Lecture Series on

# Intelligent Control

Lecture 20

**Fuzzy Logic Controller Design**

Kwang Y. Lee
Professor of Electrical Engineering
Baylor University
Waco, TX 76798, USA
Kwang_Y_Lee@baylor.edu

1

1

## Fuzzy Logic Controller Design

- Many Fuzzy control can be classified into *static* fuzzy control and *adaptive* fuzzy control. In static fuzzy control, the structure and parameters of the fuzzy controller are fixed and do not change during real-time operation.
- On the other hand, in adaptive fuzzy control, the structure and/or parameters of the fuzzy controller change during real-time operation. Fixed fuzzy control is simpler than adaptive fuzzy control but requires more knowledge of the process model or heuristic rules. Adaptive fuzzy control, on the other hand, is more expensive to implement, but requires less information and may perform better.

2

2

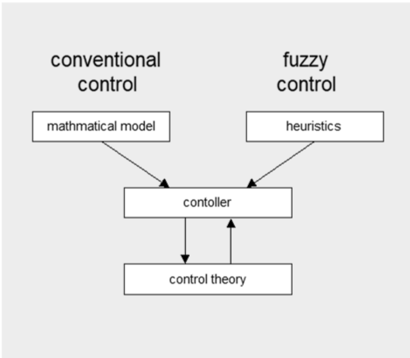
## Fuzzy Logic Controller Design

- *Fuzzy* control and *conventional* control have similarities and differences. They are similar in the sense that they must address the same issues that are common to any control problem, such as stability and performance. However, there is a fundamental difference between fuzzy control and conventional control.
- *Conventional control* starts with a mathematical model of the process and controllers are designed based on the model. *Fuzzy control*, on the other hand, starts with heuristics and human expertise (in terms of fuzzy *IF-THEN* rules) and controllers are designed by synthesizing these rules. That is, the information used to construct the two types of controllers are different.

3

3

## Fuzzy Logic Controller Design




4

4

## Fuzzy Logic Controller Design

- For many practical problems, it is difficult to obtain an accurate yet simple mathematical model, but there are human experts who can provide heuristics and rule-of-thumb that are very useful for controlling the process.
- Fuzzy control is most useful for these kinds of problems. If mathematical model of the process is unknown, we can design fuzzy controllers in a systematic manner that guarantee certain key performance criteria.
- The design techniques for fuzzy controllers can be classified into the *trial-and-error* approach and the *theoretical* approach.

5

5

## Fuzzy Logic Controller Design

- In the *trial-and-error approach*, a set of fuzzy IF-THEN rules are collected from human experts or documented knowledge base, and the fuzzy controllers are constructed from these fuzzy IF-THEN rules.
- The fuzzy controllers are tested in the real system and if the performance is not satisfactory, the rules are fine-tuned or redesigned in several trial-and-error cycles until the performance is satisfactory.
- In *theoretical approach*, the structure and parameters of the fuzzy controller are designed in such a way that certain performance criteria are guaranteed. Both approaches, of course, can be combined to give the best fuzzy controllers.

6

6

## Trial-and-Error Approach

- Select *state* and *control* variables. The state variables should characterize the key features of the system and the control variables should be able to influence the states of the system. The state variables are the inputs to the fuzzy controller and the control variables are the output of the fuzzy controller.
- Construct *IF-THEN rules* between the state and control variables. The formulation of these rules can be achieved in two different heuristic approaches. The most common approach is the *linguistic verbalization* of human experts. Another approach is to interrogate experienced experts or operators using a carefully organized questionnaire.

7

7

## Trial-and-Error Approach

- Test the fuzzy IF-THEN rules in the system. The closed-loop system with the fuzzy controller is run and if the performance is not satisfactory, fine tune or redesign the fuzzy controller and repeat the procedure until the performance is satisfactory.
- The resulting fuzzy IF-THEN rule can be in the following two types:

Type I: IF $x_1$ is $A_1^i$ AND ... AND $x_n$ is $A_n^i$, THEN $u$ is $B^j$

Type II:IF $x_1$ is $A_1^i$ AND ... AND $x_n$ is $A_n^i$,

THEN $u$ is $c_0^i + c_1^i x_1 + ... + c_n^i x_n$

8

8

## Trial-and-Error Approach

- In Type I, both the *antecedent* and *consequence* have linguistic variables. On the other hand, in Type II, the consequent is a parameterized function of the input to the fuzzy controller, or the state variables.
- Comparing the two types, the THEN part of the rule is changed from a linguistic description to a simple mathematical formula. This change makes it easier to combine the rules. In fact, Type II, the Takagi-Sugeno system, is a weighted average of the rules in the THEN parts of the rules.

9

9

## Trial-and-Error Approach

- Type II is useful in tuning the rules mathematically.
- On the other hand, it has drawbacks:
  - THEN part is a mathematical formula and therefore may not provide a natural framework to represent human knowledge, and
  - There is not much freedom left to apply different principles in fuzzy logic, so that the versatility of fuzzy systems is not fully represented in this framework.

10

10

## Theoretical Approach

- In order to analyze the performance of the closed-loop fuzzy control system theoretically, we need to have some knowledge on the model of the system; assumes a mathematical model for the system, so that mathematical analysis can be performed to establish the properties of the designed system.
- Theoretical approach can be classified into the following categories:
  1. Stable controller design
  2. Optimal controller design
  3. Sliding mode controller design
  4. Supervisory controller design
  5. Fuzzy system model-based controller design

11

11

## Automatic Tuning Method

- The variables of the premise and the consequent are defined as the following:

  Error ($E$) = process output - set point

  Error change ($DE$) = current error - last error

  Controller output = input applied to process.
- The domain of a variable, $E$ or $DE$, is partitioned into fuzzy sets, $A_i$ , $i$ = 1, …, $n$.
- Every fuzzy set is associated with a name that represents qualitative statements, e.g., for $i$ = 1, …, 5, $A_1$ = large negative ($LN$), $A_2$ = small negative ($SN$), $A_3$ = zero ($ZE$), $A_4$ = small positive ($SP$), and $A_5$ = large positive ($LP$).

12

12

## Automatic Tuning Method

- An example of a rule, where the consequent of the rule is a parameterized function of the input variables, is:

  *IF* error ($E$) is large negative ($i$ = 1) and the change in error ($DE$) is small negative ($j$ = 2),

  *THEN* the output is

  $$u_{12} = c_{12}^0 + c_{12}^1 E + c_{12}^2 DE \quad ,$$

  where the subscripts represent $Rule_{12}$, and the parameters $c_{12}^k$ , $k$ = 0, 1, and 2, need to be determined.
- In general, the parameters for $Rule_{ij}$ , for all $i$ and $j$, are determined by the Automatic Tuning Method (ATM) using the input and output data from the experiment (Ramaswami, 1993).

13

13

## Automatic Tuning Method

Most existing fuzzy logic controllers are designed without using any mathematical model of a plant. The construction procedures are generally based on the experts' understanding of the process. Therefore, the rule base of a fuzzy logic controller must be adjusted through trial and error to obtain the desired performance.

The consequent of each rule of the controller has the form

$$u_{ij} = c^0 + c_{ij}^1 E + c_{ij}^2 DE \quad ,$$

where $c^0$ is known steady-state controller output, and $c_{ij}^1$ and $c_{ij}^2$ are the unknown parameters.

14

14

## Automatic Tuning Method

To find these unknowns, the Kalman filter approach is taken because the Kalman filter estimates are the optimal mean-squared error estimates. Also, in this recursive filter there is no need to store past measurements for the purpose of computing present estimates.

In order to apply the Kalman filtering, the unknown parameters $c_{ij}^l$ are viewed as state variables, the premise variables $E(k)$ and $DE(k)$ as time-varying system coefficients, and the $u_{ij}$ as the system output variables. Then the dynamics of $c_{ij}^l$ can be modeled simply as a stochastic system in discrete-time:

15

15

## Automatic Tuning Method

*System Model:*

$$\begin{bmatrix} c_{ij}^1(k) \\ c_{ij}^2(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} c_{ij}^1(k-1) \\ c_{ij}^2(k-1) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} w_{(k-1)}$$

$$w_k \approx N(0, \infty)$$

*Measurement Model:*

$$u_{ij} = \begin{bmatrix} E(k) & DE(k) \end{bmatrix} \begin{bmatrix} c_{ij}^1(k) \\ c_{ij}^2(k) \end{bmatrix} + v_k + c^0$$

$$v_k \approx N(0, \infty) \qquad R_k^{-1} = \infty$$

16

16

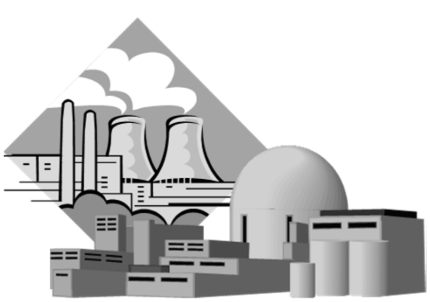
## Automatic Tuning Method

Here $w_k$ and $v_k$ are process and measurement noise, respectively, with normal distribution. In this formulation, the process noise is assumed to be completely unknown, and the measurement model is assumed to have zero measurement noise. The parameters are unknown constants and therefore their changes at steady-state are zero. Also, the variations of the two parameters are uncorrelated. From these initial assumptions for the system model, the Kalman filtering problem can be easily solved to give the steady-state solution for the parameters $c_{ij}^l$.

17

17

# Nuclear Reactor Control



18

18

## NUCLEAR REACTOR CONTROL

**The Reactor Power Plant Modeling**

$$\frac{d}{dt}n = \frac{\delta\rho - \beta}{\Lambda}n + \lambda c \qquad \frac{d}{dt}n_r = \frac{\delta\rho - \beta}{\Lambda}n_r + \frac{\beta}{\Lambda}c_r$$

$$\frac{d}{dt}c = \frac{\beta}{\Lambda}n - \lambda c, \qquad \frac{d}{dt}c_r = \lambda n_r - \lambda c_r .$$

$$P_a(t) = P_{0a}n_r(t), \qquad P_c(t) = \Omega(T_f - T_c) \qquad P_e(t) = M(T_l - T_e),$$

$$f_f P_a(t) = \mu_f \frac{d}{dt}T_f + P_c(t) \qquad (1 - f_f)P_a(t) + P_c(t) = \mu_c \frac{d}{dt}T_l + P_e(t),$$

$$\delta\rho = \delta\rho_r + \alpha_f(T_f - T_{f0}) + \alpha_c(T_c - T_{c0})$$

$$\frac{d}{dt}\delta\rho_r = G_r z_r$$

19

19

### Simulation Results

**Operation Regions**

| $G_r$ \ $n_{r0}$ | 0.1 | 0.5 | 1.0 |
|---|---|---|---|
| 0.0290 | Region 3 | Region 4 | Region 5 |
| 0.0145 | Region 2 | Region 1 | Region 6 |
| 0.0070 | Region 9 | Region 8 | Region 7 |

**Test Case Studies:**

**Case A:** Local control
100% → 90% → 100% power level changes in Region 6.
**Case B:** Global operation
40% → 50% → 40% power level changes in Region 1.
**Case C:** Emergency operation
100% → 25% huge step down from Region 5 to Region 3.
**Case D:** Shut-down/Start-up
100% → 10% → 100% ramp down and ramp up from Region 5 to Region 3.

20

20

# Automatic Tuning Method

TABLE II
ESTIMATED PARAMETERS FOR RULE$_{ij}$

| $i$ | $j$ | $c^1_{ij}$ | $c^2_{ij}$ |
|---|---|---|---|
| 1 | 1 | −14.15933 | −13.54221 |
| 1 | 2 | −8.544655 | −24.15133 |
| 1 | 3 | −6.055618 | −26.03189 |
| 1 | 4 | 0.0000000 | 0.0000000 |
| 1 | 5 | −0.2768038 | 0.5536076 |
| 2 | 1 | −17.31362 | −11.79659 |
| 2 | 2 | −8.087297 | −23.53640 |
| 2 | 3 | −4.705722 | −24.54666 |
| 2 | 4 | 0.0000000 | 0.0000000 |
| 2 | 5 | −0.2768038 | 0.5536076 |
| 3 | 1 | 0.0000000 | 0.0000000 |
| 3 | 2 | −15.16420 | −13.17804 |
| 3 | 3 | −15.16420 | −13.17804 |
| 3 | 4 | 0.0000000 | 0.0000000 |
| 3 | 5 | 0.0000000 | 0.0000000 |
| 4 | 1 | −16.14068 | 0.3889777 |
| 4 | 2 | −34.63131 | −19.88403 |
| 4 | 3 | −6.911923 | 6.517233 |
| 4 | 4 | 0.0000000 | 0.0000000 |
| 4 | 5 | 0.0277524 | 0.5550497 |
| 5 | 1 | −16.14068 | 0.3889777 |
| 5 | 2 | −35.12094 | −20.36247 |
| 5 | 3 | −5.204771 | 4.770062 |
| 5 | 4 | 0.0000000 | 0.0000000 |
| 5 | 5 | −0.0275841 | −0.5516833 |

21

21

## Automatic Tuning Method

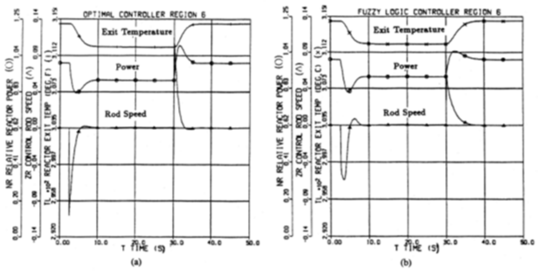



Fig. 5. Case A: Local control 100% → 90% → 100% power level change in region 6. (a) Optimal controller. (b) Fuzzy logic controller.

22

22

## Self-Organizing Fuzzy Logic Control

- Rules generated using the history of input-output data
- Fuzzy rule base updated on-line by a self-organizing procedure

Application of FARMA FLC to a Boiler-Turbine System

- Single loop control scheme
- Three input-output pairs of dominant relations
- Application of FARMA FLC to each single loops

23

23

## Free Model

Discrete Plant:

$$y(k+1) = f\,(y(k), y(k\text{-}1), ..., u(k), u(k\text{-}1),... )$$

Free Model:

$$y(k+1) = f\,(y(k), \Delta y(k), \Delta^2 y(k), \Delta^3 y(k), ..., u(k), u(k\text{-}1), \Delta^2 u(k\text{-}1), \Delta^3 u(k\text{-}1), ... )$$

where $\Delta^i$ is the Backward Difference Operator:

$$\Delta^i f(k) = \Delta^{i\text{-}1} f(k) \text{ - } \Delta^{i\text{-}1} f(k\text{-}1),\ \ \Delta^0 f(k) = f(k)$$

Inverse Model:

$$u(k) = g(y_{ref}, y(k), \Delta y(k), ..., u(k\text{-}1), \Delta u(k\text{-}1), ... )$$

24

24

## FARMA Rule

***IF*** $y_{ref}$ is $A_{1i}$, $y(k)$ is $A_{2i}$, $\Delta y(k)$ is $A_{3i}$,..., $\Delta^{n-1}y(k)$ is $A_{(n+1)i}$,
***AND*** $u(k\text{-}1)$ is $B_{1i}$, $\Delta u(k\text{-}1)$ is $B_{2i}$, ..., $\Delta^{m-1}u(k\text{-}1)$ is $B_{mi}$,
***THEN*** $u(k)$ is $C_i$ (for the *i*-th rule)

where $n$, $m$: number of output and input variables
$A_{ij}$, $B_{ij}$: antecedent linguistic values for the *i*-th rule
$C_i$ : consequent linguistic value for the *i*-th rule

**Membership function** $A_i$ for a crisp value $x_1$:

$$\mu_{A_i} = \begin{cases} 1+(x-x_1)/(b-a) & if \quad a \le x < x_1 \\ 1-(x-x_1)/(b-a) & if \quad x_1 \le x < b \\ 0 & else \end{cases}$$

where, $[a, b]$ is input or output range

25

25

## FARMA Rule

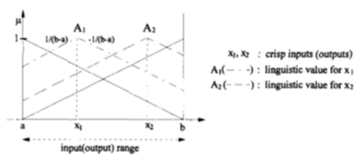



Fig. 1. The fuzzification procedure for $A_{ij}$, $B_{ij}$, or $C_i$.

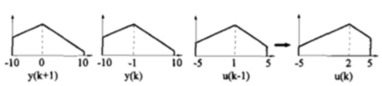



Fig. 2. The generation of a FARMA rule.

26

26

## FARMA Rule

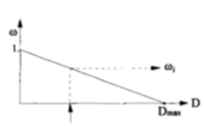



Fig. 4. The truth value with the similarity function.

$$D_i = \sqrt{(x_{1i}-x_1)^2+(x_{2i}-x_2)^2+\cdots+(x_{(n+m+1)i}-x_{(n+m+1)})^2} \quad (6)$$

(for the ith rule)

where
$x_1, x_2, \cdots,$ : crisp input variables
$x_{1i}, x_{2i}, \cdots$ : vertices of the membership functions for $A_{1i}, A_{2i}, \cdots, B_{1i}, B_{2i}, \cdots$.

27

27

## FARMA Rule

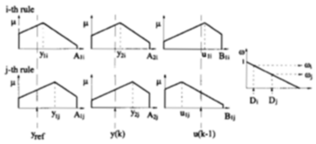



Fig. 5. Inference with the similarity function.

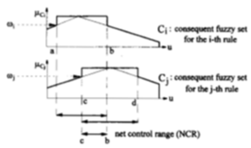



Fig. 6. The net control range (NCR) by the φ-opertion with two rules.

28

28

## FARMA Rule

**Inference** with φ-operation

$$C_n = \bigcap_i (\omega_i \varphi \mu_{C_i})$$

$$\omega_i \varphi \mu_{C_i} = \begin{cases} 1 & if \quad \omega_i \le \mu_{C_i} \\ 0 & if \quad \omega_i > \mu_{C_i} \end{cases}$$

where $C_n$ : net linguistic control action

$\omega_i$ : truth value of the $i$-th rule

$\mu_{Ci}$: membership degree of linguistic value $C_i$

**Defuzzification**

- Net Control Range (NCR)
  - $\alpha$-cut of the $C_n$ where $\alpha = max\ \mu(C_n)$
  - Subset $[p, q]$ of $[a, b]$ as the highest possibility

29

29

## FARMA Rule

- Temporary target $y_r(k+1)$

$$y_r(k+1) = y(k) + \beta\,(y_{ref} - y(k))$$

where β is the target ratio constant ($0 < \beta \le 1$)

- The final crisp control value $u(k)$

$$u(k) = \begin{cases} u(k-1) + q)/2 & for \quad y_r(k+1) > \hat{y}(k+1) \\ (p + u(k-1))/2 & for \quad y_r(k+1) < \hat{y}(k+1) \end{cases}$$

where $[p \ \ q]$ is the NCR.

30

30

## FARMA Rule

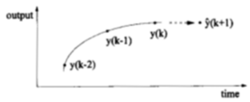



Fig. 7. Estimation of $y(k+1)$ by the second extrapolation.

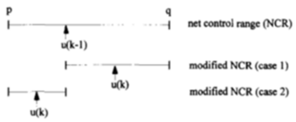



Fig. 8. The modification of the net control range and defuzzification.

*Case 1*: $\hat{y}(k+1) < y_r(k+1)$
*Case 2*: $\hat{y}(k+1) > y_r(k+1)$.

31

31

## Self-Organization

**Rule base update:**

- Performance index

  $J = |y_r(k+1) - y(k+1)|$

where $y(k+1)$: real plant output
$y_r(k+1)$: reference output

- Partition of the fuzzy rule space into a finite number of domains
  - Only one rule, i.e., a point, is stored in each domain
  - If there is a new rule in domain, replace it with the smaller $J$

32

32

## Self-Organization

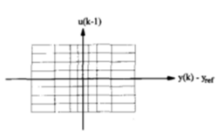



Fig. 9. Division of a 2-dimensional rule space.

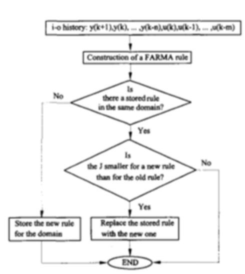
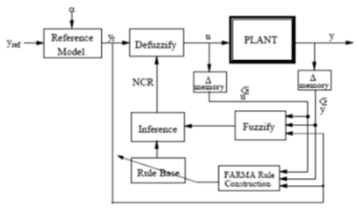



Fig. 10. The self-organization of the rule base.

33

33

## Self-Organizing Fuzzy Logic Control




The FARMA control system architecture

34

34

## Boiler-Turbine System

A MIMO nonlinear Model of a Boiler-Turbine System:

$$\dot{x}_1 = -0.0018 u_2 x_1^{9/8} + 0.9 u_1 - 0.15 u_3$$

$$\dot{x}_2 = [(0.73 u_2 - 0.16) x_1^{9/8} - x_2]/10$$

$$\dot{x}_3 = [141 u_3 - (1.1 u_2 - 0.19) x_1]/85$$

$$L = 0.05(0.13073 x_3 + 100 a_{cs} + q_e/9 - 67.975)$$

where

$$\alpha_{cs} = \frac{(1 - 0.001538 \rho_f)(0.8p - 25.6)}{\rho_f (1.0394 - 0.0012304 p)}$$

$$q_e = (0.854 u_2 - 0.147) p + 45.59 u_1 - 2.514 u_3 - 2.096$$

35

35

## Input-Outputs

- Output variables
  - $y_1$ : drum steam pressure ($P$ in kg/cm$^2$)
  - $y_2$ : electric power ($E$ in MW)
  - $y_3$ : drum water level deviation ($L$ in m)
- Input variables
  - $u_1$ : fuel valve position
  - $u_2$ : steam valve position
  - $u_3$ : feedwater valve position

➔ Limited change of input variables

36

36

## MIMO Fuzzy Logic Control

Dominant Input-Output Pairing:

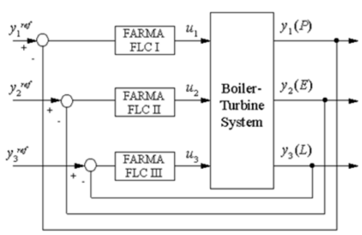
- Drum steam pressure ($y_1$): control with fuel valve ($u_1$)
- Electric power($y_2$): control with steam valve ($u_2$)
- Drum water level ($y_3$): control with feedwater valve ($u_3$)

Three Independent Single-Input-Single-Output Loops:
Three FARMA FLCs

37

37

## Boiler-Turbine Control System




Three SISO FARMA FLCs for a MIMO System

38

38

## Simulation Results

Initial steady state:

$X$=(100, 50, 449.5), $Y$=(100, 50, 0), $U$=(0.271, 0.604, 0.336)

Four cases of step-changes in references:

Case 1) $y_1^{ref} = 110, y_2^{ref} = 80, y_3^{ref} = 0,$

Case 2) $y_1^{ref} = 120, y_2^{ref} = 100, y_3^{ref} = 0,$

Case 3) $y_1^{ref} = 130, y_2^{ref} = 120, y_3^{ref} = 0,$

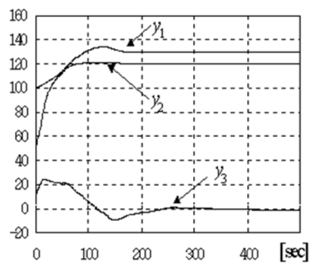
Case 4)
$$\begin{cases} y_1^{ref} = 110, y_2^{ref} = 80, y_3^{ref} = 0, & \text{for } 0 < t < 400 \\ y_1^{ref} = 120, y_2^{ref} = 100, y_3^{ref} = 0, & \text{for } 400 \le t < 800 \\ y_1^{ref} = 130, y_2^{ref} = 120, y_3^{ref} = 0, & \text{for } 800 \le t \le 1200 \end{cases}$$

39

39

# Simulation Results

Case 1

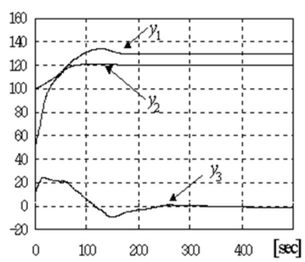



40

40

# Simulation Results

Case 3

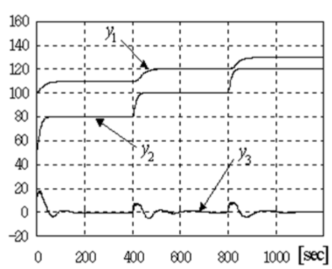



41

41

# Simulation Results

Case 4




42

42

## Conclusions

Free-Model based Self-Organizing Fuzzy Logic Control

- Automatic generation of rules and the membership functions
- On-line self-organization of fuzzy rule base

Application to a Boiler-Turbine System

- Three input-output pairs of dominant relations
- Application of FARMA FLC to each single loops

Successful simulation results in a MIMO nonlinear model

43

43