{x1}$ ***AND*** $x_2(t)$ is $F^{kj}{}_{x2}$, ***THEN*** $\tilde{g}(\underline{\tilde{\chi}}(t))$ is $F^{ki}{}_{g}$.

The complete rule bases for $\tilde{f}(\cdot)$ is given in table 1.1. The rule base for $\tilde{g}(\cdot)$ is similar. Note that $\tilde{f}(\cdot)$ and $\tilde{g}(\cdot)$ contain the model knowledge of the system.

3. The FLC Design.

The control scheme consists of extending the design techniques of Variable Structure Control [2] to fuzzy dynamical systems [6].

First, define a fuzzy error vector using a desired trajectory

$\underline{\tilde{e}} = \underline{\tilde{x}}(t) - \underline{\tilde{x}}_d(t)$ and

$\underline{\tilde{x}}_d(t) = \underline{x}_d(t)$.

Second, a fuzzy switching function is created to describe the system error

$\tilde{s}(\underline{\tilde{e}}) = \lambda_1 \tilde{\cdot} \tilde{e}_1 + \lambda_2 \tilde{\cdot} \tilde{e}_2$.

This forms a rule base $F^{ki}{}_{s}$ that functions as:

IF e_1 is $F^{k1}{}_{e1}$ ***AND*** e_2 is $F^{k2}{}_{e2}$, ***THEN***

$s(\underline{e})$ is $F^{(k1,k2)}{}_{s}$.

The rate of change of $\tilde{s}(\underline{\tilde{e}})$ is controlled, [6], by

$\dot{\tilde{s}}(\underline{\tilde{e}}) = -\rho \cdot \tilde{s}(\underline{\tilde{e}})$.

Then, a fuzzy control torque is computed,

$$\tilde{u} = [-\tilde{f}(\tilde{x}) \tilde{+} \dot{\tilde{x}}_{d2} \tilde{+} \dot{\tilde{s}}(\underline{\tilde{e}}) \tilde{=} \lambda_2 \tilde{\cdot} \tilde{e}_2] \tilde{/} \tilde{g}(\tilde{x}) \quad (5),$$

that guarantees that the error signal converges asymptotically to zero for all possible system parameters in their practical range [6]. This fuzzy torque $\tilde{u}$ is a rule base similar in construction to the $\tilde{f}(\cdot)$ and $\tilde{g}(\cdot)$ and is shown in table 1.2.

4. Controller Implementation.

The FLC was developed and simulated using Matlab, Simulink, and C code. First a Matlab program was used to construct the fuzzy model. This Matlab program first constructed a fuzzy state vector from the state. Then the program varied the uncertain parameters and constructed the fuzzy maps for the $\tilde{f}(\cdot)$ and $\tilde{g}(\cdot)$ of (4). The fuzzy state vector and the fuzzy maps $\tilde{f}(\cdot)$ and $\tilde{g}(\cdot)$ were stored as data files.

The first attempts to implement the controller in real-time were not successful due to the computation requirements of fuzzy mathematics. The early attempts were directed at computing $\tilde{u}$ using (5) in real-time with the given fuzzy input information and the data files mentioned previously. This approach was not adequate for a real-time application.

Later, an alternative approach was developed. The fuzzy model and fuzzy state vector data files were used to create the control rule base off-line. See Figure 2.

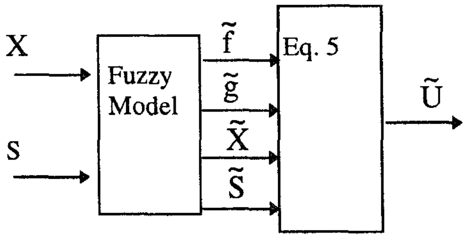

Figure 2: Creation of the Control Rule Base.

This rule base for the control torque would then be used on line, see Figure 3, to perform the fuzzy inference [6] as:

IF x_1 is $F^{k1}{}_{x1}$ *AND* x_2 is $F^{k2}{}_{x2}$ *AND* s is $F^{(k1,k2)}{}_{s}$,

THEN u is $F^{(k1,k2)}u$.

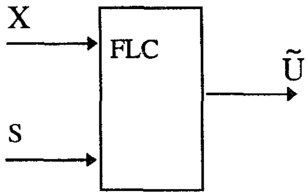

Figure 3: The Control Rule Base.

A C program was written to perform the task of the fuzzy logic controller. The data for the fuzzy state vector, the fuzzy map for the torque (u), and fuzzy rule base for the switching function (s) were handled as arrays in the program memory. The input state and s are fuzzified by a For-Loop that compares the fuzzy partitions to the actual input state and s. These fuzzy partitions are given a firing level, between zero and one, according to their relation to the actual values using the L-R parameterization technique [4]. The fuzzy inference consists of another For-Loop that sums the product of the mean value in each fuzzy partition of the fuzzy map of the torque with the weight of each rule that was found in the fuzzification of the input state and s. This sum is then defuzzified to compute the output torque. The Seugeno-Style fuzzy inference [7]. was used in this application.

The C program was needed to perform the fuzzy inference with enough speed for a real-time application. This program was incorporated into the Matlab and Simulink [8] environments by creating a Mex file [8]. After creation of this source code a Mex file is created using the CMEX command in Matlab [8].

The FLC was simulated using Matlab and Simulink [7]. The simulations were performed with different L-R partition spacing for the states (to control the resolution of the fuzzy map of the state), different weights for the error, and different control gains. The values of $\lambda_1 = 5$, $\lambda_2 = 5$, and $\rho = 35$ were found to give the best overall transient and steady state performance for this given system. The previous values of the fuzzy L-R partition spacing for θ and $\dot{\theta}$ of 0.157rad and 0.1963rad/s performed adequately. A transient response was obtained with no overshoot.

5. Hardware Implementation and Testing.

The system model used for simulation in Simulink was replaced with the robot itself. See Figure 4. Via interface software, that was developed at ASL, which converts the control signal to a analog output value by the digital-to-analog converter and acquires the digital input encoder position information. The output signal controls the servoamplifiers for the robot.

The interface software consists of S-functions [8] that are used to implement Mex files that performed the necessary interfacing. These Mex files were developed from C code produced at ASL. The S-function is a block used in the Simulink environment to execute a Mex file. See Figure 5 for the Simulink block diagram that was used to implement the controller.

The velocity state information was computed using the derivative block in Simulink. A low-pass filter was used to smooth the derivative of the encoder digital pulse signal.

The reference trajectory was a linear $\pm\pi / 2$ rad. sweep in five seconds from an initial condition of 0.4962rad.

The repeatability of the system is shown with the system error signal in Figure 6. Also the Robustness was tested by comparing the error signal of a 'no load' system to a system with a 15kg weight placed on the end effector. See Figures 7 and 8 for the system error and the actual trajectory of the Robustness Test.

6. Results and Conclusions.

The test results are shown in Figure 6, 7, and 8. Figures 6 and 7 reveal a maximum error of 0.06 rad translates to a maximum error of 0.001m in x-y coordinates for this system. Figure 7 and 8 illustrates that the system performance was not degraded from the load variation. The FLC worked performed very well for this single joint control problem. Future work will extend this design to all of the degrees of freedom of the IBM 7540 at ASL.

8. REFERENCES.

[1] Maciejowski, Multivariable Feedback Design, Addison-Weslet, 1989.

[2] Zinober, A.S., Deterministic control of Uncertain Systems, Peregrinus 1990.

[3] Corless, M. And G. Leitmann, 1981, " Continuous State Feedback Guaranteeing Ultimate Boundedness for Uncertain Dynamic Systems", IEEE Trans. On Automatic Control, Vol. 26, 1981.

[4] Yager, Ronald and Filev, Dimitar, *Essentials of Fuzzy Modeling and Control*, Wiley Interscience, New York, NY, 1994.

[5] Sugeno, M., Industrial Applications of Fuzzy Logic Control, Elsevier Science Publishing, 1985.

[6] Ben Ghalia, 1995 Fuzzy Modeling and Robust Control of Nonlinear Uncertain Dynamical Systems," PhD Dissertation, Department of ECE, Tennessee Technological University

[7] *Matlab Fuzzy Logic Toolbox*, Mathworks, 1995.

[8] *Simulink Users Guide*, Mathworks, 1994.

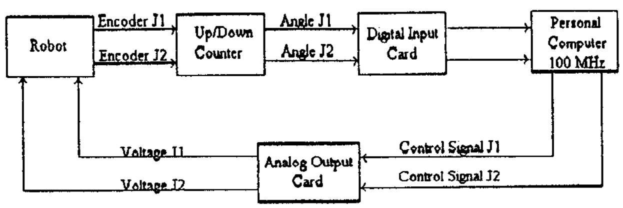

Figure 4: Hardware Configuration Block Diagram

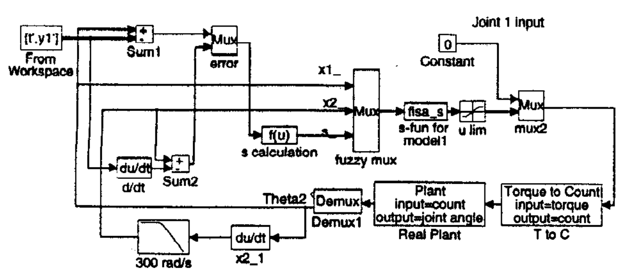

Figure 5: Simulink Block Diagram used to test the controller on Joint #2 of the IBM 7540.

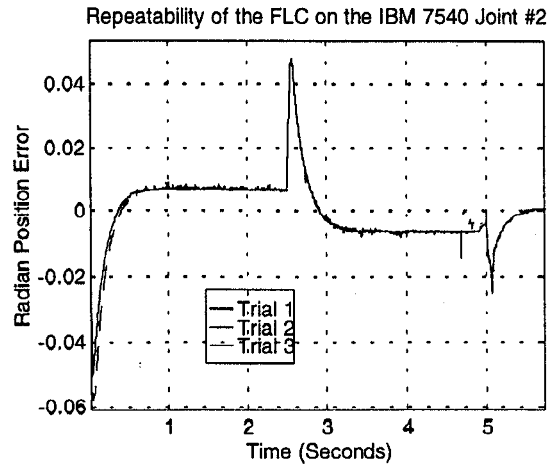

Figure 6: Repeatability of the FLC

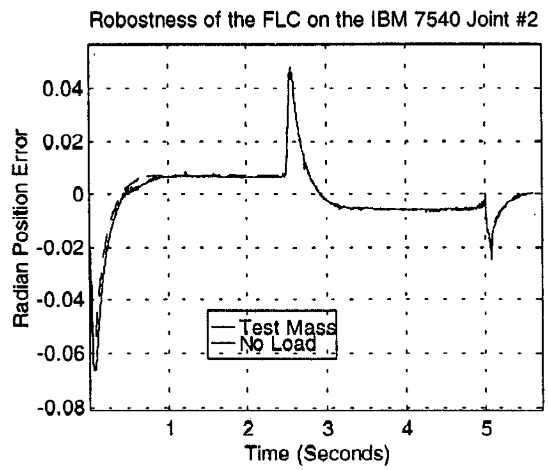

Figure 7: Robustness of the FLC

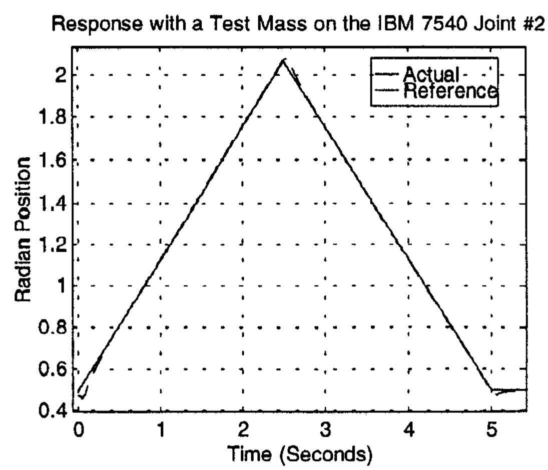

Figure 8: Robustness of the FLC

Table 1.1 - Fuzzy Rule Base of $\tilde{f}(\cdot)=[\tilde{f}_1(\cdot),\tilde{f}_2(\cdot)]^T$.

F^{k2}_{x2}

F^{k1}_{x1}	k2=1	k2=2	k2=3	k2=4	k2=5	k2=6
k1=1	$F^{(1,1)}_{f1}, F^{(1,1)}_{f2}$	$F^{(1,2)}_{f1}, F^{(1,2)}_{f2}$	$F^{(1,3)}_{f1}, F^{(1,3)}_{f2}$	$F^{(1,4)}_{f1}, F^{(1,4)}_{f2}$	$F^{(1,5)}_{f1}, F^{(1,5)}_{f2}$	$F^{(1,6)}_{f1}, F^{(1,6)}_{f2}$
k1=2	$F^{(2,1)}_{f1}, F^{(2,1)}_{f2}$	$F^{(2,2)}_{f1}, F^{(2,2)}_{f2}$	$F^{(2,3)}_{f1}, F^{(2,3)}_{f2}$	$F^{(2,4)}_{f1}, F^{(2,4)}_{f2}$	$F^{(2,5)}_{f1}, F^{(2,5)}_{f2}$	$F^{(2,6)}_{f1}, F^{(2,6)}_{f2}$
k1=3	$F^{(3,1)}_{f1}, F^{(3,1)}_{f2}$	$F^{(3,2)}_{f1}, F^{(3,2)}_{f2}$	$F^{(3,3)}_{f1}, F^{(3,3)}_{f2}$	$F^{(3,4)}_{f1}, F^{(3,4)}_{f2}$	$F^{(3,5)}_{f1}, F^{(3,5)}_{f2}$	$F^{(3,6)}_{f1}, F^{(3,6)}_{f2}$
k1=4	$F^{(4,1)}_{f1}, F^{(4,1)}_{f2}$	$F^{(4,2)}_{f1}, F^{(4,2)}_{f2}$	$F^{(4,3)}_{f1}, F^{(4,3)}_{f2}$	$F^{(4,4)}_{f1}, F^{(4,4)}_{f2}$	$F^{(4,5)}_{f1}, F^{(4,5)}_{f2}$	$F^{(4,6)}_{f1}, F^{(4,6)}_{f2}$
k1=5	$F^{(5,1)}_{f1}, F^{(5,1)}_{f2}$	$F^{(5,2)}_{f1}, F^{(5,2)}_{f2}$	$F^{(5,3)}_{f1}, F^{(5,3)}_{f2}$	$F^{(5,4)}_{f1}, F^{(5,4)}_{f2}$	$F^{(5,5)}_{f1}, F^{(5,5)}_{f2}$	$F^{(5,6)}_{f1}, F^{(5,6)}_{f2}$
k1=6	$F^{(6,1)}_{f1}, F^{(6,1)}_{f2}$	$F^{(6,2)}_{f1}, F^{(6,2)}_{f2}$	$F^{(6,3)}_{f1}, F^{(6,3)}_{f2}$	$F^{(6,4)}_{f1}, F^{(6,4)}_{f2}$	$F^{(6,5)}_{f1}, F^{(6,5)}_{f2}$	$F^{(6,6)}_{f1}, F^{(6,6)}_{f2}$
k1=7	$F^{(7,1)}_{f1}, F^{(7,1)}_{f2}$	$F^{(7,2)}_{f1}, F^{(7,2)}_{f2}$	$F^{(7,3)}_{f1}, F^{(7,3)}_{f2}$	$F^{(7,4)}_{f1}, F^{(7,4)}_{f2}$	$F^{(7,5)}_{f1}, F^{(7,5)}_{f2}$	$F^{(7,6)}_{f1}, F^{(7,6)}_{f2}$
k1=8	$F^{(8,1)}_{f1}, F^{(8,1)}_{f2}$	$F^{(8,2)}_{f1}F^{(8,2)}_{f2}$	$F^{(8,3)}_{f1}F^{(8,3)}_{f2}$	$F^{(8,4)}_{f1}F^{(8,4)}_{f2}$	$F^{(8,5)}_{f1}F^{(8,5)}_{f2}$	$F^{(8,6)}_{f1}, F^{(8,6)}_{f2}$
k1=9	$F^{(9,1)}_{f1}, F^{(9,1)}_{f2}$	$F^{(9,2)}_{f1}, F^{(9,2)}_{f2}$	$F^{(9,3)}_{f1}, F^{(9,3)}_{f2}$	$F^{(9,4)}_{f1}, F^{(9,4)}_{f2}$	$F^{(9,5)}_{f1}, F^{(9,5)}_{f2}$	$F^{(9,6)}_{f1}, F^{(9,6)}_{f2}$
k1=10	$F^{(10,1)}_{f1}, F^{(10,1)}_{f2}$	$F^{(10,2)}_{f1}, F^{(10,2)}_{f2}$	$F^{(10,3)}_{f1}, F^{(10,3)}_{f2}$	$F^{(10,4)}_{f1}, F^{(10,4)}_{f2}$	$F^{(10,5)}_{f1}, F^{(10,5)}_{f2}$	$F^{(10,6)}_{f1}, F^{(10,6)}_{f2}$
k1=11	$F^{(11,1)}_{f1}, F^{(11,1)}_{f2}$	$F^{(11,2)}_{f1}, F^{(11,2)}_{f2}$	$F^{(11,3)}_{f1}, F^{(11,3)}_{f2}$	$F^{(11,4)}_{f1}, F^{(11,4)}_{f2}$	$F^{(11,5)}_{f1}, F^{(11,5)}_{f2}$	$F^{(11,6)}_{f1}, F^{(11,6)}_{f2}$
k1=12	$F^{(12,1)}_{f1}, F^{(12,1)}_{f2}$	$F^{(12,2)}_{f1}, F^{(12,2)}_{f2}$	$F^{(12,3)}_{f1}, F^{(12,3)}_{f2}$	$F^{(12,4)}_{f1}, F^{(12,4}_{f2}$	$F^{(12,5)}_{f1}, F^{(12,5)}_{f2}$	$F^{(12,6)}_{f1}, F^{(12,6)}_{f2}$
k1=13	$F^{(13,1)}_{f1}, F^{(13,1)}_{f2}$	$F^{(13,2)}_{f1}, F^{(13,2)}_{f2}$	$F^{(13,3)}_{f1}, F^{(13,3)}_{f2}$	$F^{(13,4)}_{f1}, F^{(13,4)}_{f2}$	$F^{(13,5)}_{f1}, F^{(13,5)}_{f2}$	$F^{(13,6)}_{f1}, F^{(13,6)}_{f2}$
k1=14	$F^{(14,1)}_{f1}, F^{(14,1)}_{f2}$	$F^{(14,2)}_{f1}, F^{(14,2)}_{f2}$	$F^{(14,3)}_{f1}, F^{(14,3)}_{f2}$	$F^{(14,4)}_{f1}, F^{(14,4)}_{f2}$	$F^{(14,5)}_{f1}, F^{(14,5)}_{f2}$	$F^{(14,6)}_{f1}, F^{(14,6)}_{f2}$
k1=15	$F^{(15,1)}_{f1}, F^{(15,1)}_{f2}$	$F^{(15,2)}_{f1}, F^{(15,2)}_{f2}$	$F^{(15,3)}_{f1}, F^{(15,3)}_{f2}$	$F^{(15,4)}_{f1}, F^{(15,4)}_{f2}$	$F^{(15,5)}_{f1}, F^{(15,5)}_{f2}$	$F^{(15,6)}_{f1}, F^{(15,6)}_{f2}$
k1=16	$F^{(16,1)}_{f1}, F^{(16,1)}_{f2}$	$F^{(16,2)}_{f1}, F^{(16,2)}_{f2}$	$F^{(16,3)}_{f1}, F^{(16,3)}_{f2}$	$F^{(16,4)}_{f1}, F^{(16,4)}_{f2}$	$F^{(16,5)}_{f1}, F^{(16,5)}_{f2}$	$F^{(16,6)}_{f1}, F^{(16,6)}_{f2}$
k1=17	$F^{(17,1)}_{f1}, F^{(17,1)}_{f2}$	$F^{(17,2)}_{f1}, F^{(17,2)}_{f2}$	$F^{(17,3)}_{f1}, F^{(17,3)}_{f2}$	$F^{(17,4)}_{f1}, F^{(17,4)}_{f2}$	$F^{(17,5)}_{f1}, F^{(17,5)}_{f2}$	$F^{(17,6)}_{f1}, F^{(17,6)}_{f2}$
k1=18	$F^{(18,1)}_{f1}, F^{(18,1)}_{f2}$	$F^{(18,2)}_{f1}, F^{(18,2)}_{f2}$	$F^{(18,3)}_{f1}, F^{(18,3)}_{f2}$	$F^{(18,4)}_{f1}, F^{(18,4)}_{f2}$	$F^{(18,5)}_{f1}, F^{(18,5)}_{f2}$	$F^{(18,6)}_{f1}, F^{(18,6)}_{f2}$

Table 1.2 - Fuzzy Rule Base of $\tilde{u}$.

F^{k2}_{x2}

F^{k1}_{x1}	k2=1	k2=2	k2=3	k2=4	k2=5	k2=6
k1=1	$F^{(1,1)}_{u}$	$F^{(1,2)}_{u}$	$F^{(1,3)}_{u}$	$F^{(1,4)}_{u}$	$F^{(1,5)}_{u}$	$F^{(1,6)}_{u}$
k1=2	$F^{(2,1)}_{u}$	$F^{(2,2)}_{u}$	$F^{(2,3)}_{u}$	$F^{(2,4)}_{u}$	$F^{(2,5)}_{u}$	$F^{(2,6)}_{u}$
k1=3	$F^{(3,1)}_{u}$	$F^{(3,2)}_{u}$	$F^{(3,3)}_{u}$	$F^{(3,4)}_{u}$	$F^{(3,5)}_{u}$	$F^{(3,6)}_{u}$
k1=4	$F^{(4,1)}_{u}$	$F^{(4,2)}_{u}$	$F^{(4,3)}_{u}$	$F^{(4,4)}_{u}$	$F^{(4,5)}_{u}$	$F^{(4,6)}_{u}$
k1=5	$F^{(5,1)}_{u}$	$F^{(5,2)}_{u}$	$F^{(5,3)}_{u}$	$F^{(5,4)}_{u}$	$F^{(5,5)}_{u}$	$F^{(5,6)}_{u}$
k1=6	$F^{(6,1)}_{u}$	$F^{(6,2)}_{u}$	$F^{(6,3)}_{u}$	$F^{(6,4)}_{u}$	$F^{(6,5)}_{u}$	$F^{(6,6)}_{u}$
k1=7	$F^{(7,1)}_{u}$	$F^{(7,2)}_{u}$	$F^{(7,3)}_{u}$	$F^{(7,4)}_{u}$	$F^{(7,5)}_{u}$	$F^{(7,6)}_{u}$
k1=8	$F^{(8,1)}_{u}$	$F^{(8,2)}_{u}$	$F^{(8,3)}_{u}$	$F^{(8,4)}_{u}$	$F^{(8,5)}_{u}$	$F^{(8,6)}_{u}$
k1=9	$F^{(9,1)}_{u}$	$F^{(9,2)}_{u}$	$F^{(9,3)}_{u}$	$F^{(9,4)}_{u}$	$F^{(9,5)}_{u}$	$F^{(9,6)}_{u}$
k1=10	$F^{(10,1)}_{u}$	$F^{(10,2)}_{u}$	$F^{(10,3)}_{u}$	$F^{(10,4)}_{u}$	$F^{(10,5)}_{u}$	$F^{(10,6)}_{u}$
k1=11	$F^{(11,1)}_{u}$	$F^{(11,2)}_{u}$	$F^{(11,3)}_{u}$	$F^{(11,4)}_{u}$	$F^{(11,5)}_{u}$	$F^{(11,6)}_{u}$
k1=12	$F^{(12,1)}_{u}$	$F^{(12,2)}_{u}$	$F^{(12,3)}_{u}$	$F^{(12,4)}_{u}$	$F^{(12,5)}_{u}$	$F^{(12,6)}_{u}$
k1=13	$F^{(13,1)}_{u}$	$F^{(13,2)}_{u}$	$F^{(13,3)}_{u}$	$F^{(13,4)}_{u}$	$F^{(13,5)}_{u}$	$F^{(13,6)}_{u}$
k1=14	$F^{(14,1)}_{u}$	$F^{(14,2)}_{u}$	$F^{(14,3)}_{u}$	$F^{(14,4)}_{u}$	$F^{(14,5)}_{u}$	$F^{(14,6)}_{u}$
k1=15	$F^{(15,1)}_{u}$	$F^{(15,2)}_{u}$	$F^{(15,3)}_{u}$	$F^{(15,4)}_{u}$	$F^{(15,5)}_{u}$	$F^{(15,6)}_{u}$
k1=16	$F^{(16,1)}_{u}$	$F^{(16,2)}_{u}$	$F^{(16,3)}_{u}$	$F^{(16,4)}_{u}$	$F^{(16,5)}_{u}$	$F^{(16,6)}_{u}$
k1=17	$F^{(17,1)}_{u}$	$F^{(17,2)}_{u}$	$F^{(17,3)}_{u}$	$F^{(17,4)}_{u}$	$F^{(17,5)}_{u}$	$F^{(17,6)}_{u}$
k1=18	$F^{(18,1)}_{u}$	$F^{(18,2)}_{u}$	$F^{(18,3)}_{u}$	$F^{(18,4)}_{u}$	$F^{(18,5)}_{u}$	$F^{(18,6)}_{u}$