CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

LABORATOIRE D'ANALYSE ET D'ARCHITECTURE DES SYSTEMES

ON FUZZY CONTROL OF WATER DESALINATION PLANTS

F. OLAFSSON, M. JAMSHIDI, A. TITLI

LAAS REPORT 95197

MAY 1995

CNRS-LAAS--95197

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

LABORATOIRE D'ANALYSE ET D'ARCHITECTURE DES SYSTEMES

ON FUZZY CONTROL OF WATER DESALINATION PLANTS

F. OLAFSSON, M. JAMSHIDI, A. TITLI

MASTER

LAAS REPORT 95197

MAY 1995

On fuzzy control of water desalination plants

Finnur Olafsson [1]
Mohammad Jamshidi [2]
André Titli

Groupe Automatique Symbolique
LAAS-CNRS
7 avenue du Colonel Roche
31077 Toulouse
FRANCE

1. Norwegian Institute of Technology, Trondheim, Norway.
2. University of New Mexico, Albuquerque, NM, USA

Toulouse, France. May, 1995

Table of Contents

Preface

This report is submitted to the International Foundation For Water Science and Technology (IFFWSAT, Abu Dhabi, the United Arab Emirates) as the feasibility and initial report on a multi-year research in the intelligent control of water desalination systems.

The project was performed at *Laboratoire d'Analyse et d'Architecture des Systèmes*, L.A.A.S, Toulouse, France. This was made possible in part through support of IFFWSAT and the ERASMUS exchange program between NTH (Norwegian Institute of Technology) and INSAT (*Institut National des Sciences Appliquées* Toulouse). One month was spent at IFFW-SAT, with Dr. Darwish M. K Al-Gobaisi as founder and Director and his team of researchers and scientists.

Acknowledgements

The supports of CNRS and IFFWSAT are sincerely appreciated. The vision of Dr. Darwish M. K Al-Gobaisi for the recognition of the fact that fuzzy logic and artificial intelligence techniques and tools have a potential impact on water reuse and desalination systems is greatly acknowledged and appreciated.

ABSTRACT

In this report we have chosen a sub-system of an MSF water desalination plant, the brine-heater, for analysis, synthesis, and simulation. This system has been modelled and implemented on computer. A fuzzy logic controller (FLC) for the top brine temperature control loop has been designed and implemented on the computer. The performance of the proposed FLC is compared with three other conventional control strategies: PID, cascade and disturbance rejection control.

One major concern on FLC's has been the lack of stability criteria. An up to-date survey of stability of fuzzy control systems is given. We have shown stability of the proposed FLC using the Sinusoidal Input Describing Functions (SIDF) method. The potential applications of fuzzy controllers for complex and large-scale systems through hierarchy of rule sets and hybridization with conventional approaches are also investigated.

1 Introduction

Water is a very essential ingredient for life and the world's most precious resource. Without it, no one can live for a long period of time, and is thus an essential commodity for any society. In areas where water is not found naturally (as ground-water, fresh water lakes, rivers or rain), distillation of seawater (known as *desalination*) may be both economic and justified. This especially applies to the Middle East, where water resources are scarce and often over-exploited, and where they have access to low cost energy. In this region water also is a major factor for maintaining peace and stability.

Today there are several different desalination methods in use, which can broadly be classified as membrane and distillation techniques.

- Membrane techniques
  Two commercially important desalination processes, electrodialysis and reverse osmosis (RO) uses membranes to separate salt and water. These two techniques are mainly used for brackish water, and in small to medium sized productions.
- Distillation techniques
  These processes uses thermal energy to vaporise the seawater, and the vapour produced is condensed as fresh water. Three major techniques exists; Multi-Stage Flash (MSF), Multi-Effect Distillation (MED) and Vapour Compression (VC). The two first are often installed in conjunction with an electricity plant, using available surplus steam, while VC uses electrical energy.

The choice of technology depends on several factors such as energy availability, desired production, salinity of water etc. For large productions using seawater, MSF desalination is the most widely applied process.

The control of MSF plants has more or less remained unchanged over the past 30 years, and initiative has been taken to investigate modern control techniques (Al-Gobaisi *et al.*,1994a; Rao *et al.*,1994). In order to get a better understanding of the process, rigorous dynamic non-linear models of MSF plants are under development (Husain *et al.*, 1994a,b). These models can also be used for studying the effect of advanced control methods, and in the design of such control systems.

In order to continuously improve the operating performance and plant availability, new control strategies are being investigated. One such control method is *fuzzy control.* Fuzzy control has turned out to be suitable for controlling non linear processes, and systems with imprecise, or those in which a mathematical model is not readily available. Fuzzy control is based on fuzzy sets and logic, which was first proposed in the seminal paper on fuzzy sets by Zadeh (1965). A fuzzy set has, unlike a classical set, unsharp boundaries, with a gradual transition from membership to non-membership. This coincides more with the real world where things are seldom black or white, but rather shades of grey in between.

The main area of application of fuzzy logic has been in control engineering, where we have numerous examples ranging from control of cement kilns (Larsen, 1980), control of subway systems (Yasunobu and Hasegawa, 1985) to autonomous control of a model car (Sugeno and Nishida, 1985); the first laboratory implementation came in (Mamdani, 1974) with the control of a steam engine.

The main purpose of this work is, therefore, to choose a suitable sub-system of the MSF plant to be controlled, implement a stable fuzzy controller, and compare its performance with conventional methods.

This report is organised as follows:

Chapter 2 gives a description of MSF desalination plants and a non-linear model of the brine-heater. Theory about fuzzy sets, logic and control follows in Chapter 3. Stability of fuzzy control systems, including the top brine heater system is covered in Chapter 4. Simulations and results are presented in Chapter 5. The notions of hierarchy and hybridization of fuzzy logic with conventional or other AI-based approaches are briefly discussed in Chapter 6. The conclusions and discussions are presented in Chapter 7.

2 Multi Stage Flash Desalination Plant

The basic principle of thermal desalination is vaporising the brine[1]. The vapour obtained is very pure, and when it condenses it gives high quality fresh water. The brine's boiling-point is a function of pressure and salt concentration. When at its boiling-point, vapour can be obtained in two ways, either by heat addition or by pressure reduction. The former is known as *boiling* and the latter as *flashing*. When these processes are put in subsequent stages it is called Multi-Effect Distillation (MED) and Multi-Stage Flash (MSF). In large-scale desalination plants, the MSF principle is the most used of the two technologies.

In this chapter a description of the MSF process and control is presented. A dominant subsystem of the plant, the brine heater, is chosen for further study. A non-linear mathematical model of the brine-heater will be given.

2.1 Process Description

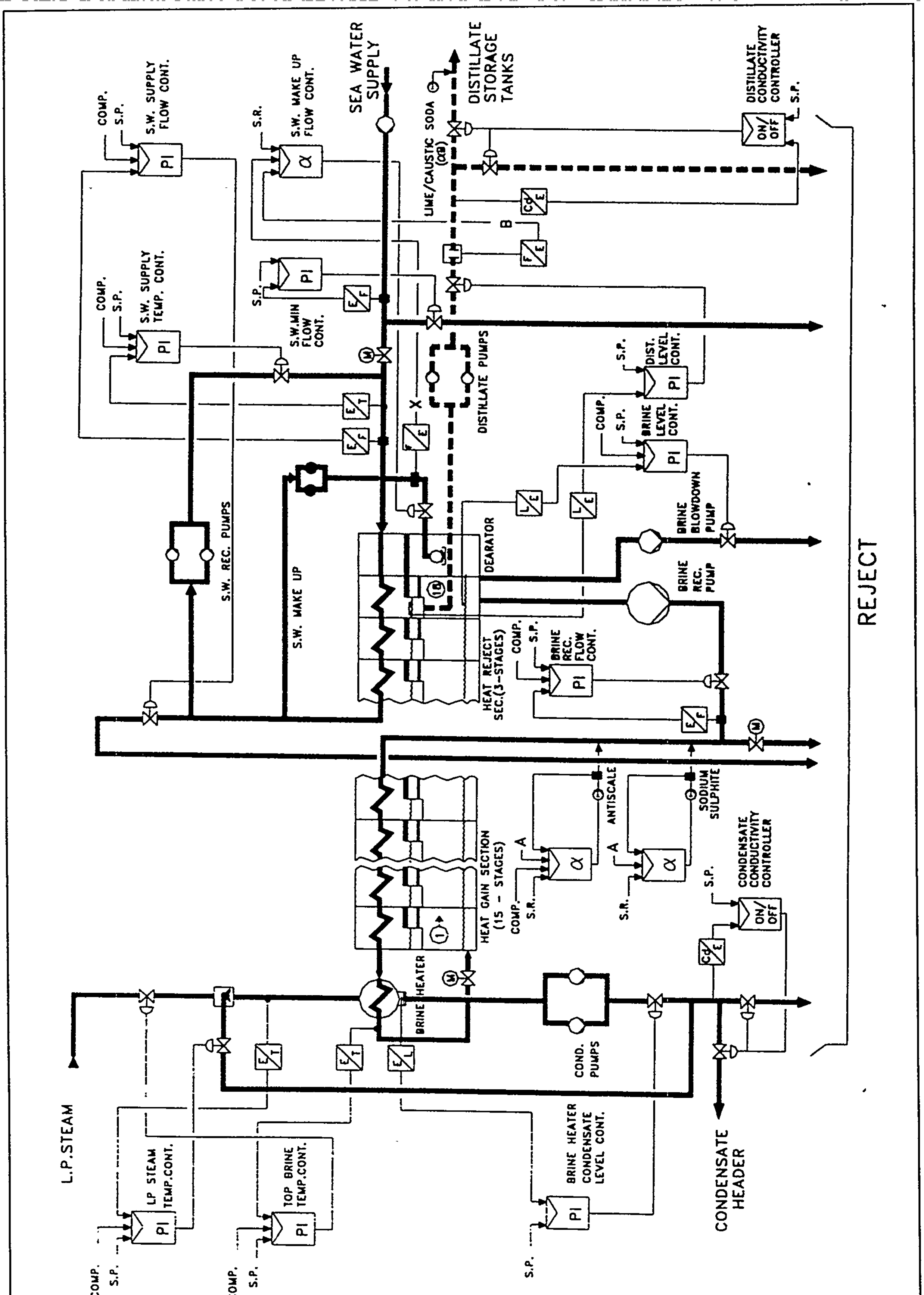
A typical MSF plant is shown in Figure 2-1. There are three mains sections: the heat rejection stages, the heat recovery (or gain) stages and the brine heater. The heat rejection and recovery section consists of several stages. One of these stages is illustrated in Figure 2-2. The number of stages is determined according to the desired production rate and performance, an example is 15 for the recovery section and 3 for the rejection section.

Seawater enters the last stage of the rejection section and flows through a series of heat exchangers. Upon leaving the first stage of the rejection section, a part of the heated seawater is rejected back to the sea (thus the name rejection section). The rest is fed through a *dearator* as *make up* flow to be mixed with concentrated brine in the last stage. The dearator removes oxygen to prevent corrosion of tubes. During the winter season, the seawater is mixed with rejected seawater to maintain a minimum seawater temperature.

From the last stage of the rejection section, part of the concentrated brine is rejected to the sea as *blow-down*, while most is pumped into the last stage of the recovery section. This flow is known as the *recycle flow*, and is the most important flow in the process. The main reason for recycling the brine is the high cost of chemical additives which is added to prevent scaling[2] of salts in the tubes.

The recycle flow now passes through more heat-exchangers in the recovery section, its temperature increases as it proceeds towards the brine heater. The brine heater is the heat input section of the plant. In the brine heater the temperature is additionally increased to a maximum value, the Top Brine Temperature (TBT). Here the brine reaches its saturation temperature at the system pressure. The heating medium is low pressure, saturated steam. The steam normally comes from an adjacent power plant, thus exploiting low cost surplus steam. The temperature of the steam is controlled by a *desuperheater*, where condensate from the brine heater is sprayed on the steam. This is done to ensure dry and saturated steam flow at constant tem-

1. a strong saline solution, i.e. stronger than seawater
2. The literature uses both *scaling* and *fouling* about deposits in tubes.



Courtesy of IFFWSAT

Figure 2-1: Typical MSF desalination plant

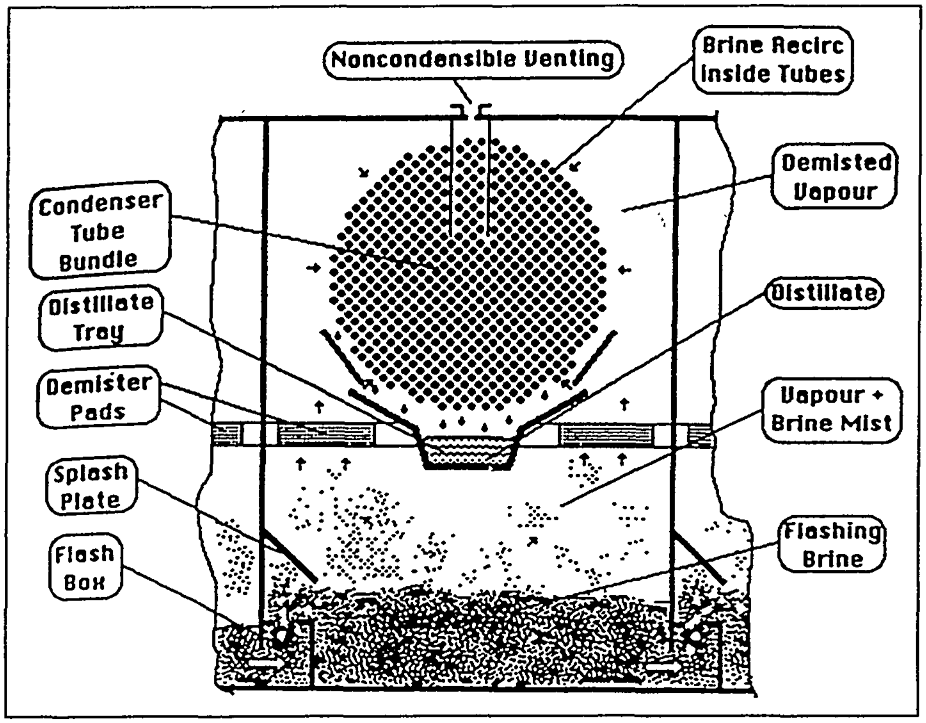


Figure 2-2: A stage in an MSF plant (Hanbury *et al.*, 1993)

perature.

The brine then flows on the floor into the first stage of the recovery section through an orifice, thus reducing pressure. As the brine already was at its saturation temperature for a higher pressure, it will become superheated and flashes to give off vapour to become saturated again. The vapour passes through demisters (wire mesh) to remove any entrained brine droplets, and condensates on the tubes of the counter-flowing recycling brine. The condensate drips from the tube surface into a distillate tray, and flows through all the stages in a channel. The heat of condensation supplies a large part of the heat required to raise the feed to its boiling-point.

The pressure in each stage is lower than the pressure in the preceding stage, the pressure drop being maintained by the brine level which acts like a liquid seal. If the brine level is lower than the orifice height, the inter-stage pressure difference is lost, and hence no flashing will occur (this is known as *blow through*).

The process of flashing and condensation is repeated in each stage as the brine flows towards the last stage of the rejection section. The brine level profile in all of the stages is controlled by the brine level in the last stage by the blow-down flow. Here the concentrated brine is mixed with fresh make up, and the whole process is repeated. The distilled water is pumped from the last stage, chemically treated and stored in storage tanks.

2.1.1 Operational Factors

There are principally two ways to increase the recovery rate (production).

- Increase the flash range

  The flash range is defined as the difference in temperature between entering seawater temperature and the TBT. The flash range can thus be increased either by increasing TBT or by decreasing the temperature of the entering flow to the last stage.

- The limit of TBT is determined by considerations of scaling. The TBT can be increased by adding costly high-temperature additives. The limit is then about 120°C. The steam supply is normally not a limiting factor.
- The temperature of seawater is given by nature and can not be changed. In the winter season the seawater feed is limited to about 24°C by adding rejected seawater from the first rejection stage.

- Increase the recycling flow

  The upper limit here is given by the recycle flow velocity, which must be maintained below an upper limit to avoid corrosion on the tubes. In the other end, one must have a minimum flow to avoid fouling of the tubes.

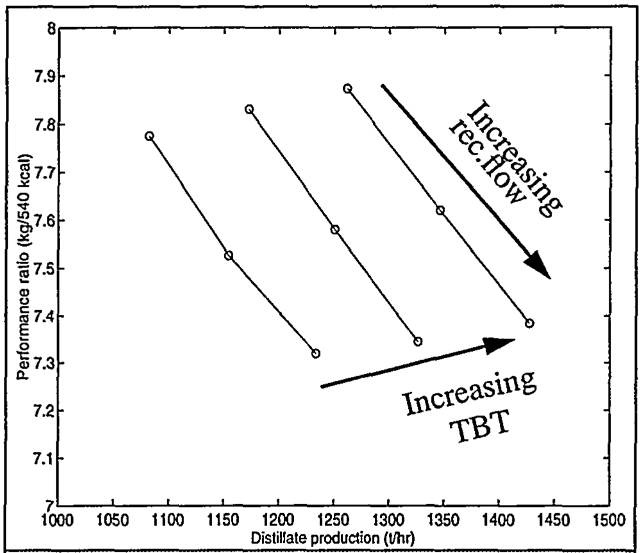
Increased recovery rate can also be improved by design, choice of materials, improved anti fouling additives etc. This will not be discussed here.

2.1.2 Optimal Performance

The operating range for a plant is between 60% and 120% of nominal production. One of the performance ratios used is:

$$PR = \frac{\text{kg of product}}{\text{kg of steam at 540 kcal/kg}} \tag{2-1}$$

It is mainly a function of the TBT and the recycle flow. An operational plot, as shown in Figure 2-3, shows *PR* at different combinations of TBT and recycle flow. One could be misled to believe that the plant should always be run at its maximum performance ratio. This is not the case. For instance during summer the demand of water is high, and the plant must run at maximum production at any cost.



Data supplied by Mr. Woldai at IFFWSAT

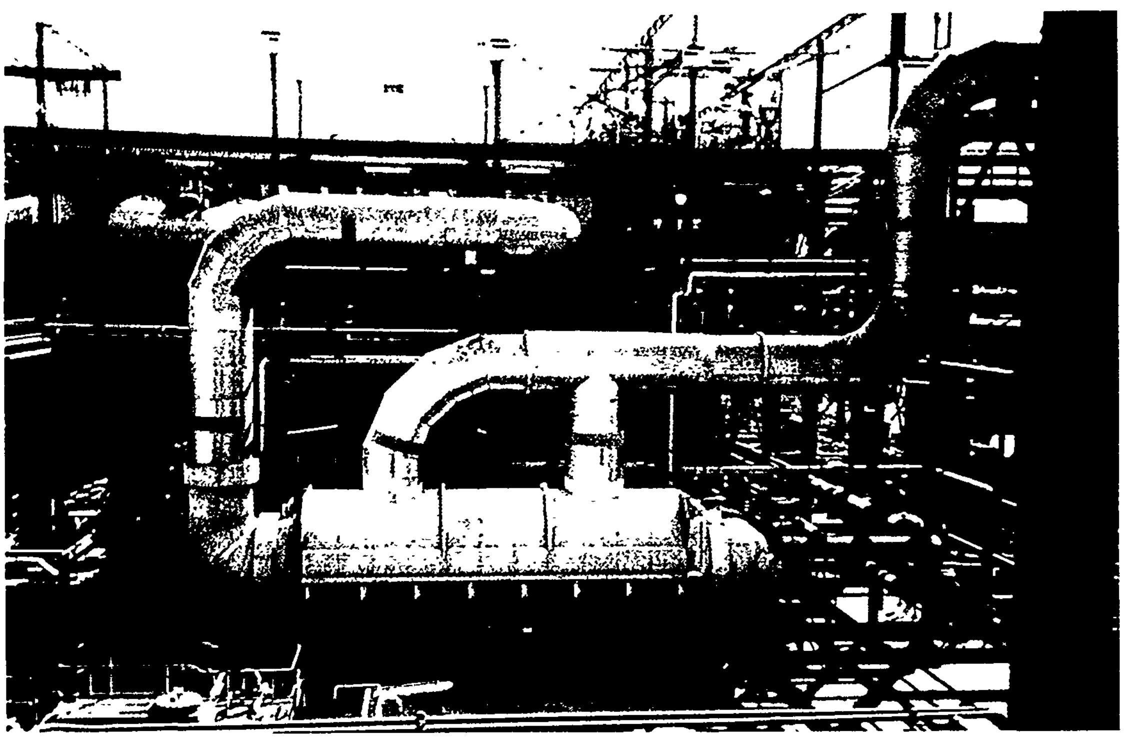
Figure 2-3: Operational plot

In fact, most of the time the plants are running on maximum load, the need for partial load is mainly due to limited storage and infrastructure. However, plants are frequently shut-down due to malfunctioning of pumps/valves, necessary maintenance etc., so much work could be done to make shut-down and start-up automated. Today this is done by experienced operators.

2.2 Choice of Subsystem for Control

There are about 20 control loops in an MSF plant (see Figure 2-1) ranging from chemical additive to temperature and level control loops. The main controlled variables are (El-Saie and Hafez, 1994):

- Top brine temperature, controlled by controlling the steam flow valve.
- Brine recycle flow, controlled by the brine recycle pump.
- Steam temperature, controlled by spraying cooling fluid on the steam.
- Make-up flow, controlled by the make-up flow pump.
- Brine level in last stage, controlled by the blow-down pump.
- Distillate level in last stage, controlled by the distillate pump.



Cooling brine enters to the left (green). Steam enters on top (red)

Picture 1: Brine heater

The plant represents a large-scale system, therefore we divide it into a number of subsystems which can be controlled independently. We have chosen the brine heater unit as the subsystem in this study. The two main reasons for this are:

1. It is the most important loop.

   Together with the brine recycling flow it is the main factor for the production rate. In (Al-Gobaisi *et al.*, 1994b) a study shows that it is the most critical loop for overall stability and economy of plant operation.

2. It is a non-linear system.

   The non-linearity relates to changing process characteristic (El-Saie and Hafez, 1994). Examples of this are: fouling of brine heater tubes which changes the overall heat transfer coefficient, change of recycling flow, variations in steam supply, valve characteristics and actuators. The brine heater process gain can change as much as 18% for different load conditions, and can have an 8% change between clean and fouled tube conditions (El-Saie and Hafez, 1994).

This makes conventional control, over a wide area of operation, quite difficult. The system, being a thermal system, is inherently stable. Due to large dimensions, it has a low bandwidth and is sluggish. Fluctuations of the TBT are the consequences of disturbances in the steam supply and recycle flow. The temperature of steam plays a minor role, as the steam is desuperheated. The brine inlet temperature is a function of the TBT, so it is not an external disturbance, but rather a crosscoupling.

2.3 Brine Heater Model

In (Babcock, 1994), we find a dynamic non-linear model of the brine heater from the MSF plant at Um-al-Nar-East 4-6 (UANE4-6). The brine heater is a steam-fluid, shell-tube heat-exchanger. (Babcock, 1994) use three sub-models to model the total system. The heater, valve and steam giving a system order of three. In this model they have not considered the distributed phenomena due to the large dimensions. To show the distributed phenomena better, we decided to extend the model by dividing the heater into 10 segments making it a 12th order system. The idea is taken from (Husain *et al.*, 1994), but they use the average (arithmetic), instead of the mean logarithmic, temperature difference.

We have the following assumptions:

- The density of brine in the tubes of the heater is constant. The density of the brine is a function of entering recycle brine temperature and salt concentration. As the brine concentration does not change insides the tubes the effect is negligible.
- Constant specific heat capacity of brine. Same as above.
- Heat loss to surroundings is negligible.
- Equal heating steam conditions on all tubes.
- Inlet temperature is independent of the brine outlet temperature. (Babcock, 1994) have not discussed this effect.
- Heat capacity of tubes is negligible.

2.3.1 The Heater

- Mass balance of brine in tubes:

$$\frac{dm_{Brine}}{dt} = \dot{m}_{Brine,\,in} - \dot{m}_{Brine,\,out} = 0 \qquad \text{(2-2)}$$

- Energy balance for one segment:

$$\frac{dU}{dt} = \rho_{Brine} \cdot c_p \cdot \tilde{V}_{Brine} \cdot \frac{dT_i}{dt} = \dot{Q}_{HS,i} + \dot{m}_{Brine} \cdot (T_{i-1} - T_i) \tag{2-3}$$

This is rearranged to give:

$$\frac{dT_i}{dt} = \frac{1}{\rho_{Brine} \cdot c_p \cdot \tilde{V}_{Brine}} \cdot \left(\dot{Q}_{HS,i} + \dot{m}_{Brine} \cdot (T_{i-1} - T_i)\right) \tag{2-4}$$

- Total brine volume in tubes:

$$V_{Brine} = \frac{\pi \cdot d^2}{4} \cdot L \cdot N_{Tubes} \tag{2-5}$$

Brine volume in one segment:

$$\tilde{V}_{Brine} = \frac{V_{Brine}}{n_{seg}} \tag{2-6}$$

- Total heat transfer area:

$$A = \pi \cdot d \cdot L \cdot N_{Tubes} \tag{2-7}$$

Heat transfer area for one segment:

$$\tilde{A} = \frac{A}{n_{seg}} \tag{2-8}$$

- Logarithmic mean temperature difference:

$$\Delta T_{LN,i} = \frac{T_i - T_{i-1}}{\ln\left(\frac{T_{cond} - T_{i-1}}{T_{cond} - T_i}\right)} \tag{2-9}$$

- Heat flow from steam to brine in one segment:

$$\dot{Q}_{HS,i} = k \cdot \tilde{A} \cdot \Delta T_{LN,i} \tag{2-10}$$

Total heat flow:

$$\dot{Q}_{HS} = \sum_{i=1}^{n_{seg}} \dot{Q}_{HS,i} \tag{2-11}$$

where $i=1...n_{seg}$ and $T_0=T_{Brine,in}$.

2.3.2 Steam

- Mass balance for steam in the heater shell:

$$\frac{dm_{HS}}{dt} = V_{HS} \cdot \frac{d\rho_{HS}}{dt} = \dot{m}_{HS} - \dot{m}_{cond} \tag{2-12}$$

rearranged to give steam density:

$$\frac{d\rho_{HS}}{dt} = \frac{(\dot{m}_{HS} - \dot{m}_{cond})}{V_{HS}} \tag{2-13}$$

- Energy balance of steam:

$$\frac{dU_{HS}}{dt} = \dot{Q}_{HS} - \dot{m}_{cond} \cdot \Delta h_V = 0 \tag{2-14}$$

giving the condensate flow:

$$\dot{m}_{cond} = \frac{\dot{Q}_{HS}}{\Delta h_V} \tag{2-15}$$

- Assuming ideal gas behavior of heating steam:

$$\rho_{HS} = \frac{p_{shell} \cdot M_{HS}}{R \cdot (T_{cond} + 273.15K)} \tag{2-16}$$

giving the shell pressure

$$p_{shell} = \rho_{HS} \cdot M_{HS} \cdot R \cdot (T_{cond} + 273.15K) \tag{2-17}$$

- For the condensation temperature the Antoine equation is used:

$$\log P = A - \left(\frac{B}{T} + C\right) \tag{2-18}$$

which is a semi-empirical relation that shows the correlation between vapour pressure and condensation temperature. Rearranging and with constants:

$$T_{cond} = \frac{-1750.268}{\log\left(\frac{p_{shell}}{1.332 \times 10^{-3}}\right) - 8.1077} - 235 \tag{2-19}$$

- Volume of the shell is approximately a constant times the volume of brine. The constant is determined from the brine heater's geometry:

$$V_{HS} = 4.07 \cdot V_{Brine} \tag{2-20}$$

2.3.3 The Valve

- Mass balance in valve:

$$\frac{dm_{HS}}{dt} = \dot{m}_{HS,\,in} - \dot{m}_{HS,\,out} = 0 \tag{2-21}$$

- Energy balance in valve:

$$\frac{dU_{HS}}{dt} = \dot{m}_{HS,\,in} \cdot h_{HS,\,in} - \dot{m}_{HS,\,out} \cdot h_{HS,\,out} = 0 \qquad (2\text{-}22)$$

- Steam flow characteristic:

$$\dot{m}_{HS} = k_{vs} \cdot c_g \sqrt{\frac{p_{sh} \cdot (p_{in} - p_{sh})\, \rho_s}{T_{HS}}} \cdot \varphi_{ep} \qquad (2\text{-}23)$$

- An equal percentage valve characteristic is used:

$$\varphi_{ep} = \frac{k_{vo}}{k_{vs}} \cdot e^{3.22 \cdot h_{in}} \qquad (2\text{-}24)$$

- The valve actuator dynamics is modelled with an on/off characteristic and dead-zone:

$$\frac{dh_{in}}{dt} = \begin{cases} 0.02, & \text{if } h_{cont} - h_{in} > \varepsilon \\ -0.02, & \text{if } h_{cont} - h_{in} < \varepsilon \\ 0, & \text{if } |h_{cont} - h_{in}| < \varepsilon \end{cases} \qquad (2\text{-}25)$$

2.3.4 Operating Conditions

As can be seen from the above non-linear model, the brine heater model has basically four disturbances. These are: the steam supply pressure, brine heater inlet temperature, brine recycle flow and the steam temperature to the brine heater.

From (Hornburg *et al.*, 1993) we find, in the contractors 's process design data, the range of variations for these variables, except for the brine heater inlet temperature. This is because the brine heater inlet temperature is dependant of the TBT, and is not an external disturbance. The brine heater is designed to deliver a TBT between 90°C and 106°C, depending on summer/winter and high/low operating conditions. The range of the disturbances are (Hornburg *et al.*,1993):

- Steam supply pressure
  - Maximum: 2.1 bar. Minimum: 1.4 bar.
- Brine recycle flow
  - Maximum: 4030 kg/s. Minimum: 3581 kg/s.
- Temperature of steam to brine heater
  - Maximum: 121°C. Minimum: 105°C.

The model should be validated against these figures, but this task lies outside the scope of this report.

The non-linear model can be implemented on any commercially available simulation environment, (Babcock, 1994) uses SPEEDUP[1]. We will implement the model in MATLAB/SIMULINK[2] in Chapter 5.

1. AspenTech, UK. 2. The MathWorks Inc., USA

3 Fuzzy Sets, Logic and Control

In this chapter we describe fuzzy sets, fuzzy logic and fuzzy control. Fuzzy logic has been successfully used in control engineering in the form of fuzzy logic controllers (FLC). The FLC of type Takagi-Sugeno is presented.

3.1 Fuzzy Sets And Logic

3.1.1 Fuzzy Sets

A classical set *C*, denoted as a "crisp set" to distinguish it from a fuzzy set, can be described by its characteristic function:

$$\mu_C: U \rightarrow \{0, 1\} \tag{3-1}$$

where *U* is the *universe of discourse*, i.e. a collection of objects which can be discrete or continuous. In a crisp set a variable either belongs or doesn't belong to the set. (Zadeh, 1965) extended this definition by introducing a fuzzy set *F* whose characteristic function is given by:

$$\mu_F : U \rightarrow [0,1] \tag{3-2}$$

thus giving a member of a fuzzy set all "truth" values between 0 and 1, and the possibility for a variable to belong to several fuzzy sets.

The characteristic function, or mapping function, is also called a membership function. This function can have any arbitrary shape, but the most common ones are piece-wise linear functions such as triangular, trapezoidal or singletons. These functions are easy to handle in a computer as storage and computations are simple.

3.1.2 Fuzzy Sets Operations

In order to manipulate with fuzzy sets we must have fuzzy set operations, like intersection, union and complement. In (Zadeh, 1965) we find the following definitions:

Union. The union of two fuzzy sets A and B with respective membership functions $\mu_A(x)$ and $\mu_B(x)$ is a fuzzy set C, written as $C=A \cup B$, whose membership function is related to those of A and B by:

$$\forall x \in X: \mu_C(x) = \max [\mu_A(x), \mu_B(x)] \tag{3-3}$$

This is also referred to as the *max*-operation.

Intersection. The intersection of two fuzzy sets A and B with respective membership functions $\mu_A(x)$ and $\mu_B(x)$ is a fuzzy set C, written as $C=A \cap B$, whose membership function is related to those of A and B by:

$$\forall x \in X\colon \mu_C(x) = \min\left[\mu_A(x),\, \mu_B(x)\right] \tag{3-4}$$

This is also referred to as the *min*-operation.

Complement. The complement of a fuzzy set A with membership function $\mu_A(x)$ is denoted as A and is defined as:

$$\forall x \in X\colon \mu_{\bar{A}}(x) = 1 - \mu_A(x) \tag{3-5}$$

These operations can be visualised as shown in Figure 3-1.

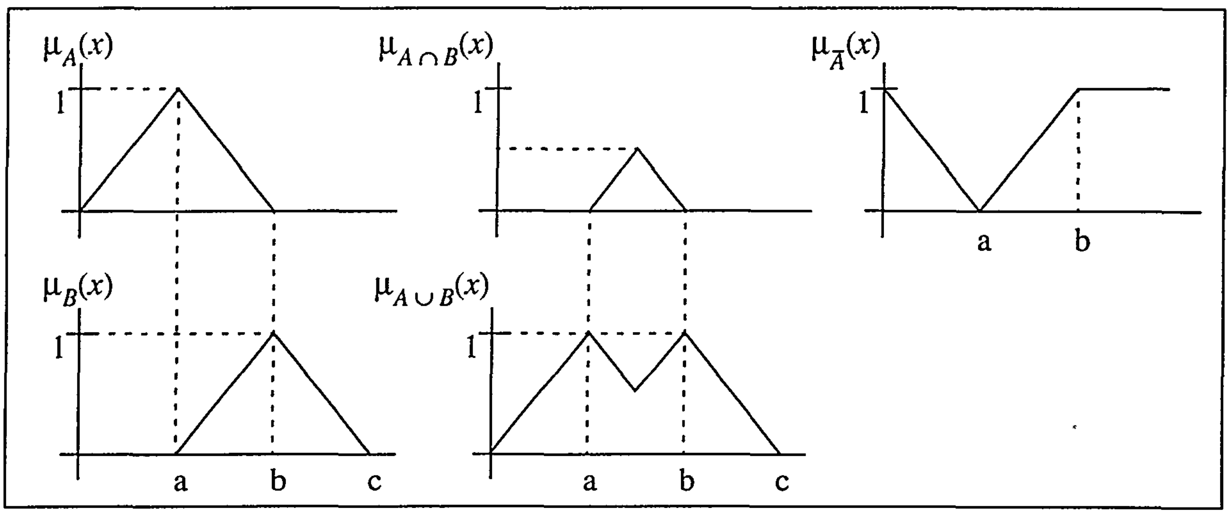


Figure 3-1: Example of fuzzy set operations

Most of the basic identities which holds for crisp sets also holds for fuzzy sets. This includes DeMorgan's laws, distributive laws and associative properties.

The following two laws in classical logic does, however, not hold for fuzzy logic:

- The law of contradiction:

$$A \cap \bar{A} \neq \varnothing \tag{3-6}$$

because:

$$\forall x \in X\colon \mu_{A \cap \bar{A}}(x) = \min\left[\mu_A(x), (1 - \mu_A(x))\right] \leq 1/2 \tag{3-7}$$

- The law of excluded middle:

$$A \cup \bar{A} \neq X \tag{3-8}$$

because:

$$\forall x \in X\colon \mu_{A \cup \bar{A}}(x) = \max\left[\mu_A(x), (1 - \mu_A(x))\right] \geq 1/2 \tag{3-9}$$

where $\varnothing$ denotes the empty set, and X is the whole universe of discourse.

3.1.3 Fuzzy Implication

Fuzzy sets can be labelled with appropriate names to describe linguistic variables, e.g. "hot", "comfortable" and "cold". Then by formulating fuzzy *if-then* rules, or just *fuzzy rules*, we can reason with linguistic terms. These rules show the input/output relation. A simple rule with one input and one output is:

$$\text{IF } x \text{ is } A \text{ THEN } y \text{ is } B \tag{3-10}$$

where x is the input, A and B are fuzzy sets and y is the output.

This can be expressed as:

$$\text{IF } x \text{ is } A \text{ THEN } y \text{ is } B \rightarrow (x, y) \text{ is } A \times B \tag{3-11}$$

where $\rightarrow$ stands for "translates into", and $A \times B$ is the Cartesian product of the fuzzy sets A and B. The membership function of $A \times B$ is given by:

$$\mu_{A \times B}(x, y) = \min[\mu_A(x), \mu_B(y)] \tag{3-12}$$

using the min-operator for intersection. This is also known as Mamdani's implication. Other implications exists, based on other operators than the min-operator. The reader is referred to (Lee, 1990; Driankov *et al.*, 1993) for further information.

3.1.4 Fuzzy Inference

In practice, more than one rule is useful. A set of fuzzy rules is usually built on the basis of the fuzzy partition of the input sets. We can then reason with these rules to obtain an output inferred from the inputs. When the rules are put together we get the *knowledge base* or *rule base*.

In general we have a rule base with n-rules on the form (single input, single output for simplicity):

$$\text{IF } x \text{ is } A_i \text{ THEN } y \text{ is } B_i \text{ , } i = 1, \ldots, n \tag{3-13}$$

This can be written as the combined implication:

$$\text{IF } x \text{ is } A_i \text{ THEN } y \text{ is } B_i \rightarrow (x, y) \text{ is } (A_1 \times B_1 \cup \ldots \cup A_n \times B_n) \text{ , } i = 1, \ldots, n \tag{3-14}$$

In short we can write:

$$(x, y) \text{ is } \bigcup_i A_i \times B_i \tag{3-15}$$

that is, a Cartesian mapping from the input space to the output space. This can also be interpreted as a fuzzy graph (Zadeh, 1995) as shown in Figure 3-2.

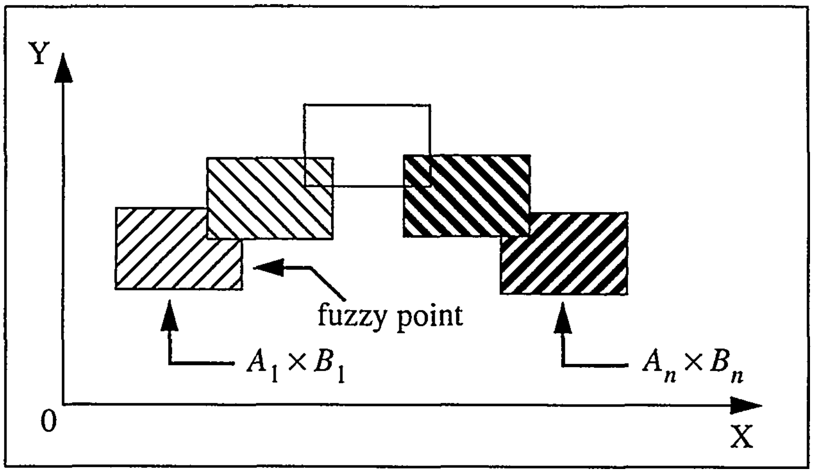


Figure 3-2: Interpretation of fuzzy rules as a fuzzy graph (Zadeh, 1995)

We then have an inference scheme, called *compositional rule of inference* (Zadeh, 1973):

$$\begin{array}{c} (x, y) \text{ is } R \\ x \text{ is } A \\ \hline y \text{ is } R \bullet A \end{array} \qquad \text{(3-16)}$$

The operation of composition is defined by:

$$\mu_{R \bullet A}(v) = \max_u(\min(\mu_R(u, v), \mu_A(u))) \qquad \text{(3-17)}$$

where $R = \sum_i A_i \times B_i$, $\mu_R(u, v)$ and $\mu_A(x)$ are the membership functions of R and A. Eq.(3-17) is also referred to as the *max-min* inference.

The operation of finding the linguistic value of y based on a linguistic value of x, may be interpreted as the projection of the intersection of the fuzzy graph with the cylindrical extension of A (Zadeh, 1995). This can be viewed graphically in Figure 3-3.

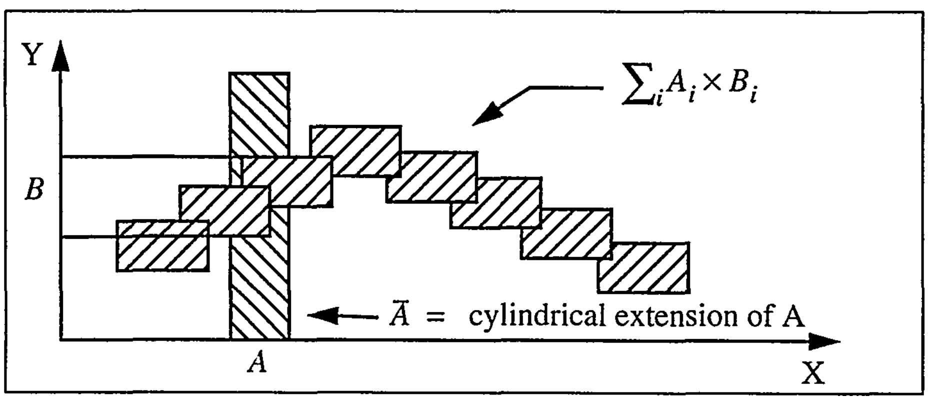


Figure 3-3: Interpolation of fuzzy graph (Zadeh, 1995)

3.1.5 Defuzzification

This process involves transferring the fuzzy output set given from the inference engine to a crisp value, e.g. the control signal to a valve. The most common method in control application is the Centre of Area (COA) method, which calculates the centre of area of a fuzzy set A on the universe X:

$$x^* = \frac{\sum_{x \in X} x \cdot \mu_A(x)}{\sum_{x \in X} \mu_A(x)} \qquad \text{(3-18)}$$

This defuzzification strategy gives a smooth output, which is preferred in control to avoid excessive stress on actuators.

Other methods like Mean Of Maxima (MOM) and First Of Maxima (FOM) defuzzification methods can be found in (Jamshidi *et al.*, 1993; Driankov *et al.*, 1993).

3.2 Fuzzy Control

Much of the success of FLC is due to its capability to integrate heuristic knowledge about a process into a controller. A PID is a very general controller, it will give the same linear signal no matter the amplitude of the inputs. With an FLC it is possible, in an easy way, to express the range of the inputs, thus making a specialised controller for each process. This fact, however, makes it a difficult task to express a general design method for FLC's.

3.2.1 Fuzzy Logic Controller

A typical FLC is shown in Figure 3-4

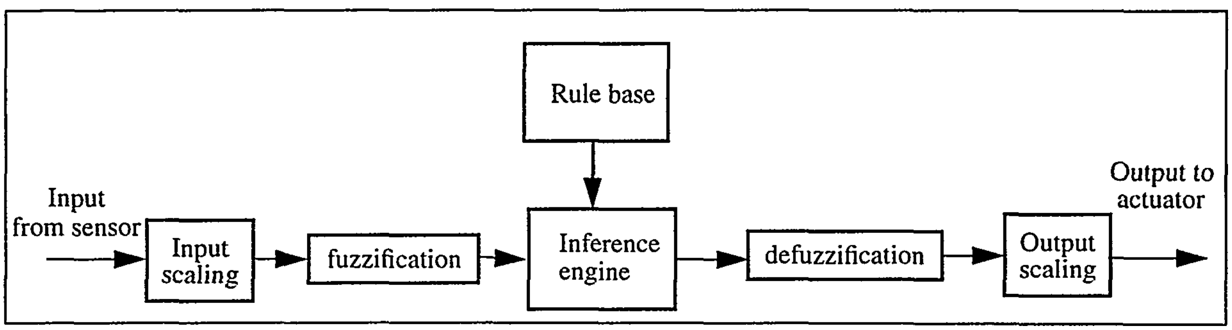


Figure 3-4: Block diagram of typical FLC

The FLC consists of the following computational steps:.

- *Input scaling*

  This consists of mapping the sensor value to a normalised universe of discourse, which is usually the universe [-*1*,*1*]. This is to facilitate computations.

- *Fuzzification*

  This is the process of mapping a sensor value x_0 onto a fuzzy set with the membership function $\mu(x)$ to obtain $\mu(x_o)$. This makes the input sensor values compatible with the fuzzy rules in the rule base.

- *Inference engine*

  This means combining the fuzzified inputs to obtain the fuzzy output set. This is done through the rule base which expresses the relation between the inputs and outputs, in the form of fuzzy rules. The fuzzy rules are then inferred with an inference mechanism, usually the max-min inference, creating a fuzzy output set.

- *Defuzzification*

  This is the process of obtaining the crisp (numerical) value from the inferred fuzzy output set. As mentioned before, the most common one used is the COA.

- *Output scaling*

  This is the process of transferring the normalised crisp output to the physical universe of discourse.

Of these steps, the input and output scaling are optional. An FLC can be seen upon as a static non-linear transfer element as no operations on time are performed inside the FLC (Driankov, *et al.*, 1993). The non-linearities comes from the fuzzification/defuzzification steps (due to the non-linear membership functions), and the inference engine (the rule base and due to the inference mechanism like the max-min inference).

3.2.2 Takagi-Sugeno Fuzzy Logic Controller

The Takagi-Sugeno FLC (TSFLC) uses fuzzy sets in the antecedents, but the output consequence is, in general, a crisp non-linear function of the input variables or can be a constant value (Takagi and Sugeno, 1985). The fuzzy rules of a TSFLC are of the form (with single output for simplicity):

$$\begin{array}{l} R^{(1)}: \text{if } x_1 \text{ is } A_1^{(1)} \text{ and ... and } x_n \text{ is } A_n^{(1)} \text{ then } u_1 = f_1(x_1, x_2, ..., x_n) \\ \vdots \qquad\qquad\qquad\qquad\qquad \vdots \\ R^{(m)}: \text{if } x_1 \text{ is } A_1^{(m)} \text{ and ... and } x_n \text{ is } A_n^{(m)} \text{ then } u_m = f_m(x_1, x_2, ..., x_n) \end{array} \tag{3-19}$$

where A_i^j are fuzzy sets, x is the input vector, u is the output and f is, in general, a non-linear function.

The inputs of the TSFLC can be a state vector or the output of the system (error, change of error, integral of error etc.). One can, with this method, assign a control function for each region of the input-space as a single rule. This makes it possible to weaken or amplify certain frequencies according to the inputs. This can also be done by conventional switching of controllers, but the fuzzy method offers a smooth transition between the individual controllers.

The total output of Eq.(3-19) is the weighted combination of all the fired rules:

$$u = \frac{\sum_{i=1}^{m} \mu_{u_i} \cdot u_i}{\sum_{i=1}^{m} \mu_{u_i}} \tag{3-20}$$

where μ_{u_i} is the firing strength of the i-th rule found by the min-operation for fuzzy implication (Eq.(3-12)). As the rule consequence is a non-normalised function, the need for output scaling can be omitted. Furthermore, explicit defuzzification is needless, since the result of each rule is a crisp value and the global result is determined by Eq.(3-20).

3.2.3 Fuzzy Logic Controller Design

Unlike linear control theory, no formal method for designing FLC's exists. FLC's are, in most cases, designed independent of the system's model, so the design mainly relies on experience and knowledge of the process. FLC's are sometimes referred to as *knowledge-based* controllers. This is a subjective method, and is one of the reasons for the controversy over FLC in the control community. FLC's are most appropriate when the problem at hand is too intractable and a mathematical model or representation is not even available (Jamshidi, 1996).

The relevant issues in FLC design[1] are (Lee, 1990; Driankov *et al.*, 1993; Kruse *et al.*, 1994):

1. Definition of input and output variables.
   The essential parameters reflecting the process state should be chosen as inputs. In the case of a fuzzy PID like controller, the inputs are error, change of error or sum of errors. The output variables are the outputs to the actuators, normally one has only one output from the FLC. This can either be the actual control signal, u, or the change in control, $\dot{u}$.
2. Fuzzy partition of the input and output domains.

   Fuzzy sets with linguistic labels are partitioned on the input and output domains. In the case of a TSFLC, the crisp output functions must be determined. The domains can either be normalised, or be the actual physical domain. The domain is, in any case, determined by knowledge of the process, such as physical limits. The number of fuzzy sets determines the granularity of the control. The shape of the membership functions is, as before mentioned, normally chosen to be linear piece-wise functions.
   The number of possible rule combinations increases exponentionally with the number of input sets, and linearly with the number of output sets, so there is a practical limitation on the fuzzy partitioning. This can partly be solved by using a hierarchical control structure, where inputs are combined together before entering the FLC (Jamshidi, 1996). This will be discussed in Chapter 6 'Hierarchy and Hybridization of Fuzzy Logic Control Systems' on page 44.
3. Rule base.

   The rule base consists of a set of fuzzy rules based on expert knowledge. How to derive this rule base is not a straight forward procedure, and no mature guidance exists for this problem. In (Driankov *et al.*, 1993; Lee, 1990) we find three approaches:

   - Derivation of rules based on knowledge from a process operator and/or control engineer. This is either done by an *introspective* verbalization of human expertise, or a well formulated questionnaire.
   - The use of a fuzzy model of the process to directly derive the rule base.
   - Self organizing/learning controllers. The FLC has a supervisory rule base which consists of meta-knowledge which enables the FLC to evaluate its performance. Based on this evaluation, the FLC can derive and modify the rule base.
4. Scaling factors.

   Input scaling factors determines the sensitivity for the input variables. The output

1. we assume that the need for using an FLC has been justified

scaling factor determines the output energy intensity (this can be looked upon as the controller's gain).

5. Defuzzification method.

   Most often this is chosen to be the COG, which produces a smooth control surface. In the case of the TSFLC this step is, as mentioned before, omitted.

6. Implementation and simulation of the fuzzy control system.

   There exists a number of software packages[1] for implementation and design of FLC's, and which allows simulation with a plant model (Jamshidi *et al.*, 1993).This, however, requires a mathematical model of the plant, which is not always available.

7. Tuning and optimization.

   Usually an iterative design process is necessary. Few methodologies exists as where and how to tune the FLC, but is rather a result of trial and error. The tuning can be done on the scaling factors, the shape of the fuzzy sets, number of fuzzy sets, and the rule base. This gives a very high number of tuning parameters. A natural mix seems to connect the FLC with other Artificial Intelligence mechanisms, such as Neural Networks and Genetic Algorithms, to obtain an optimised controller.

8. Hardware implementation.

   If the FLC works satisfactorily, a real-time implementation to the plant can be done with one of the increasingly numbers of fuzzy micro-controllers. An overview of available hardware can be found in (Jamshidi *et al.*, 1993).

Other design issues, such as completeness and consistency of the rule base, will not be discussed here. For a more in-depth overview of the design subject, the reader is referred to (Driankov *et al.*, 1993).

1. The MathWorks, Inc. have now released their Fuzzy Control Toolbox

4 Stability of Fuzzy Logic Control Systems

4.1 Introduction

One of the most fundamental issues in any control system-fuzzy or others is the stability. Briefly, a system is said to be stable if it would come to its equilibrium state after any external inputs, initial conditions and/or disturbances have impressed the system. The issue of stability is of even greater relevance when questions of safety, lines, and environment are at state like in such systems as nuclear reactors, traffic systems and aeroplanes autopilots. The stability test of fuzzy control system or lack of it has been a subject of criticism by many control engineers in some control engineering literature (IEEE Control Systems Magazine, 1993).

Almost any linear or non-linear system under the influence of a closed-loop crisp controller has one type of stability test or other. For example, the stability of a linear time-in variant system can be tested by a wide variety of methods such as Routh-Hurwitz, Root locus, Bode plots, Nyquist criterion, and even through traditionally non-linear systems methods of Lyapunov, Popov and Circle criterion. The common requirement in all these tests is the availability of a mathematical model-be it in time or frequency domain. A reliable mathematical model for a very complex and large-scale system may, in practice, be unavailable or unfeasible. In such cases, a fuzzy controller may be designed based on expert knowledge or experimental practice. However, the issue of the stability of a fuzzy control system still remains and must be addressed. The aim of this chapter is to present an up-to-date survey of available techniques and tests for fuzzy control systems stability.

Fuzzy controllers represent static non-linearities (Bretthauer and Optiz, 1994) and as such the stability problems belongs to non-linear control systems. In this chapter, a survey of fuzzy control systems stability will be given and a few more promising approaches will be described in more detail.

4.2 Fuzzy Control Systems Stability Classes

From the viewpoint of stability a fuzzy controller can be either acting as a conventional (low-level) controller or as a supervisory (high-level) controller (Jamshidi, 1996). Depending on the existence and nature of a system's mathematical model and the level in which fuzzy rules are being utilized for control and robustness, four classes or fuzzy control suitability problems can be distinguished. These four classes are:

Class 1: Process model is crisp and linear and fuzzy controller is low level.

Class 2: Process model is crisp and non-linear and the fuzzy controller is low level.

Class 3: Process model (linear or non-linear) is crisp and a fuzzy tuner or an adaptive fuzzy controller is present at high level.

Class 4: Process model is fuzzy and fuzzy controller is low level.

Figures 4.1-4.4 show all the four classes of fuzzy control systems whose stability is of concern. In this presentation we are concerned mainly with the first three classes. For the last class, traditional non-linear control theory would fail and is beyond the scope of this section. It will be discussed very briefly. The techniques for testing the stability of the first two classes of systems (Figures 4.1 and 4.2) are shown in Table 4.1. As shown, the methods are divided into two main groups-time and frequency.

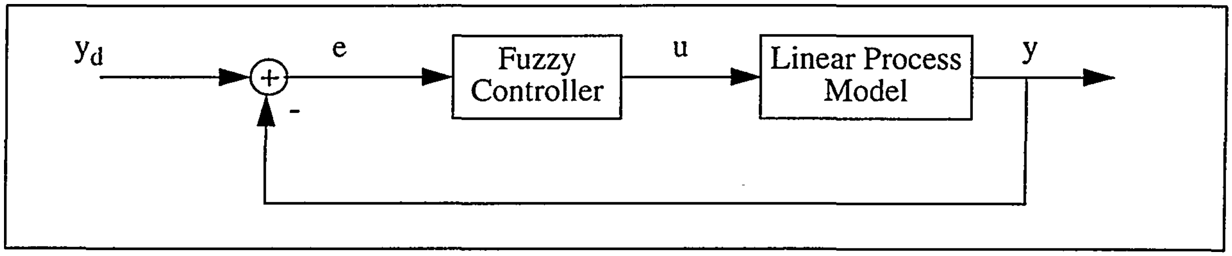


Figure 4-1: Class 1 of fuzzy control system stability problem.

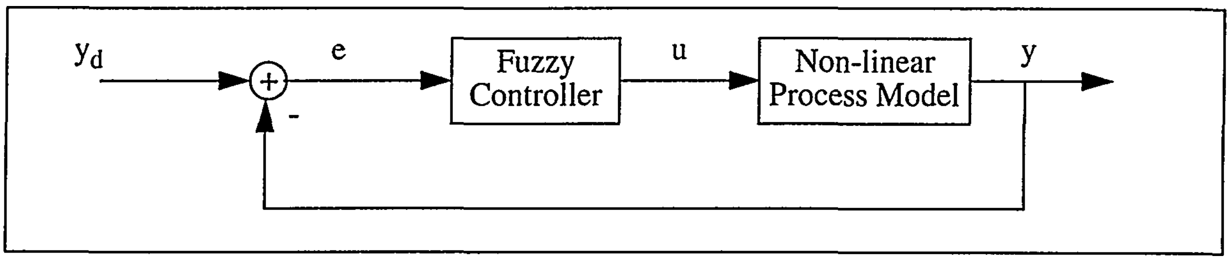


Figure 4-2: Class 2 of fuzzy control system stability problem.

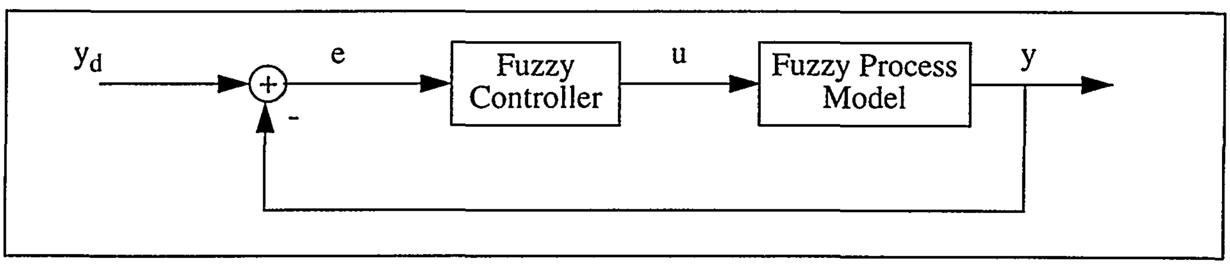


Figure 4-3: Class 3 of fuzzy control system stability problem.

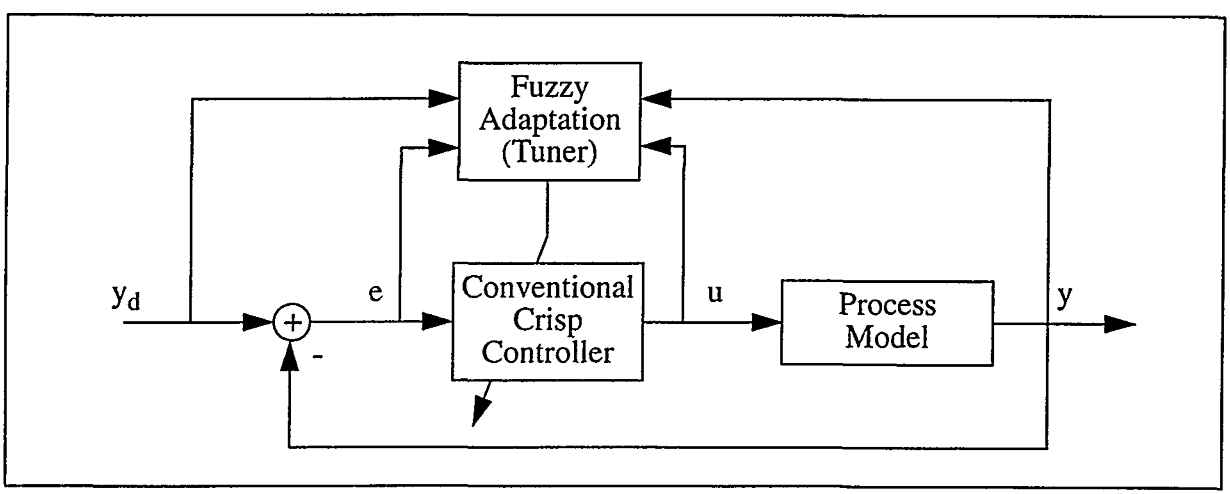


Figure 4-4: Class 4 of fuzzy control system stability problem.

Table 4-1: Stability Analysis Methods for Fuzzy Control Systems with Known Model

Stability Analysis Methods

Time Domain	Frequency Domain
State-space	Harmonic Balance (Describing Function)
Lyapunov Theory	
Hyperstability Theory	Popov Criterion
Bifurcation Theory	
Graph Theory	Circle Criterion

4.2.1 Time Domain Methods

The state-space approach, considered by many authors (Aracil, *et al.*, 1988; Chen, 1988; Chen and Tsao, 1989; Wang, *et al.*, 1990; Hojo, *et al.*, 1991; Hwang and Liu, 1992; Driankov, *et al.*, 1993; Kang, 1993; Demaya, *et al.*, 1994). The basic approach here is to subdivide the state space into a finite number of cells based on the definitions of the membership functions. Now, if a separate rule is defined for every cell, a cell-to-cell trajectory can be constructed from the system's output induced by the new outputs of the fuzzy controller. If every cell of the modified state space is checked, one can identify all the equilibrium points including the system's stable region. This method should be used with some care since the inaccuracies in the modified description could cause oscillatory phenomenon around the equilibrium points.

The second class of methods is based on the Lyapunov's method. Several authors (Aracil, *et al.*, 1989; Langari and Tomizuka, 1990; Bouslama and Ichikawa, 1992; Chen, *et al.*, 1993; Chen, 1987; Driankov, *et al.*, 1993; Franke, 1993; Gertler and Chang, 1986; Hoja, *et al.*, 1991; Kiszka, *et al.*, 1985; Tanaka and Sugeno, 1992; Wang, 1993, 1994; Tahani and Sheikholeslam, 1994) have used this theory to come up with criterion for stability of fuzzy control systems. The approach is along the same lines as in conventional stability approaches, i.e. show that the time derivative of the Lyapunov function at the equilibrium point is negative semi definite. Many approaches have been proposed. One approach is to define a Lyapunov function and then derive the fuzzy controller's architecture out of the stability conditions. Another approach uses Aiserman's method (Bretthauer and Optiz, 1994) to find an adopted Lyapunov function, while representing the fuzzy controller by a non-linear algebraic function $u = f(y)$, when y is the system's output. A third method calls for the use of so-called *facet functions*, where the fuzzy controller is realized by box-wise multi-linear facet functions with the system being described by a state space model. To test stability, a numerical parameter optimization scheme is needed.

Hyperstability approach, considered by other authors (Barreiro and Aracil, 1992; Optiz, 1993, 1994) has been used to check stability of systems depicted in Figure 4.1. The basic approach her is to restrict the input-output behaviour of the non-linear fuzzy controller by inequality and to derive conditions for the linear part of the closed-loop system to be satisfied for stability.

Bifurcation theory (Driankov, *et al.*, 1993) can be used to check stability of fuzzy control systems of class described in Figure 4.2. This approach represents a tool in deriving stability conditions and robustness indices for stability from small gain theory. The fuzzy controller, in this case, is described by a non-linear vector function. The stability, in this scheme, could only be lost if one of the following conditions become true: (i) the origin becomes unstable if a pole crosses the imaginary axis into the right-half plane-static bifurcation, (ii) the origin becomes unstable if a pain of poles would cross over the imaginary axis and assumes positive real parts-Hopf bifurcation, or (iii) new additional equilibrium points are produced.

The last time-domain method is the use of *graph theory* (Driankov, *et al.*, 1993). In this approach conditions for special non-linearities are derived to test the BIBO stability.

4.2.2 Frequency Domain Methods

There are three primary groups of methods which have been considered her (see Table 4.1). The *Harmonic balance* approach, considered by Braee and Rutherford (1978, 1979), and Kickert and Mamdani (1978), among others, has been used to check the stability of the first two classes of fuzzy control systems (see Figures 4.1 and 4.2). The main idea is to check if permanent oscillations occur in the system and whether these oscillations with known amplitude or frequency are stable. The non-linearity (fuzzy controller) is described by a complex-valued describing function and the condition of Harmonic balance is tested. If this condition is satisfied, then a permanent oscillation exists. This approach is equally applicable to MIMO systems.

Circle criterion (Aracil, *et al.*, 1989, 1991; Optiz, 1994; Ray and Majumder, 1984; Ray, *et al.*, 1984) and Popov Criterion (Böhm, 1992; Buhler, 1993) have been used to check stability of the first class of systems (Figure 4.1). In both criteria, certain conditions on the linear process model and static non-linearity (controller) must be satisfied. It is assumed that the characteristic value of the non-linearity remains within certain bounds, and the linear process model must be open-loop stable with proper transfer function. Both criteria can be graphically evaluated in simple manners.

The stability of adaptive fuzzy control systems has been treated in details by Wang (1994a, 1994b) and is best used when augments the design process as is described in (Jamshidi, 1996).

4.3 Lyapunov Stability of Fuzzy Control Systems

As mentioned before, one of the most fundamental criterion of any control system is to insure stability as a part of the design process. In this section, some theoretical results on this important topic is detailed, followed by a few stability-guaranteeing design examples.

We begin with the ith implication of a fuzzy system:

$$\begin{aligned} P^i:\ & \text{IF } x(k) \text{ is } A_1^i \text{ and...and } x(k-n+1) \text{ is } A_n^i \\ & \text{THEN } x^i(k+1) = a_1^i x(k) + \dots + a_n^i x(k-n+1) \end{aligned} \tag{4-1}$$

with $i = 1,..., l$. The present discussion is limited to autonomous fuzzy control systems. The stability of a fuzzy control system with presence of the inputs will be considered shortly. The consequent part of (4-1) represents a set of linear subsystems can be rewritten as (Tanaka and Sugeno, 1992), i.e.

$$\text{IF } x(k) \text{ is } A_1^i \text{ and...and } x(k-n+1) \text{ is } A_n^i \qquad \text{(4-2)}$$
$$\text{THEN } x^i(k+1) = A_i x(k)$$

where $x(k)$ is defined by $\boldsymbol{x}(k) \equiv [x(k)\ x(k-1)\ \dots\ x(k-n+1)]^T$ and $n \times n$ matrix $\boldsymbol{A}_i$ is:

$$\boldsymbol{A}_i = \begin{bmatrix} a_1^i & a_2^i & \dots & a_{n-1}^i & a_n^i \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ : & : & \dots & : & : \\ 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} \qquad \text{(4-3)}$$

The output of the fuzzy stem described by Equations (4-1) - (4-3) is given by

$$x(k+1) = \frac{\sum_{i=1}^{l} w^i \boldsymbol{A}_i \boldsymbol{x}(k)}{\sum_{i=1}^{l} w^i} \qquad \text{(4-4)}$$

where w^i is the overall truth value of the ith implication (Jamshidi, 1996) and l is the total number of implications. In sequel, a theorem and a lemma for crisp discrete-time systems is first given before a Lyapunov method for the problem is presented.

Theorem 4.1 Consider a discrete-time system described by

$$\boldsymbol{x}(k+1) = f(\boldsymbol{x}(k)) \qquad \text{(4-5)}$$

where $\boldsymbol{x}(k) \in \Re^n$, $f(\cdot)$ is an n-dimensional non-linear function with a property $f(0) = 0$ for all k. Assume that a scalar continuous function $v(\boldsymbol{x}(k))$ exists which has the following properties

a) $v(0) = 0$

b) $v(\boldsymbol{x}(k)) > 0 \qquad$ for $\boldsymbol{x}(k) \neq 0$

c) $\lim_{\|x(k)\| \to \infty} v(\boldsymbol{x}(k)) = \infty$

d) $\Delta v(\boldsymbol{x}(k)) = (v(k+1) - v(k)) < 0 \qquad$ for $\boldsymbol{x}(k) \neq 0$

Then the equilibrium point $\boldsymbol{x}(k) = 0$ of (4-5), obtained by solving $\boldsymbol{x}(k) = f(\boldsymbol{x}(k))$, is asymptotically stable in the large for all k and $v(\boldsymbol{x}(k))$ is a Lyapunov function.

Clearly, this theorem is the discrete version of the standard Lyapunov stability theorem (Jamshidi, *et al.*, 1992). Now, consider the following Lemma:

Lemma 4.1 If $\boldsymbol{P}$ is a positive definite matrix such that

$$\boldsymbol{A}^T\boldsymbol{P}\boldsymbol{A} - \boldsymbol{P} < 0 \quad \text{and} \quad \boldsymbol{B}^T\boldsymbol{P}\boldsymbol{B} - \boldsymbol{P} < 0$$

where $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{P}$ are $n \times n$ matrices, then

$$\boldsymbol{A}^T\boldsymbol{P}\boldsymbol{B} + \boldsymbol{B}^T\boldsymbol{P}\boldsymbol{A} - 2\boldsymbol{P} < 0$$

Proof: Evaluating the left side of the inequality would result in

$$A^TPB + B^TPA - 2P = -(A-B)^TP(A-B) + A^TPA + B^TPB - 2P$$
$$= -(A-B)^TP(A-B) + (A^TPA - P) + (B^TPB - P)$$

since the last two terms are assumed to be positive definite and since P is a positive definite matrix,

$$-(A-B)^TP(A-B) \le 0$$

and the lemma's conclusion follows. ❑

We now present the first stability result of fuzzy control systems, due to Tanaka and Sugeno (1992), among others.

Theorem 4.2 The equilibrium point of a fuzzy system (4-4) is globally asymptotically stable if there exists a common positive definite matrix P for all subsystems such that

$$A_i^TPA_i - P < 0 \tag{4-6}$$

for i=1,...,l.

Proof: Let the scalar Lyapunov function $v(x(k)) = x^T(k)Px(k)$, where P is an $n \times n$ positive definite matrix. The function $v(x(k))$ satisfies the following properties: (i) $v(0) = 0$, (ii) $v(x(k)) > 0$ for $x(k) \neq 0$, and (iii) $\lim_{\|x(k)\| \to \infty} v(x(k)) = \infty$.

Next, we evaluate

$$\Delta v(x(k)) = v(x(k+1)) - v(x(k)) = x^T(k+1)Px(k+1) - x^T(k)Px(k) \tag{4-7}$$

$$= \left(\frac{\sum_{i=1}^{l} w^i A_i x(k)}{\sum_{i=1}^{l} w^i}\right)^T P \left(\frac{\sum_{i=1}^{l} w^i A_i x(k)}{\sum_{i=1}^{l} w^i}\right) - x^T(k)Px(k)$$

$$= x^T(k)\left[\left(\frac{\sum_{i=1}^{l} w^i A_i^T x(k)}{\sum_{i=1}^{l} w^i}\right)^T P \left(\frac{\sum_{i=1}^{l} w^i A_i x(k)}{\sum_{i=1}^{l} w^i}\right) - P\right]x(k)$$

$$= \frac{\sum_{i,j=1}^{l} w^i w^j x^T(k)(A_i^TPA_j - P)x(k)}{\sum_{i,j=1}^{l} w^i w^j}$$

$$= \frac{\left[\sum_{i=1}^{l} (w^i)^2 x^T(k)(A_i^TPA_i - P)x(k) + \sum_{i<j}^{l} w^i w^j x^T(k)Px(k)\right]}{\sum_{i,j=1}^{l} w^i w^j}$$

where $w^i \geq 0$ for all i=1,...,l and $\sum_{i=1}^{l} w^i > 0$. Now, by assumption of the theorem, Equation (4-6), the first matrix is negative definite and the second matrix relation is negative definite by Lemma 4.1. Hence, we have $\Delta v(x(k)) < 0$. Now, in lieu of Theorem 4.1, the fuzzy system (4-4) is globally asymptotically stable. ❑

It is noted that the above theorem can be applied to any non-linear system which can be approximated by a piece-wise linear function if the stability condition (4-6) is satisfied. Moreover, if there exists a common positive definite matrix $\boldsymbol{P}$, then all the $\boldsymbol{A}_i$ matrices are stable. Since Theorem 4.2 is a sufficient condition for stability, it is possible not to find a $\boldsymbol{P} > 0$ even if all $\boldsymbol{A}_i$ matrices are stable. In other words, a fuzzy system may be globally asymptotically stable even if a $\boldsymbol{P} > 0$ is not found. The fuzzy system is not always stable even if all the $\boldsymbol{A}_i$'s are stable.

Theorem 4.3 Let $\boldsymbol{A}_i$ be stable and non-singular matrices for i = 1,...,l. Then $\boldsymbol{A}_i\boldsymbol{A}_j$ are stable matrices for i, j = 1,...,l, if there exists a common positive definite matrix $\boldsymbol{P}$ such that

$$\boldsymbol{A}_i^T\boldsymbol{P}\boldsymbol{A}_i - \boldsymbol{P} < 0 \tag{4-8}$$

Proof: Rewriting (4-8), we obtain

$$\boldsymbol{P} - (\boldsymbol{A}_i^{-1})^T\boldsymbol{P}\boldsymbol{A}_i^{-1} < 0$$

since $(\boldsymbol{A}_i^{-1})^T = (\boldsymbol{A}_i^T)^{-1}$. Thus, $\boldsymbol{P} < (\boldsymbol{A}_i^{-1})^T\boldsymbol{P}\boldsymbol{A}_i^{-1}$ for i = 1,...,l. In view of this inequality and the one in (4-8), it follows that

$$\boldsymbol{A}_i^T\boldsymbol{P}\boldsymbol{A}_i < (\boldsymbol{A}_i^{-1})^T\boldsymbol{P}(\boldsymbol{A}_i^{-1})$$

or by pre- and post-multiplying by $\boldsymbol{A}_j^T$ and $\boldsymbol{A}_j$, respectively we have $\boldsymbol{A}_j^T\boldsymbol{A}_i^T\boldsymbol{P}\boldsymbol{A}_i\boldsymbol{A}_j - \boldsymbol{P} < 0$. Thus, must be a Hurwitz matrix for i, j = 1...,l. ❑

In summary, to check the stability of a fuzzy system, one must find a common positive definite $\boldsymbol{P}$.This task may not be an easy task. One possible algorithm to check the stability is to

1. Find a $\boldsymbol{P}_i > 0$ such that $\boldsymbol{A}_i^T\boldsymbol{P}\boldsymbol{A}_i - \boldsymbol{P} < 0$ for i = 1,...,l. If $\boldsymbol{A}_i$ is stable, it is always possible to find a $\boldsymbol{P}_i$.
2. Check if a $\boldsymbol{P}_j \in \{\boldsymbol{P}_i \mid i\text{=1,...,}l\}$ exists such that $\boldsymbol{A}_i^T\boldsymbol{P}_j\underline{\boldsymbol{A}}_i - \boldsymbol{P}_j < 0$.

   If so, a common $\boldsymbol{P}$ has been obtained. Otherwise go to step (1) for the next value of i.

Stability of Non-autonomous Fuzzy Control Systems

Thusfar, the criteria which have been presented treat autonomous (no inputs) systems. Consider the following non-autonomous fuzzy system

$$\begin{aligned} p^i\text{: IF } x(k) \text{ is } A_1^i \text{ and...and } x(k-n+1) \text{ is } A_n^i \\ \text{and } u(k) \text{ is } B_1^i \text{ and...and } u(k-m+1) \text{ is } B_m^i \\ \text{THEN } x^i(k+1) = a_0^i + a_1^i x(k) + \ldots + a_n^i x(k-n+1) \\ + b_1^i u(k) + \ldots + b_m^i u(k-m+1) \end{aligned} \tag{4-9}$$

In this section, we use some results from Tahani and Sheikholeslam (1994) to test the stability of the above system. We begin by a definition.

Definition 4.1 The non-linear system

$$x(k+1) = f(x(k), u(k), k), \qquad y = g(x(k), u(k), k)$$

is totally stable if and only if for any bounded input $\boldsymbol{u}(k)$ and bounded initial state $\boldsymbol{x}_0$, the state $\boldsymbol{x}(k)$ and the output $\boldsymbol{y}(k)$ of the system is bounded, i.e., we have

$$\text{For all } \|\boldsymbol{x}_0\| < 0 \text{ and for all } \|\boldsymbol{u}\| < \infty \Rightarrow \|\boldsymbol{x}(k)\| < \infty \text{ and } \|\boldsymbol{y}(k)\| < \infty \tag{4-10}$$

Here, $\boldsymbol{u}(k) \equiv \left[u(k)\ u(k-1)\ \dots\ u(k-n+1)\right]^T$.

Now, consider the following theorem:

Theorem 4.4 The fuzzy system (4-9) is totally stable if there exists a common positive definite matrix $\boldsymbol{P}$ such that the following inequalities

$$\boldsymbol{A}_i^T \boldsymbol{P} \boldsymbol{A}_i - \boldsymbol{P} < 0 \tag{4-11}$$

for $i = 1,\dots,l$, and $\boldsymbol{A}_i$ is defined by (4-3). The proof of this theorem can be found in Kholeslam, (1994). Note that the conditions (4-11) is the same as those in (4-6) of Theorem 4.2 for fuzzy system (4-1). Proof of this theorem can be found in Sheikholeslam (1994).

The next section presents an alternative (non-Lyapunov) criterion for fuzzy control systems.

4.4 Fuzzy System Stability via Interval Matrix Method

Recent results on the stability of time-varying discrete interval matrices by Han and Lee (1994) can lead us to some more conservative, but computationally more convenient, stability criteria for fuzzy systems of the Takagi-Sugeno type shown by Equation (4-1). Before we can state these new criteria some preliminary discussions will be necessary. Consider a linear discrete-time system described by a difference equation in state form:

$$\boldsymbol{x}(k+1) = (\boldsymbol{A} + \boldsymbol{G}(k))\,\boldsymbol{x}(k), \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{4-12}$$

where $\boldsymbol{A}$ is $n \times n$ constant asymptotically stable matrix, $\boldsymbol{x}$ is the $n \times 1$ state vector, and $\boldsymbol{G}(k)$ is an unknown $n \times n$ time-varying on the perturbation matrix's maximum modulus, i.e.

$$|\boldsymbol{G}(k)| \le \boldsymbol{G}_m, \quad \text{for all } k \tag{4-13}$$

where and the inequality holds elementwise. Now, consider the following theorem:

Theorem 4.5 The time-varying discrete-time system (4-12) is asymptotically stable if

$$\rho(|\boldsymbol{A}| + \boldsymbol{G}_m) < 1 \tag{4-14}$$

where $\rho(\cdot)$ stands for maximum (or largest) of the eigenvalues. The proof of this theorem is straightforward, based on the evaluation of the spectral norm $\|\boldsymbol{x}(k)\|$ of $\boldsymbol{x}(k)$ and showing that if condition (4-14), then $\lim_{k \to \infty} \|\boldsymbol{x}(k)\| = 0$. The entire proof can be found in Han and Lee, (1994).

Definition 4.2 An *interval matrix* $A_I(k)$ is an $n \times n$ matrix whose elements consist of intervals $[b_{ij}, c_{ij}]$ for $i, j = 1,..., n$, i.e.

$$A_I(k) = \begin{bmatrix} [b_{11}, c_{11}] & \dots & [b_{1n}, c_{1n}] \\ : & & : \\ & [b_{ij}, c_{ij}] & \dots \\ [b_{n1}, c_{n1}] & \dots & [b_{nn}, c_{nn}] \end{bmatrix} \tag{4-15}$$

Definition 4.3 The *centre matrix*, A_c, and the *maximum difference matrix*, A_m of $A_I(k)$, in (4-15) are defined by

$$A_c = \frac{B+C}{2}, \quad A_m = \frac{C-B}{2} \tag{4-16}$$

where $B = \{b_{ij}\}$ and $C = \{c_{ij}\}$. Thus, the interval matrix $A_I(k)$ in (4-15) can also be rewritten as

$$A_I(k) = [A_c - A_m, A_c + A_m] = A_c + \Delta A(k)\,\mathrm{p} \tag{4-17}$$

with $|\Delta A(k)| < A_m$.

Lemma 4.2 The interval matrix $A_I(k)$ is asymptotically stable if matrix A_c is stable and

$$\rho(|A_c| + A_m) < 1 \tag{4-18}$$

or in canonical form,

$$\rho(|T^{-1}A_cT| + |T^{-1}|A_m|T|) < 1 \tag{4-19}$$

The proof can be found in Han and Lee (1994). The above Lemma can be used to check the sufficient condition for the stability of fuzzy systems of Takagi-Sugeno type given in Equation (4-2). Consider a set of m fuzzy rules like (4-2),

$$\begin{gathered} \text{IF } x(k) \text{ is } A_1^1 \text{ and ... and } x(k-n+1) \text{ is } A_n^1 \text{ THEN } x^1(k+1) = A_1\underline{x}(k) \\ \vdots \\ \text{IF } x(k) \text{ is } A_1^m \text{ and ... and } x(k-n+1) \text{ is } A_n^m \text{ THEN } x^m(k+1) = A_m\underline{x}(k) \end{gathered} \tag{4-20}$$

where A_i matrices for $i = 1,...., m$ are defined by (4-3). One can now formulate all the m matrices A_i, $i = 1,...., m$, as an interval matrix of the form (4-17) by simply finding the minimum and maximum of all the elements at the top row of all the A_i matrices. In other words, we have

$$A_I(k) = \begin{bmatrix} [\underline{a}_1, \bar{a}_1] & \dots & [\underline{a}_n, \bar{a}_n] \\ 1 & \dots & 0 \\ 0 & & : \\ : & 1 \dots & 0 \\ 0 & \dots 1 & 0 \end{bmatrix} \tag{4-21}$$

where $\underline{a}_i$ and $\bar{a}_i$ for i= 1,...., m are the minimum and maximum of the respective elements of the first rows of A_i in (4-3). Using the above definitions and observations, the fuzzy system (4-20) can be rewritten (for simplicity) in compact form as

$$\text{IF } \underset{\sim}{x}(k) \text{ is } \underset{\sim}{A} \text{ THEN } \underset{\sim}{x}(k+1) = A_I(k)\underset{\sim}{x}(k) \tag{4-22}$$

where the antecedent of the above rule corresponds to all the m rules of (4-20), i.e. $\underset{\sim}{x}(k)$ corresponds to all m $\underset{\sim}{x}(k)$'s and $\underset{\sim}{A}$ corresponds to all the m "and"-ed antecedents. Now, we consider the following lemma:

Lemma 4.3 The fuzzy system (4-22) is asymptotically stable if the interval matrix $A_I(k)$ in (4-21) is asymptotically stable, i.e., the conditions of Lemma 4.2 are satisfied.

4.5 Stability of Fuzzy Logic Controllers Using Describing Functions

Here we will look into a stability criterion using describing functions. Marin and Titli (1995) also discuss the use of describing functions for the design of FLC's, but this will not be investigated further in this study.

We will first look at stability in Sinusoidal Input Describing Function (SIDF) sense, then at the SIDF of the Takagi-Sugeno type FLC.

4.5.1 Stability in SIDF sense

The SIDF of a non-linear element NL is defined by:

$$L_{NL}(X, \omega) = \frac{2}{\pi \cdot X}\int_0^{\pi}\left(NL(r) \cdot je^{-j\alpha} \cdot d\alpha\right) \tag{4-23}$$

with $r(t) = X \cdot \sin(\omega t)$, $X \in D_a$, $D_a = [A_i, A_s]$, $(A_i, A_s) \in \Re^2_+$ and the assumption that NL has a symmetric characteristic, i.e $NL(-r) = -NL(r)$.

Physically, this approximation means that the describing function L_{NL} replaces NL by neglecting higher harmonics. Therefore, the open-loop system (OLS) must be a low-pass system, i.e:

$$\forall X \in D_a, \omega : \left\|L_{OLS}(X, \omega)\right\| \gg \left\|L_{OLS}(X, n \cdot \omega)\right\|, \text{ for } n \geq 2 \tag{4-24}$$

With the transfer function of the plant and the SIDF of the NL, a criterion of stability in SIDF sense, using the Nyquist criterion, is derived:

Definition 4.4 The closed loop system is globally stable in D_a in SIDF sense iff the Nyquist plot of L_{OLS} satisfies the Nyquist criterion for X in D_a.

4.5.2 The SIDF of the Takagi-Sugeno type FLC

We will here assume a Takagi-Sugeno type FLC[1] with error and change of error as inputs, and constants in the output functions. The fuzzy rules are on the form:

$$\text{rule } i\text{: if } e_n(t) \text{ is } A_i \text{ and } \dot{e}_n(t) \text{ is } B_i \text{ then } u_n^i = c_i \tag{4-25}$$

where A_i and B_i are fuzzy sets over normalised universes of discourse, defined by their membership functions $\mu_A(e_n)$ and $\mu_B(\dot{e}_n)$. The inputs and the output are scaled in the following way: $e_n = K_e \cdot e$, $\dot{e}_n = K_{\dot{e}} \cdot \dot{e}$, and $u = K_u \cdot u_n$.

The output of the FLC is then:

$$FLC(e, \dot{e}) = u(e, \dot{e}) = K_u \cdot \frac{\sum_i v^i(e_n, \dot{e}_n) \cdot c_i}{\sum_i v^i(e_n, \dot{e}_n)} \tag{4-26}$$

where $v^i(e_n, \dot{e}_n) = \min(\mu_{A_i}(e_n), \mu_{B_i}(\dot{e}_n))$ using the min-operator (see also Section 3.2.2 "Takagi-Sugeno Fuzzy Logic Controller" on page 17.).

The following assumptions are sufficient for obtaining a symmetric FLC:

$$\forall (e_n, \dot{e}_n, i) \begin{cases} \mu_{A_i}(e_n) = \mu_{A_{-i}}(-e_n) \\ \mu_{B_i}(\dot{e}_n) = \mu_{B_{-i}}(-\dot{e}_n) \\ c_i = -c_{-i} \end{cases} \tag{4-27}$$

With the assumptions in (4-27) the Global Sinusoidal Input Describing Function (GSIDF) can be shown to be:

$$L_{FLC}(X, \omega) = K_u \cdot K_e \cdot \sum_{i \geq 0} L_n^i(X_n, \omega_n) \cdot c_i \tag{4-28}$$

with

$$L_n^i(X_n, \omega_n) = \frac{2}{\pi \cdot X_n} \int_0^{\pi} \left(m_n^i(e_n, \dot{e}_n) \cdot je^{-j\alpha} \cdot d\alpha \right) \tag{4-29}$$

and where

$$\begin{aligned} m_n^i(e_n, \dot{e}_n) &= \frac{v^i(e_n, \dot{e}_n)}{\sum_i v^i(e_n, \dot{e}_n)} \\ X_n &= K_e \cdot X \\ \omega_n &= \frac{K_{\dot{e}} \cdot \omega}{K_e} \\ e_n &= X_n \cdot \sin\alpha \\ \dot{e}_n &= X_n \cdot \omega_n \cdot \cos\alpha \end{aligned} \tag{4-30}$$

1. (Marin and Titli, 1995) considers the general case of FLC

The equations ((4-28)-(4-29)) have to be solved numerically by simulations, as no analytical functions exists. It is then possible to find the GSIDF over a range of amplitudes and frequencies. The GSIDF of the FLC can be used to show stability together with the transfer function of the plant to be controlled using Definition 4.4. This will in fact be done with a TSFLC for the TBT control loop with the 3rd order brine-heater model, in Section 5.4 "Stability of the Fuzzy Control System" on page 40.

In this chapter a number of sufficiency conditions have been presented to check for the asymptotic stability of fuzzy control systems with Takagi/Sugeno type rules, i.e. Equation (4-1). All the criteria presented here are somewhat conservative. It is noted that if a given condition, say Equations (4-6, 4-11, or 4-18) is not satisfied it does not mean that the system is necessarily unstable. On the other hand, if the condition is true, then the system is, in fact, stable. The contents of this section hopes to serve as a starting point for many new results to come toward a solid stability theory for fuzzy control systems. This challenge still exists for both control engineers and mathematicians.

5 Simulationresults

In this chapter we present three conventional control strategies used in this study, and the implementation with the non-linear brine heater models on MATLAB/SIMULINK. A TSFLC is designed and implemented also on MATLAB/SIMULINK. Four different simulation scenarios are chosen, simulated, and the results are presented. The stability of the designed TSFLC control system is shown, using the stability criterion in SIDF sense, as discussed in Chapter 4.

5.1 Control Strategies

5.1.1 Conventional Control

(Babcock Inc., 1994) has analysed three control structures for the TBT's control loop. These are PID control, cascade control and disturbance rejection control[1].

PID Control

This strategy consists of one PID controller that observes the TBT and controls the steam valve. The control scheme is shown in Figure 5-1.

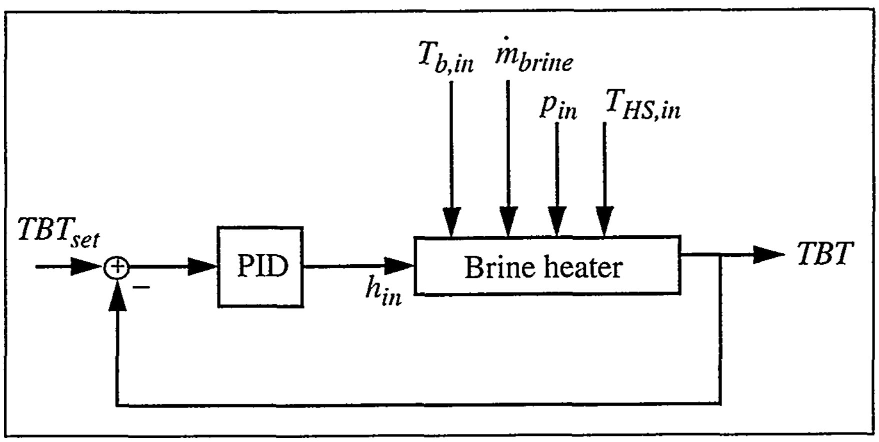


Figure 5-1: PID control scheme

The textbook algorithm of a PID controller is:

$$u(t) = K_p\left(e(t) + \frac{1}{T_i}\int_0^t e(s)ds + T_d\frac{de}{dt}\right) \tag{5-1}$$

1. In (Babcock Inc., 1994) they call it *disturbance value control*

The model is not linearisible, so well known design methods, like pole placement techniques, can not be used. There are methods that use the open-loop response of a step input, like the Ziegler-Nichols open-loop method (Åström and Wittenmark, 1989), to determine the PID parameters. In (Babcock Inc., 1994) they use another method, proposed by Chien, Hrones and Reswick (1952)[1]. This method is similar to the one of Ziegler-Nichols (Åström and Wittenmark, 1989), but with other adjusting parameters. (Babcock Inc., 1994) then found the following PID parameters:

- K_P=0.47, T_i=9.65 and T_d=0.91.

Cascade Control

This strategy consists of two controllers, as shown in Figure 5-2. The TBT is controlled by a main loop controller which gives the setpoint to a secondary controller. The secondary controller acts on the steam flow and controls the valve opening. The parameters for the two controllers were found in (Babcock Inc., 1994), where they used the same method as for the PID control to find the controllers parameters.

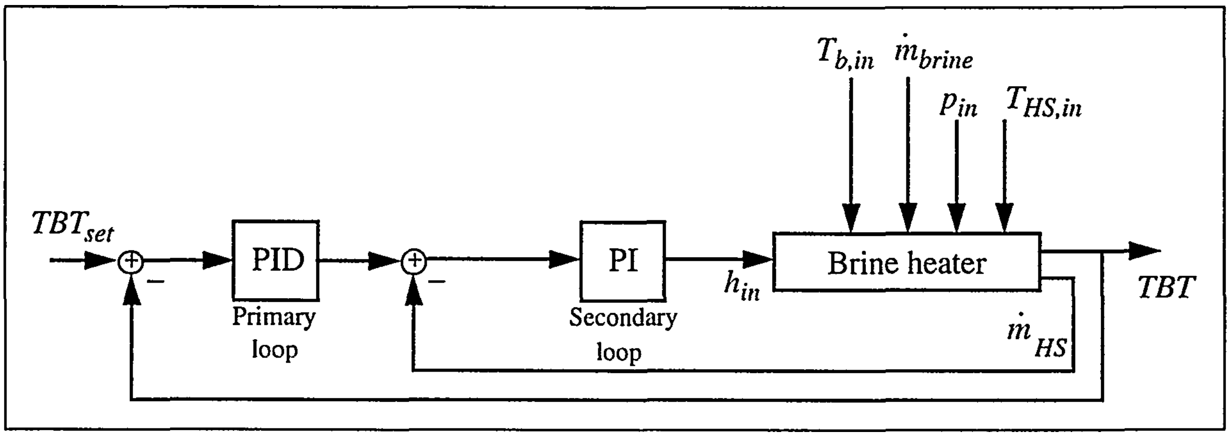


Figure 5-2: Cascade control scheme

The parameters are:

- Main controller (PID):

  K_P=23.79, T_i=10.45 and T_d=0.95

- Secondary controller (PI):

  K_P=0.018 and T_i=5.46

Disturbance Rejection Control

This strategy is based on the cascade control scheme, but also takes the disturbances steam pressure and recycle flow into account before they influence the TBT. This strategy is shown in Figure 5-3.

This is done by measuring the disturbances, and giving an extra control signal to the valve. The disturbance rejection function is in (Babcock Inc., 1994) and is found to be:

1. In (Babcock Inc., 1994) they give no further references.

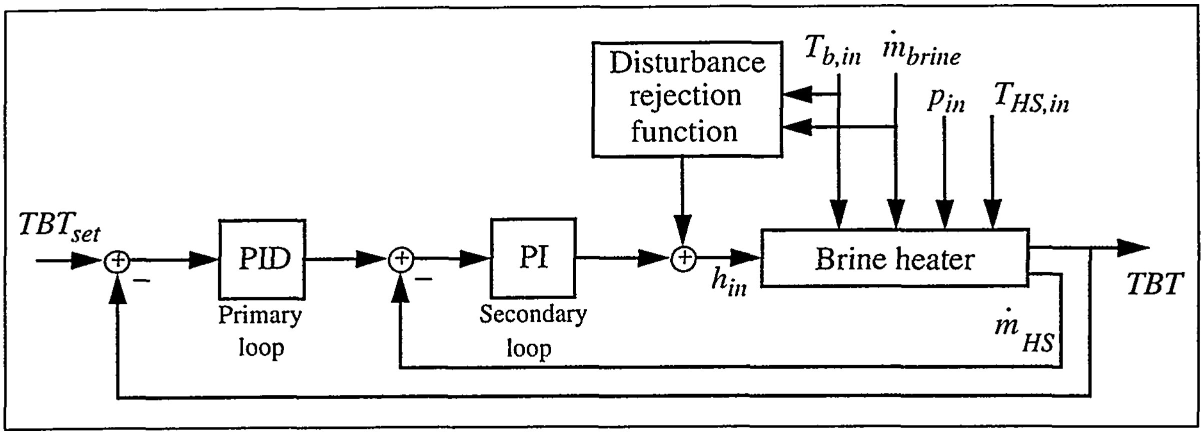


Figure 5-3: Disturbance rejection control scheme

$$dh_{comp}(T_{b,in}, \dot{m}_{brine}) = 0.084 \cdot \frac{1}{°C}(88°C - T_{b,in}) + 1.48 \times 10^{-4} \frac{kg}{s}\left(\dot{m}_{brine} - 3888\frac{kg}{s}\right) \quad (5\text{-}2)$$

5.1.2 Fuzzy Control

We wanted to construct a simple FLC, so we constricted our FLC to have two inputs and one output. We then decided to use a PID-like TSFLC. With a two dimensional input, one has the choice between a PD or a PI-like controller. An integral action is normally needed to achieve a good performance, so the PI-type TSFLC was chosen for further design. The block diagram of this control structure is the same as in Figure 5-1, only with an FLC in the place of the PID block.

The inputs to this controller are e, defined as $e = y_{set} - y$, and $\dot{e}$, defined as $\dot{e} = \frac{de}{dt}$. The output is $\dot{u}$. The output is a function of the inputs, and when it is integrated it gives the control $u = FLC(e, \int e)$, which is a PI-like controller. The input space was partitioned (for both inputs) into three fuzzy sets, 'NEG', 'ZE' and 'POS' on the normalised universe of discourse, as shown in Figure 5-4.

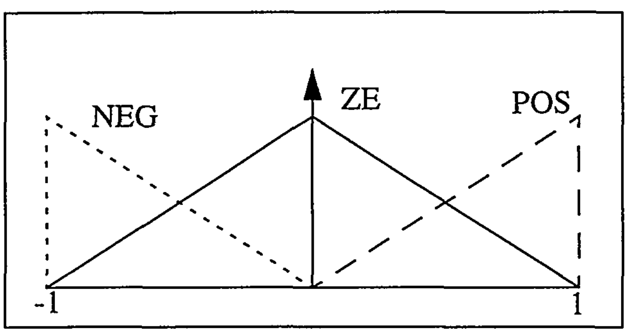


Figure 5-4: Fuzzy input sets

The output functions were chosen to be singletons on the normalised universe of discourse, with the following linguistic labels and singleton values: 'NEG'=-1, 'ZE'=0, and 'POS'=1.

With the inputs partitioned into three fuzzy sets, the number of possible rules are 3^2=9. The rule base was then constructed by using meta rules for control such as:

if error is above the setpoint and moving away, then decrease control

if error is zero and not changing, then control is unchanged

The above two examples can be expressed as the fuzzy rules:

$$\begin{gathered}\text{if } e \text{ is NEG and } \dot{e} \text{ is NEG then } \dot{u} \text{ is NEG}\\ \text{if } e \text{ is ZE and } \dot{e} \text{ is ZE then } \dot{u} \text{ is ZE}\end{gathered} \tag{5-3}$$

This was done with all the input combinations, and the rules can be expressed as in Table 5-1.

e \ $\dot{e}$	NEG	ZE	POS
NEG	NEG	NEG	ZE
ZE	NEG	ZE	POS
POS	ZE	POS	POS

Table 5-1: The TSFLC rule base

The initial input scaling factors, K_e and $K_{\dot{e}}$, were chosen by looking at the maximum error, e_{max}, and change of error, $\dot{e}_{max}$, under PID control. We then chose $K_e = 1/e_{max} = 1/0.8$ and $K_{\dot{e}} = 1/\dot{e}_{max} = 1/0.15$. Initial output scaling was determined by knowing that the maximum change of the valve actuator is 0.02, thus giving K_u =0.02.

For implementation, a fuzzy control toolbox (Babuska, 1994) for MATLAB was first used. This turned out to be very slow, as it is very general. Therefore a TSFLC was implemented in an m-file, **sugeno.m,** which can be found in Appendix A. This gave a quicker inference time than the fuzzy control toolbox, but was still a bit slow. Therefore, in order to further enhance computational performance, the MATLAB file was translated to a C-file, **sugeno.c**, using MC[1] (Comsol). The translated file was then compiled to a *mex*-file, resulting in the file **sugenomex.mex4**. This file can be called from MATLAB/SIMULINK. The evolution of these controllers, from the fuzzy control toolbox to the mex-file, had a drastic effect on the inference time. The inference time was improved by a factor of about 70.

By simulation of the closed-loop fuzzy control system, it was found that the output gain chosen was too low. By a procedure of trial and error, a final value of K_u=0.0375 was found. The input scaling factors were left unchanged.

The control surface of the TSFLC is shown in Figure 5-5.

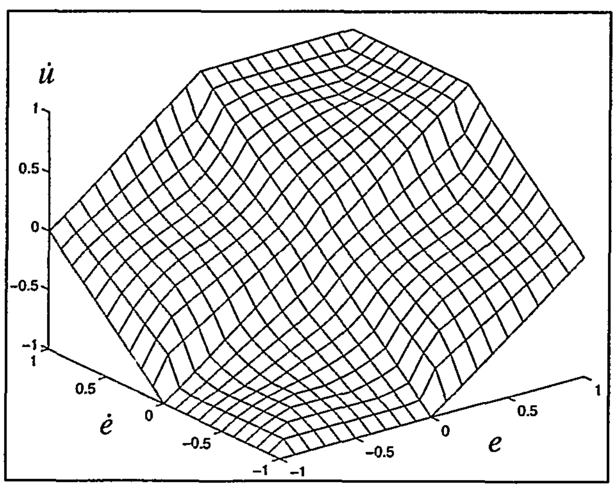


Figure 5-5: Normalised control surface of final FLC

1. This program is a MATLAB to C-translator.

5.2 Implementation of Brine Heater Models and Control Strategies

All implementation of models and control strategies were programmed and simulated using MATLAB/SIMULINK (MathWorks, 1993) on a Sun4 workstation under the UNIX operating system. Two different models were used for simulation. They were first implemented as ordinary m-files, but later implemented as mex-functions in MATLAB. This improved the simulation time with a factor of about 45. The two models are:

- The original 3rd order model of (Babcock Inc., 1994) in **brineheatermex3.c**.
- Our extended 12th order model in **brineheatermex.c**.

The listings of the models can be found in Appendix A. In the listings, one will also find the physical properties and constants used for the models. These models are then called from within SIMULINK, and connected to the different control strategies. With 4 different control strategies and 2 models, the total number of block-diagrams is 8. One of these SIMULINK block-diagrams, the FLC scheme for the 3rd order model (**brine_h_flc.m**[1]), is shown in Figure 5-6. The other block-diagrams are found in Appendix A.

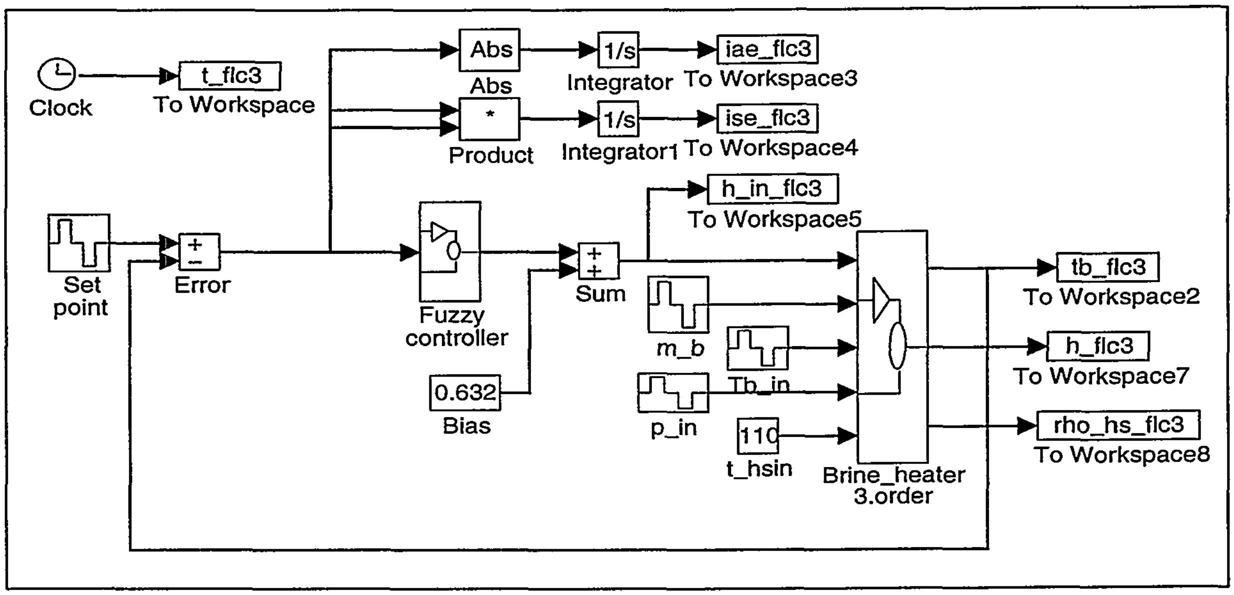


Figure 5-6: SIMULINK block-diagram with fuzzy logic controller

The integration method used was *rk45*, which is a fifth order Runge-Kutta method. This method uses a variable integration time-step Δt which was chosen to be between $\Delta t_{min} = 0.1$ and $\Delta t_{max} = 0.2$. The accuracy of the integration was chosen to be $1 \cdot e^{-3}$. All simulations were performed for 240 seconds.

1. **brine_h12_flc.m** is the same, except it uses the 12th order **brineheatermex.c**

5.3 Simulations

5.3.1 Choice of Simulation Scenarios

Three different simulation scenarios were studied (Babcock Inc., 1994):

- Step in brine heater inlet temperature, $T_{b,in}$, from 88°C to 86°C. Simulation A.
- Step in brine recycle flow, $\dot{m}_b$, from 3888 kg/s to 4165 kg/s. Simulation B.
- Step in steam supply pressure, p_{in}, from 1.5 to 1.7 bar. Simulation C.

To investigate the response to a set-point change, we included a fourth simulation:

- Step in TBT setpoint, TBT_{ref}, from 95°C to 96°C. Simulation D.

As this being a non-linear simulation, the variable should change as in Figure 5-7. This is

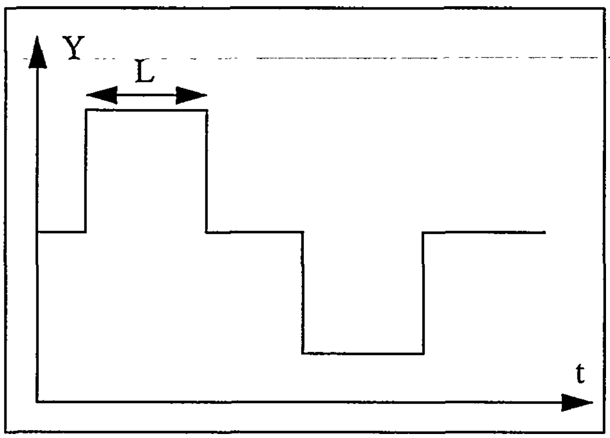


Figure 5-7: Variation of variables

because as a non-linear system, it behaves differently depending on the initial condition, the step size and the direction of the step.The second step is applied in the opposite direction, with the same magnitude, as the first step.

The simulation scenarios (A, B, and C) in (Babcock Inc., 1994) were reproduced on the 3rd order model, with great accuracy. This was taken as a verification of the implemented models and control strategies. We then simulated the 4 scenarios (with the variation of variable as in Figure 5-7) on the 3rd and 12th order model. The first step was applied after 2s, and the duration, *L*, was 60s. The results of these simulations follows in the next four sub-chapters, and are also discussed. In simulation C and D, the disturbance rejection scheme is not included, as it gives the same results as the cascade control scheme.

5.3.2 Simulation A. Step in brine inlet temperature

As shown in Figure 5-8 that of the conventional strategies on the 3rd order model, the PID controller has the largest overshoot, but about the same settling time as the cascade controller. The cascade controller is slow, but the overshoot is small. The settling time of the disturbance rejection control is slow, and slightly oscillating.

For the 12th order model, we notice that the disturbance rejection control gives an inverse-

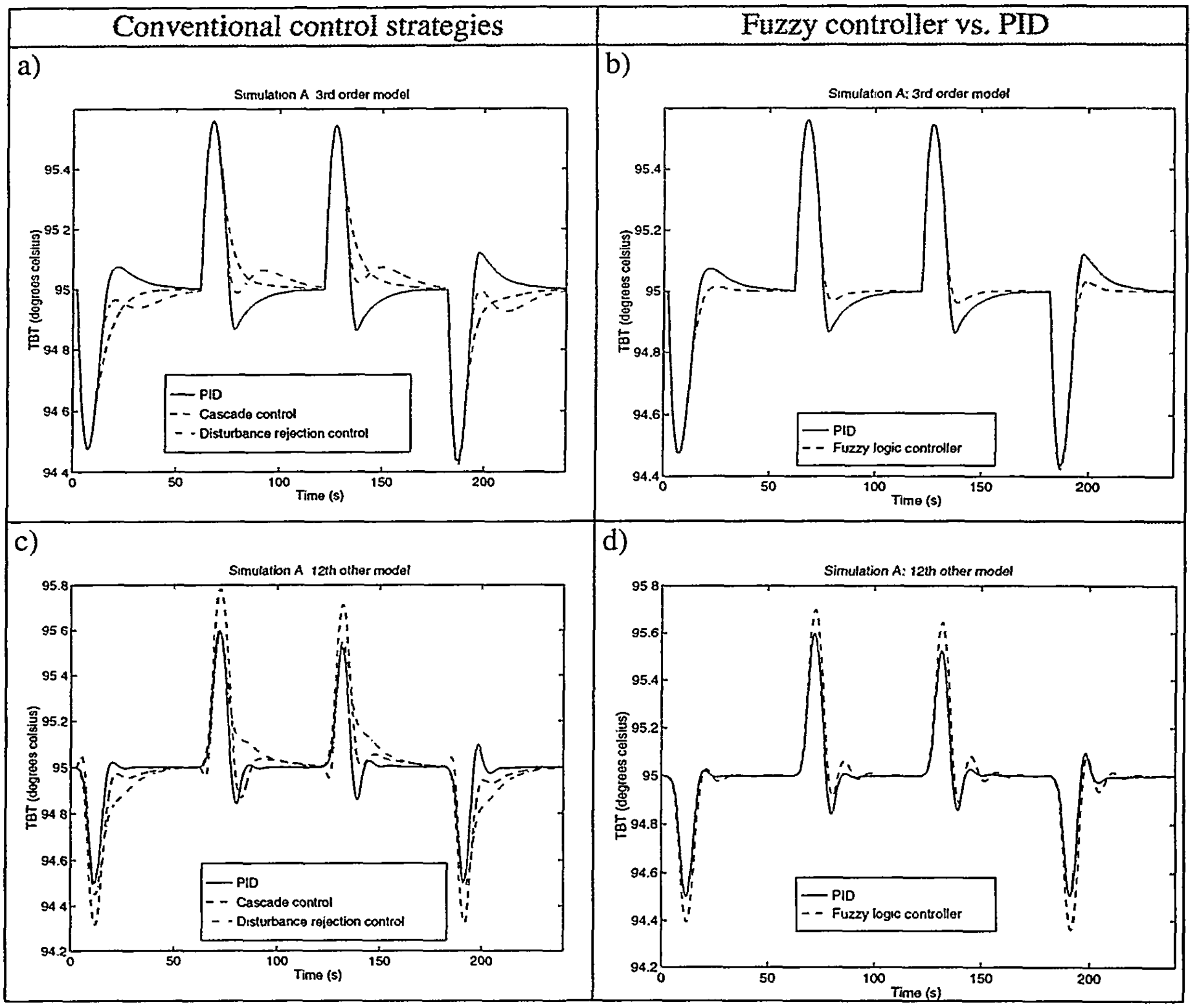


Figure 5-8: Simulation A. Step in brine inlet temperature

response. This is due to that the higher order system introduces a small time delay. The disturbance rejection function has no compensation for this effect, and this results in the inverse response.The PID control performs well, with a fast settling time, however somewhat oscillatory. The cascade control performance is a slightly slower on the settling time, but less oscillatory.

As for the FLC, it performs very well on the 3rd order model with a fast response and a small overshoot. For the 12th order model, the FLC gives a bigger deviation than the PID, but still has about the same settling time. The overshoot is less than for the PID, but the FLC is slightly more oscillatory.

5.3.3 Simulation B. Step in brine recycle flow

In this simulation we see (Figure 5-9) the effect of the non-linearity in the system. The FLC strategy shows an oscillatory behaviour for the 12th order model, but also exists for the conventional strategies with a smaller amplitude.

For the conventional strategies we see that the PID outperforms the other two, with lesser deviation and faster rise time, but with approximately the same settling time. There is no significant overshoot in any of the control strategies.

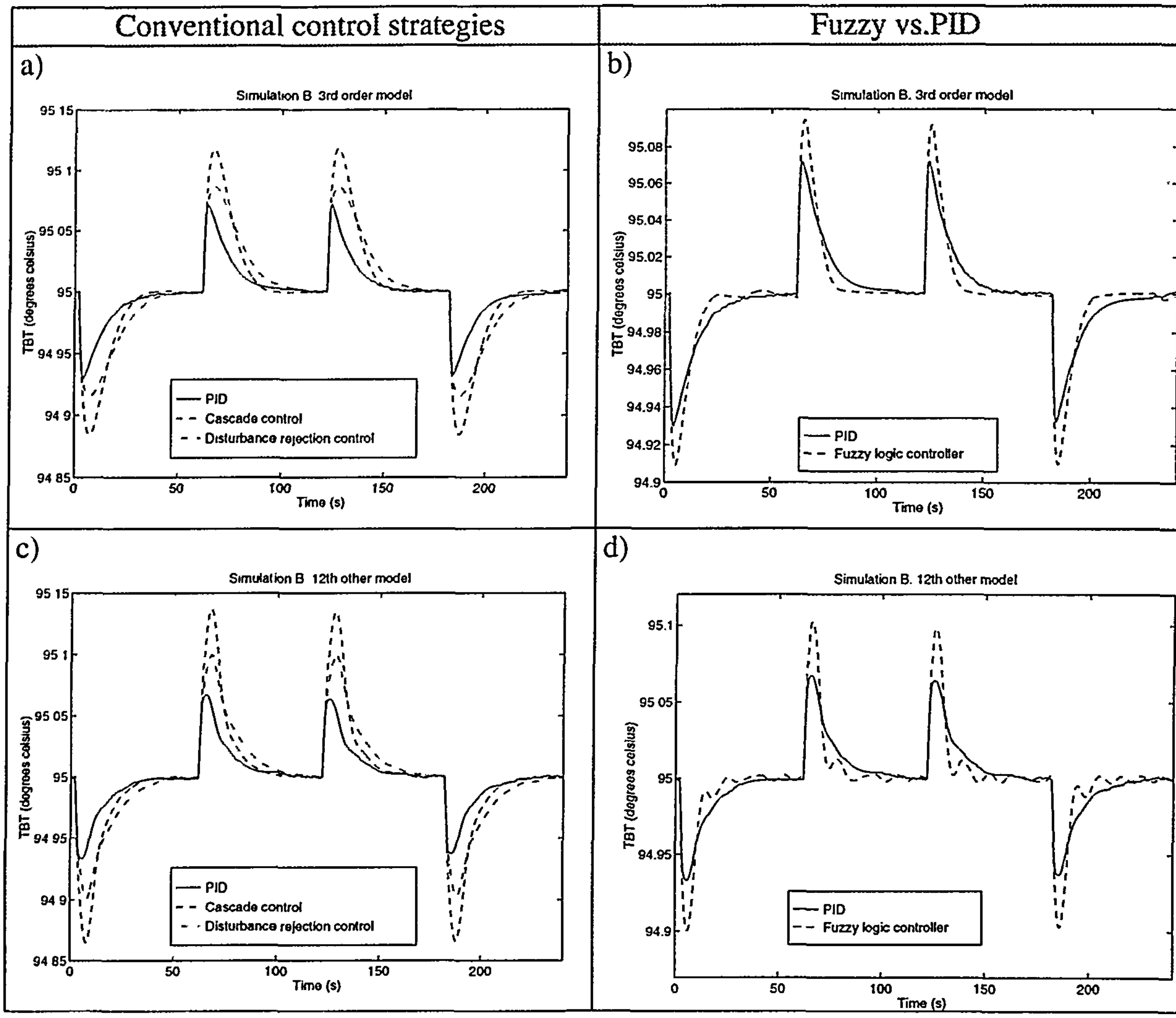


Figure 5-9: Simulation B. Step in brine recycle flow.

When comparing the PID and FLC, we see that the FLC allows a bigger deviation, but the settling time is quicker. However, as mentioned earlier, the FLC is oscillating more than the PID.

5.3.4 Simulation C. Step in steam supply pressure

The cascade control is heavily oscillating (see Figure 5-10) in the two last steps, and has not settled before the next step is applied. In the last step change it looks unstable, but will in fact stabilise. The PID clearly outperforms the cascade control, with less overshoot and a fast settling time. The controllers show about the same performance on the 12th order model.

The FLC has more or less the same performance as the PID, but gives slightly less overshoot. We notice that in the last step that the FLC is quicker than the PID in the 3rd order model, and oscillating more for the 12th order model.

This simulation scenario is a good example of the non-linearities in the process, and the importance of applying a step change in both directions from the nominal point. We see clearly that the system response depends on the initial condition and the step direction.

5.3.5 Simulation D. Step in TBT set point

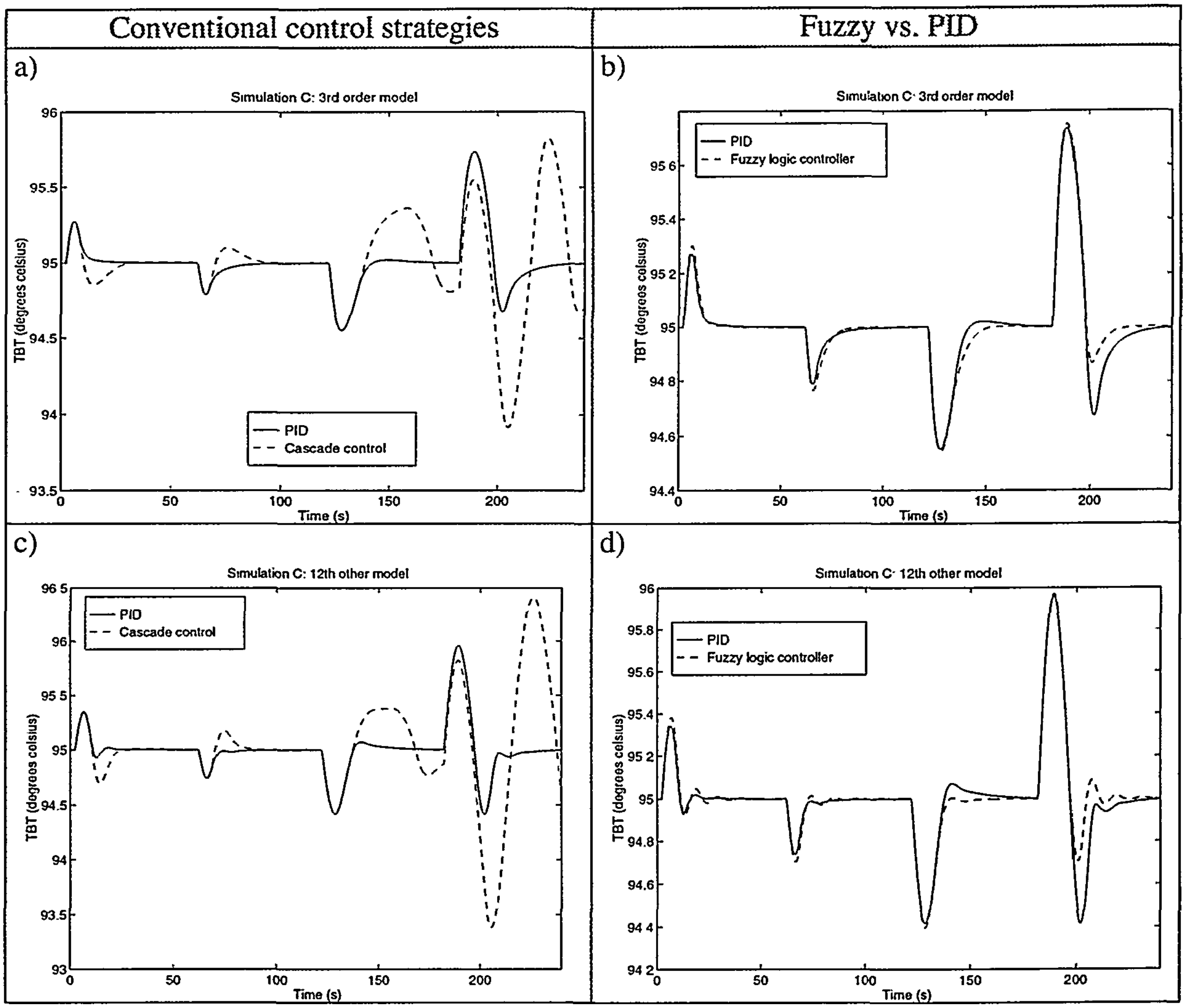


Figure 5-10: Simulation C. Step in steam supply pressure.

Shown in Figure 5-11, the cascade controller is oscillating and slow. The PID overshoots, and uses a long time to reach the steady state condition.

In this simulation the FLC is performing very well, with a quick response and insignificant overshoot.

The behaviour in both models is approximately the same.

5.4 Stability of the Fuzzy Control System

In our model we have a dynamic non-linear element in the form of the valve actuator. This is highly non-linear and not linearisible. In order to be able to show stability of the fuzzy control system, we have to find the SIDF for the brine heater model. We use the 3rd order model in our further stability analysis.

The stability procedure was as follows:

1. First we determine the amplitude, X, and frequencies intervals, ω. With a maximum valve opening, i.e. $h_{in} = 1$, the TBT was found to be 97°C. The amplitude interval was then determined to be $X \in [-3°C, 3°C]$. By looking at the previous simulations, the frequency interval was chosen to be $\omega \in [(2\pi)/30, (2\pi)/5]$. The number of discrete val-

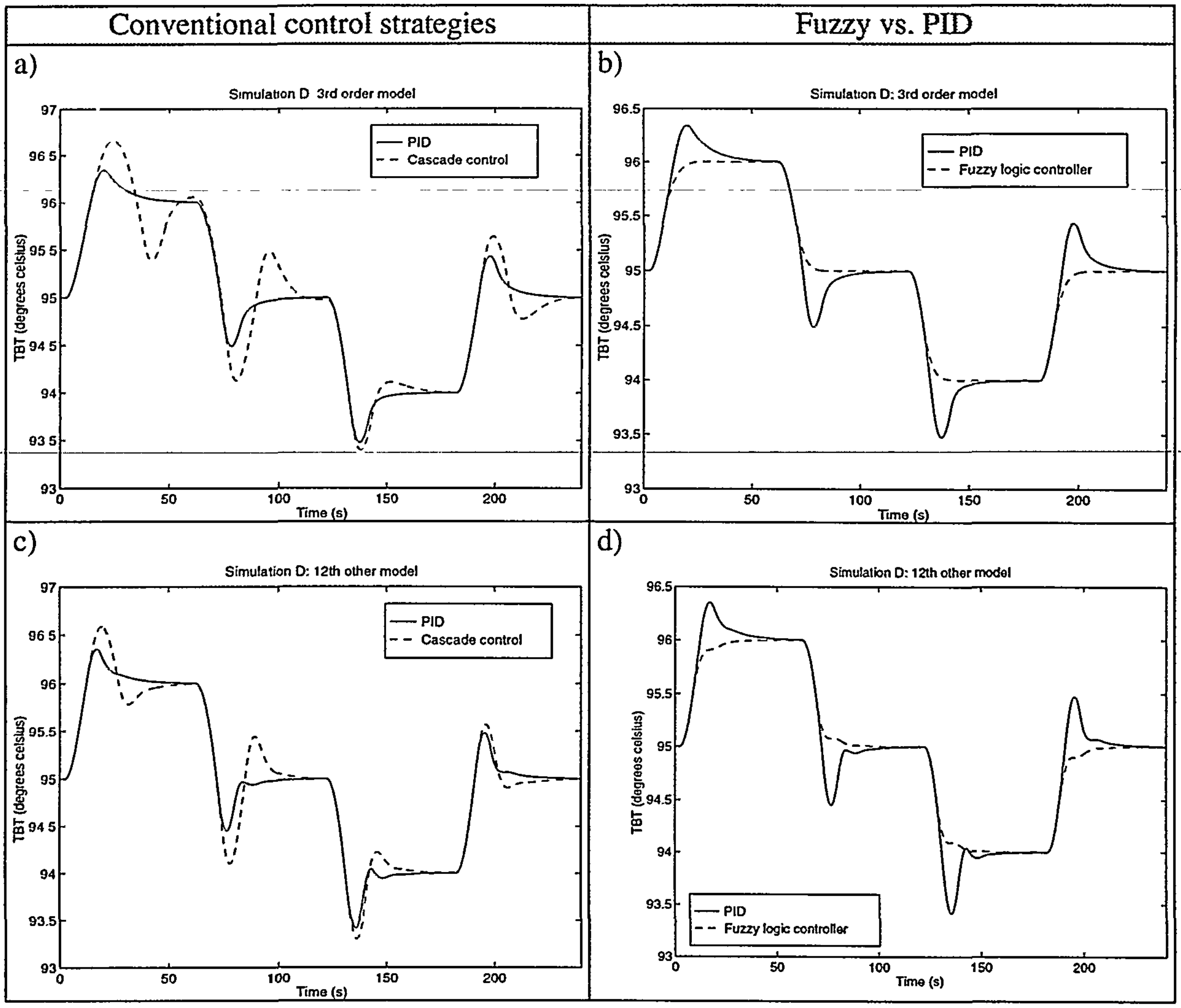


Figure 5-11: Simulation D. Step in TBT set point

ues for the amplitude interval, n_X, and the frequency domain, n_ω, were chosen to be 50. This gives a total number of possible input combinations to 2500. The intervals were then logarithmically spaced.

2. The SIDF of the FLC is found by using the Eq.(4-28) to Eg.(4-30). The SIDF of the FLC was then multiplied with the integrator, giving $L_{IFuzzy}(X, \omega) = L_{Fuzzy}(X, \omega) \cdot L_I(\omega)$. This was implemented in the file **SIDF.m**, found in Appendix A. The result is shown in Figure 5-12.

3. We then calculated the SIDF of the brine heater model. From the SIDF of the FLC we have 2500 output combinations. Therefore, to reduce this number of inputs to 50, a modified amplitude vector, X_{mod}, was constructed. This X_{mod} is then used as the input amplitude to the brine heater system. We find $X_{mod,max}$ and $X_{mod,min}$ by the following equations:

$$\forall (X, \omega): \; X_{mod,\,max} = \max(X \cdot \mathrm{abs}(L_{IFuzzy}(X, \omega))) \tag{5-4}$$

$$\forall (X, \omega): \; X_{mod,\,min} = \min(X \cdot \mathrm{abs}(L_{IFuzzy}(X, \omega))) \tag{5-5}$$

This gave us $X_{mod} \in [0.0241, 0.406]$, which was logarithmically spaced. The frequency domain is left unchanged. The 3rd order brine heater model was then simulated (with a fixed time-step $\Delta t = 0.05$) for 30 periods in order to simulate beyond transient conditions. This was implemented in the file **Lsys.m**, which can be found in Appendix A. For each

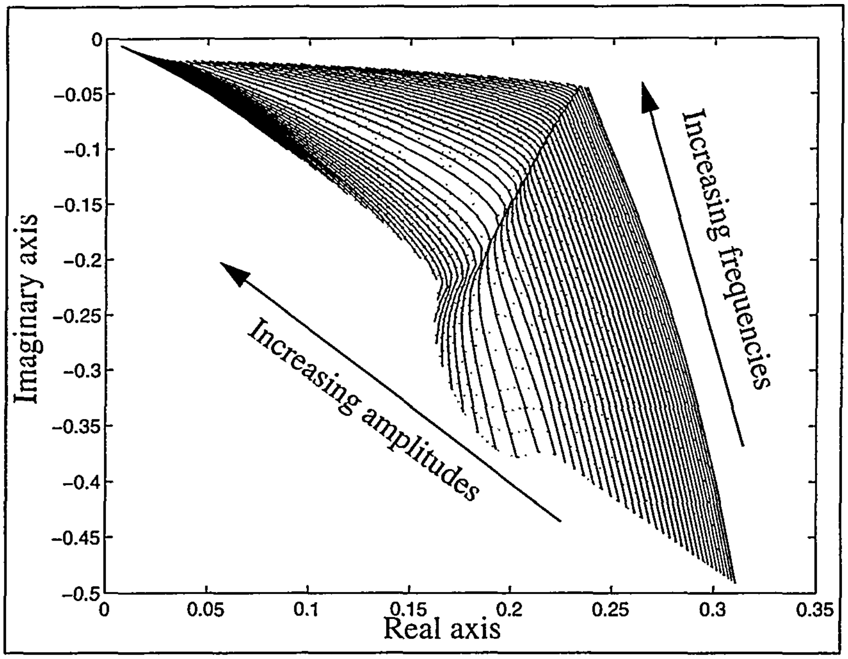


Figure 5-12: SIDF of FLC, $L_{IFuzzy}(X, \omega)$.

input combination, $X_{mod,i}, \omega_j$, the describing function was found using the discrete integral:

$$L_{system}(X_{mod,i}, \omega_j) = \frac{\omega_j}{X_{mod,i} \cdot \pi} \cdot \sum_{t=0}^{T_F} \mathrm{TBT}_{i,j}(t) \cdot \left(\sin(\omega_j t) + \sqrt{-1}\cos(\omega_j t) \right) \cdot \Delta t \qquad (5\text{-}6)$$

In order to be able to multiply the SIDF of the FLC and the SIDF of the system in the frequency plane, the $L_{system}(X_{mod}, \omega)$ has to be modified so it fits with the amplitude/frequency pairs given by the FLC. This was done by linear interpolation, giving the adjusted $L_{sys}(X_{mod}, \omega)$ as shown in Figure 5-13.

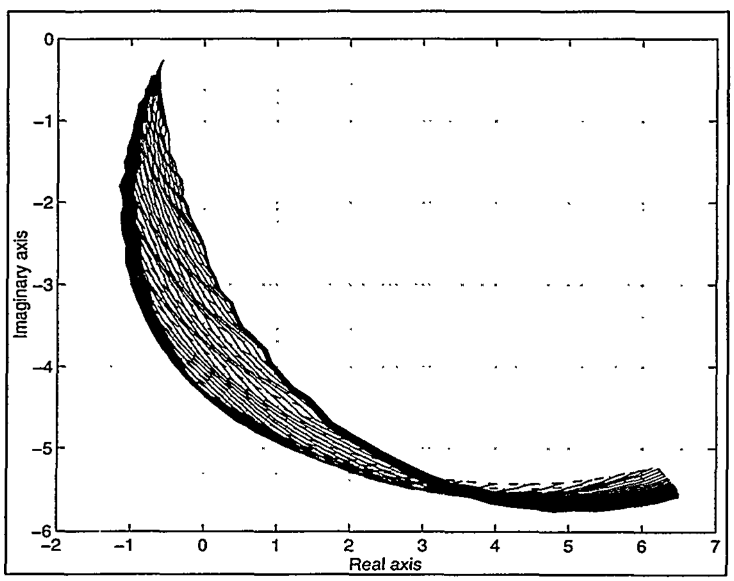


Figure 5-13: SIDF of 3rd order brine heater model, $L_{sys}(X_{mod}, \omega)$.

4. The Open-Loop System's (OLS) SIDF is obtained by:

$$L_{OLS}(X, \omega) = L_{IFuzzy}(X, \omega) \cdot L_{sys}(X_{mod}, \omega) \tag{5-7}$$

We can then plot a Nyquist plot of the OLS as seen in Figure 5-14. We see that the open-loop system's polar plot does not encircle *(-1,0)* thus we can conclude by Nyquist's Criterion that the closed-loop system is stable (assuming that the open-loop system is minimum-phase).

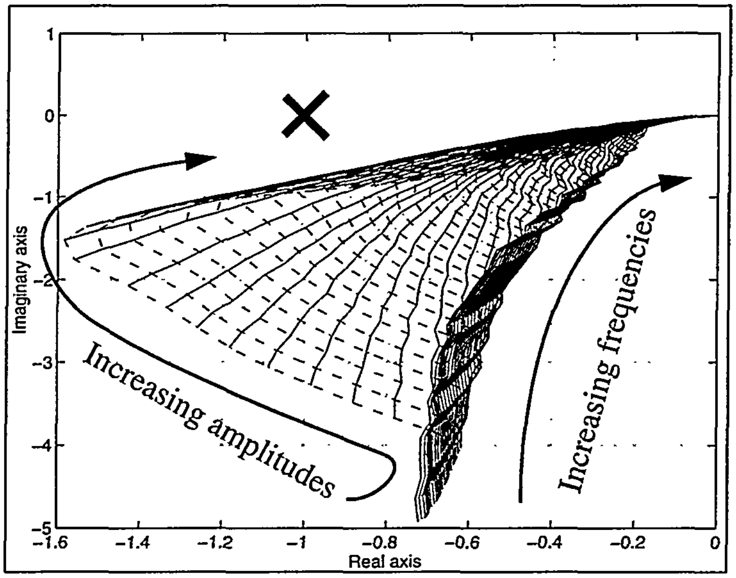


Figure 5-14: Nyquist plot of $L_{OLS}(X, \omega)$

6 Hierarchy and Hybridization of Fuzzy Logic Control Systems

6.1 Rule-Base Reduction

It is well known that the number of rules grows exponentionally with respect to the number of system's sensory feedback variables. Although the hierarchical structure suggested by Raju *et al.* (1991) does reduce the number of rules considerably, it is still not computationally effective. Consider, as an example, the all too familiar inverted pendulum problem. For the 4th order non-linear (or linear) model, the number of rules reduce from $5^4 = 625$ to $3 \cdot (5)^2 = 75$. Clearly, this number of rules is still too many and unacceptable. Moreover, this approach is not of any use for 2-variable systems, on one hand, and the intermediate variables $u_1, u_2, \ldots, u_L$ (see Figure 6-1) are not fuzzy sets of controller input sensory variables, i.e some rules loose their "readability" as an expert's opinion. They represent an aggregate of fuzzy decisions whose contributions into higher-level rules can not be obtained readily.

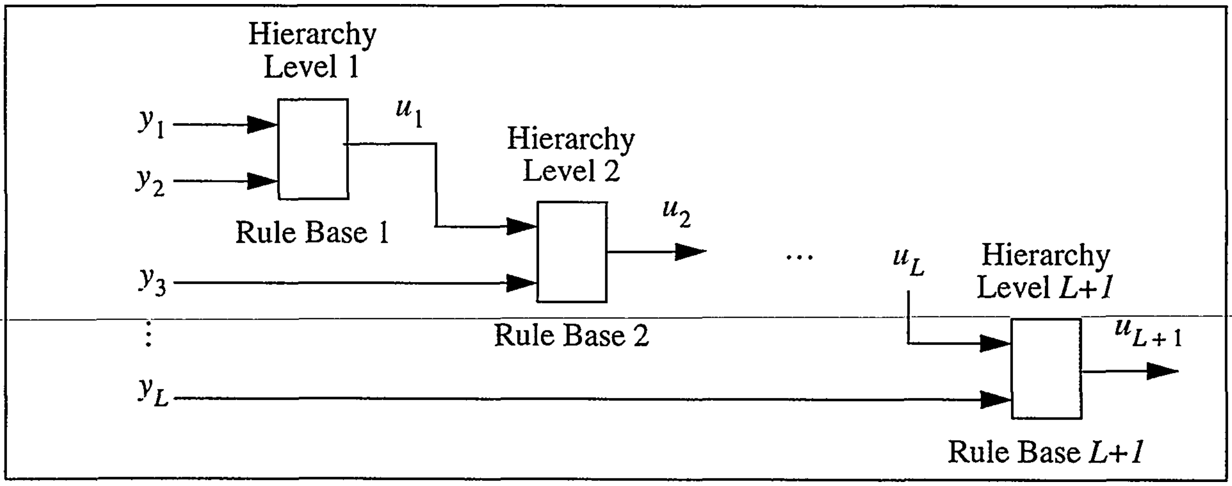


Figure 6-1: Hierarchical Fuzzy Controller

Here, the hierarchical structure of Raju *et al.* (1991) is combined as one layer of a systematic procedure to drastically reduce the number of rules, i.e. the size of the overall rule base or the inference engine.

6.1.1 Sensory Fusion

An approach for rule-base reduction is to investigate the physical possibility and feasibility of combining or fusing sensory signals (variables) before being fed to the fuzzy controller (inference engine). Assume that a fuzzy controller has three inputs (y_i, i=1,2,3) and one output (u). Furthermore, let each variable (input or output) be represented by 5 linguistic variables: NM (Negative Medium), NS (Negative Small), AZ (Approximately Zero), PS (Positive Small) and

PM (Positive Medium). This combination would call for $k = m^n = 5^3 = 125$ rules, described below:

$$
\begin{aligned}
&R_1: \text{ IF } y_1 \text{ is } A_1 \text{ and } y_2 \text{ is } B_1 \text{ and } y_3 \text{ is } C_1 \text{ THEN } u \text{ is } D_1 \\
&R_2: \text{ IF } y_1 \text{ is } A_2 \text{ and } y_2 \text{ is } B_2 \text{ and } y_3 \text{ is } C_2 \text{ THEN } u \text{ is } D_2 \\
&\vdots \\
&R_{125}: \text{ IF } y_1 \text{ is } A_{125} \text{ and } y_2 \text{ is } B_{125} \text{ and } y_3 \text{ is } C_{125} \text{ THEN } u \text{ is } D_{125}
\end{aligned}
\tag{6-1}
$$

Now, if one would look into combining the sensory data (variables y_i, i=1,2,3) in one of the following possible four ways:

(i) All three variables fused

$$Y = ay_1 + by_2 + cy_3 \tag{6-2}$$

where a, b, and c are positive parameters dictated by physical considerations and designer's experience.

(ii) Variables 1 and 2 are fused

$$Y_1 = ay_1 + by_2$$
$$Y_2 = y_3$$

(iii) Variables 1 and 3 are fused

$$Y_1 = ay_1 + by_3$$
$$Y_2 = y_2$$

(iv) Variables 2 and 3 are fused

$$Y_1 = ay_2 + by_3$$
$$Y_2 = y_1$$

In this manner the number of fuzzy rules would be reduced from 125 in the unfused (general) case to $k_1 = 5$ or $k_i = 25$, $i = 2,3$, and 4, depicting the above four cases. In this way a remarkable reduction in the size of the rule base would result. In fact, the reduction has a lower bound if <u>all</u> variables could, somehow, be fused. The rules for the first case are shown below:

$$
\begin{aligned}
&R_1: \text{IF } Y \text{ is } A_1 \text{ THEN } u \text{ is } D_1 \\
&\quad\vdots \qquad\qquad \vdots \\
&R_5: \text{IF } Y \text{ is } A_5 \text{ THEN } u \text{ is } D_5
\end{aligned}
\tag{6-3}
$$

Should, every two variables could be combined, then for an even number of variables, the reduction is even more pronounced. For example, if $n = 4$, then the rules would reduce from $5^4 = 625$ to $5^2 = 25$ - a 96% reduction versus 80% for $n = 3$. Figure 6-2 illustrates this simple idea for $n = 2, 3$ and4.

In spite of the remarkable reduction of the size of the rule base for lower values of n, if n is large, say $n = 10$, there are $5^{10} = 9,765,625$ rules in the original base which could be reduced to $5^5 = 3,125$ rules if every two sensory variable can be fused. Although $3,125$ is the upperbound by way of sensory fusion, the hierarchical structure of Raju *et al.* (1991) would result in only $9 \cdot (5)^2 = 225$ rules, i.e. a linear increase as a function of n. Now, as

$n > 10$, the number of rules get larger and larger regardless of what approach one would choose.

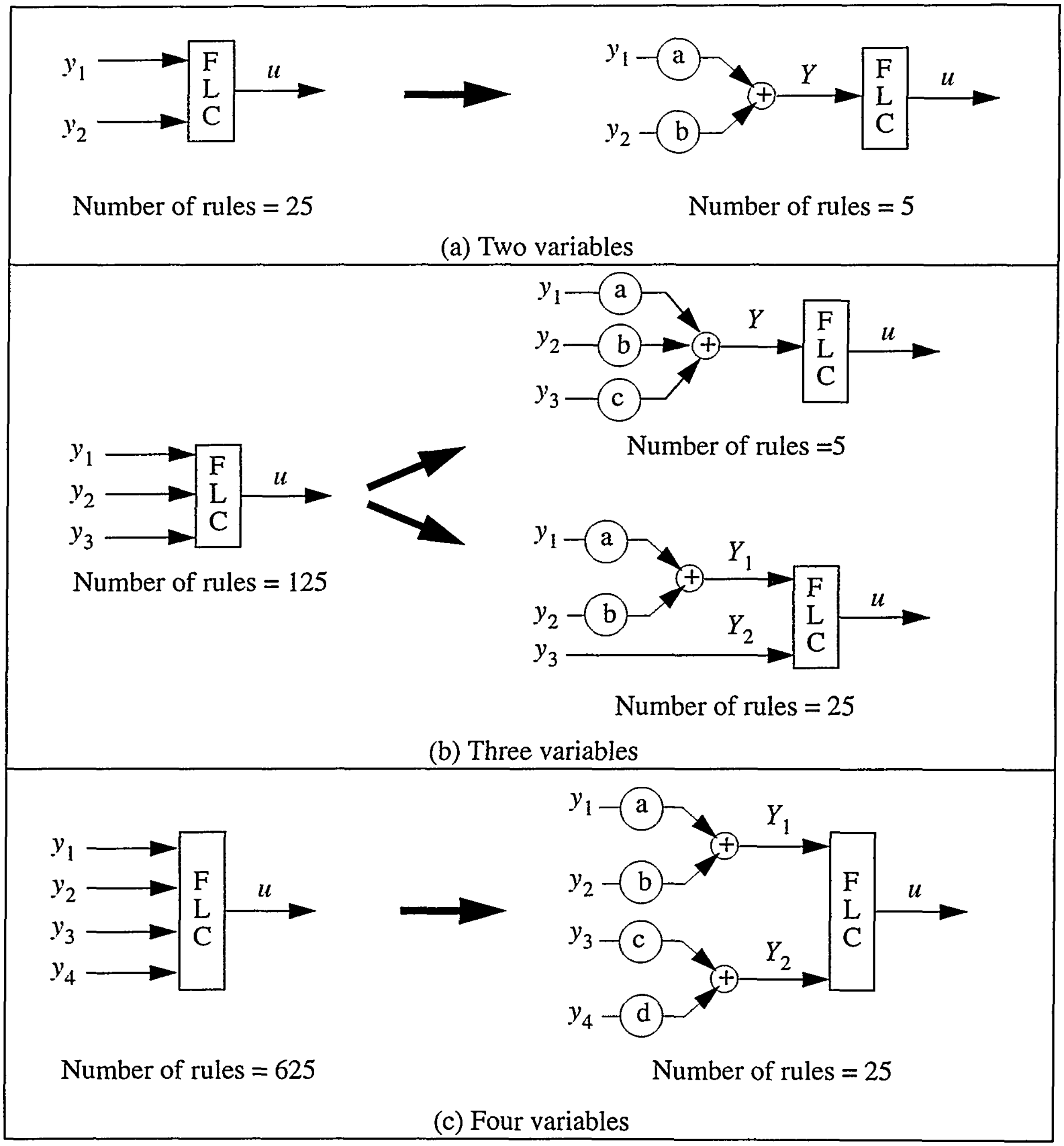


Figure 6-2: Fuzzy logic controller's rule base reduction for 3 cases

6.1.2 Hierarchy and Sensory Fusion Approach

Here, the variables are simply combined first, as in Figure 6-2, and then are organised into a hierarchical structure similar to that of Figure 6-1. For one or two input variables, there is no problem in reducing the size of the rule base. Figure 6-3 shows four possible fuzzy control structures corresponding to 3, 5, 8, and 10 input variables (sensory values). The reduction in the size of the rule base is quite remarkable. For the four cases shown the size of the rule base reduces from 80% for $n = 3$, 98.4% for $n = 5$, 99.98% for $n = 8$ to nearly 100%

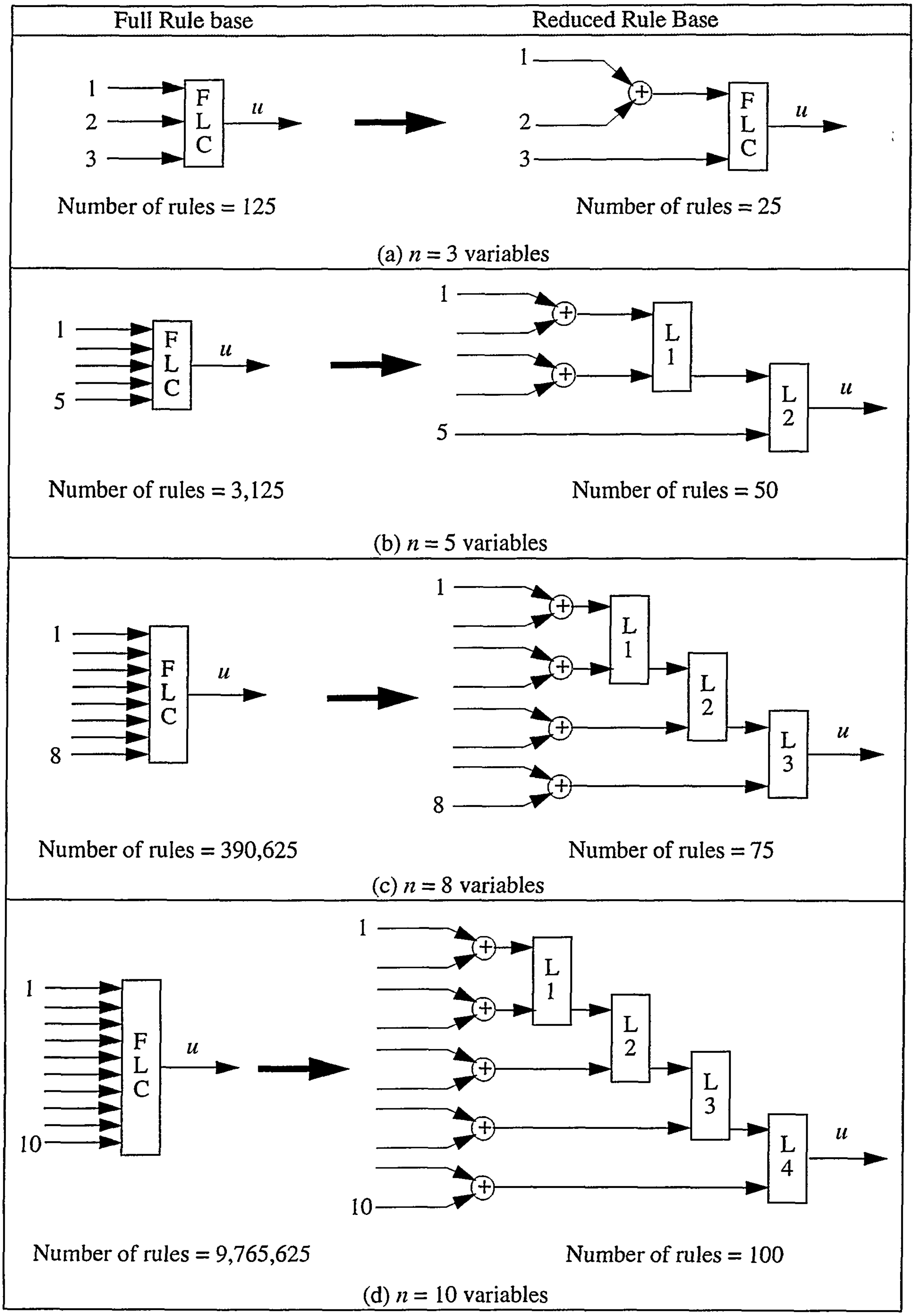


Figure 6-3: Rule base reduction using hierarchical and sensory fusion

(99.998976%) for $n = 10$. Figure 6-4 (a)-(c) shows four of five approaches for the size of the rule base. The first one is the single-level standard fuzzy controller rule base which is growing exponentionally. The second chart depicts the hierarchical structure of Raju *et al.* (1991), i.e. a linear increase in the number of rules before and beyond $n = 2$. The third chart shows the progression of the number of rules as a result of sensory fusion. The fourth graph represents the progression of rules as a result of combining sensory fusion (two variables at a time) and hierarchy of the rules. Finally, the fifth chart (Figure 6-4 b) shows a *very ideal* (impossible) or absolute minimum rules which could be obtained by fusing all variables and using a single level of hierarchy.

Clearly, depending on how many variables can be fused and in what order they are put into a hierarchical structure, the size of the rule base would be reduced differently. At this point, questions such as which variables to fuse with which ones and what are the most suitable values for coefficients *a*, *b* and *c* in (6-2) are open. The only certain issue here is that the decisions on which variables to fuse and which ones to group at which level of hierarchy depends mostly on one fact - *knowledge* about the system. For the time being, the fusion can be done through the following rule,

$$E = \alpha e + \beta \Delta e \tag{6-4}$$

where e and Δe are error and its rate of change E is the fused variable and parameters α and β are, at present time, chosen arbitrarily.

Going back to the proposed scheme of fuzzy rule base reduction, as the number of variables increase the percent reduction of rules quickly increases to "*nearly*" 100% as shown in Figure 6-5.

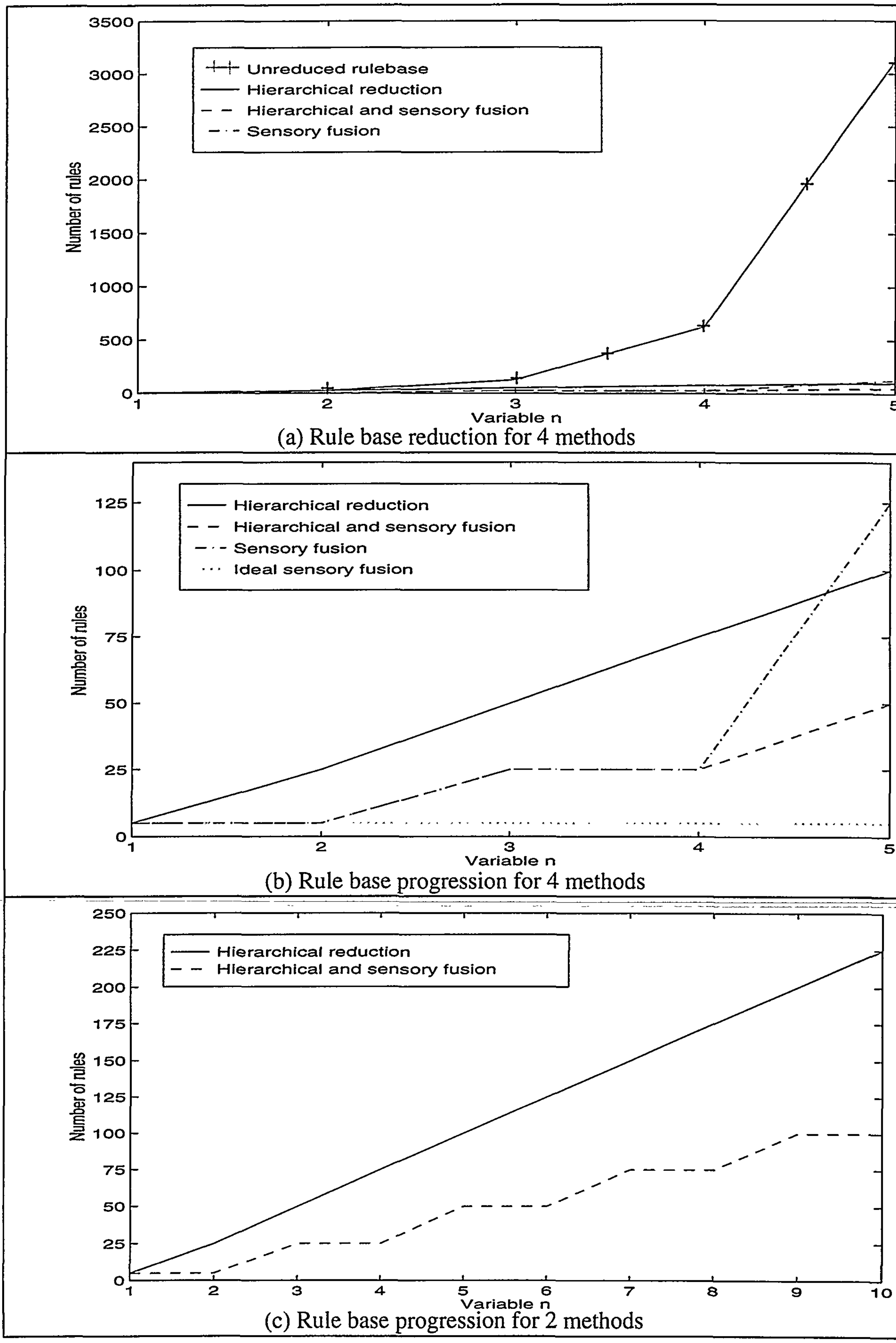
Next section will introduce schemes to reduce a rule base through hybridization and structural perturbation, decomposition, and decentralisation.

6.2 Hybrid Fuzzy Control Systems

The notions of hybridization and hierarchy in which either a fuzzy controller is combined with another controller, e.g. conventional or use fuzzy logic for both control purpose and plant behaviour (state identification). Here, what we mean by "*state identification*" is a classification of various states that a given system can attain throughout its dynamic history. As an example, consider the case of a flexible link robot (Akbarzadeh and Jamshidi, 1994) where the link can be "straight", "oscillating", "bent to the right", "bent to the right", etc. The consequences of the rules which determine the *approximate* behaviour of the robot would, in turn, influence the behaviour of a low-level fuzzy controller either through the inference engine or membership functions or both.

Sayyarrodsari and Homaifar (1995) have presented a similar concept in their work in which a fuzzy partitioning of the input space of the controller would allow the high-level hierarchy define the *firing boundaries* of the lower-level rules within the inference engine.

The integration of fuzzy control or fuzzy logic with standard or non-standard approaches of control, hierarchy, decentralization, model reduction, search, optimization, and clustering will be briefly presented. Some of the proposed approaches in this section are intentionally left as an open research problem to entice some interest in the readers.



(a) Rule base reduction for 4 methods

(b) Rule base progression for 4 methods

(c) Rule base progression for 2 methods

Figure 6-4: Rule base size for 5 possible large-scale fuzzy control structures

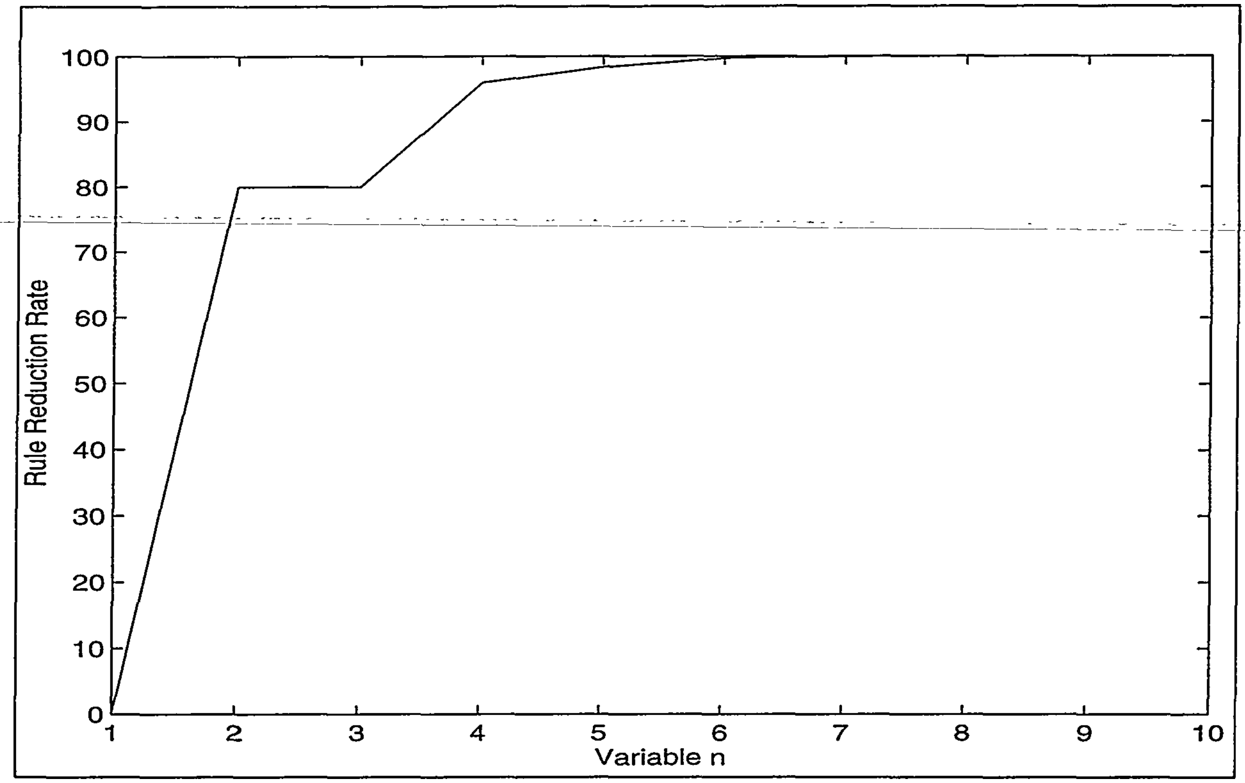


Figure 6-5: Percent rules reduction vs. variable *n*. Hierarchical-Sensory fusion method

6.2.1 Fuzzy-PID Controller

The set-point accuracy of standard PID controllers and non-linear characteristics of fuzzy controllers can be integrated to a hybrid fuzzy-PID architecture. The adaptation of a PID (multi-term) controller is presented in (Jamshidi, 1996). Here, a different point of view and structure is presented (Pedrycz, 1993).

Consider a hybrid structure as in Figure 6-6. The switch s/w provides a combination of the

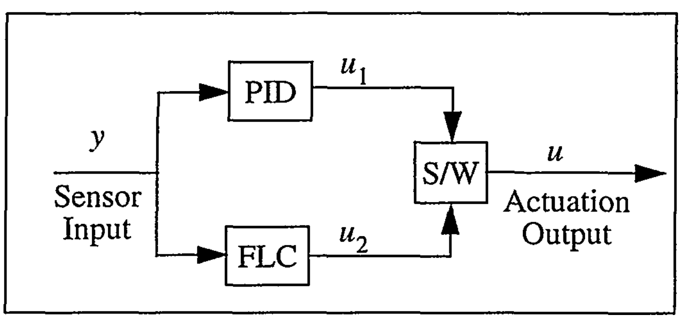


Figure 6-6: A hybrid fuzzy-PIID control structure

control signal u_1 by the fuzzy logic controller and u_2 by the PID controller. The control u is given by

$$u = \alpha u_1 + (1 - \alpha) u_2 \tag{6-5}$$

where

$$\alpha = \frac{c(e, \Delta e)}{c(e, \Delta e) + f(e, \Delta e)}$$

and $c(e, \Delta e)$ and $f(e, \Delta e)$ are two fuzzy sets representing a degree or a measure of being *close* to and *far* from the zero error as functions of error and change of error.

Figure 6-7 shows a schematic of these fuzzy sets. In this way, the control (6-5) would behave as follows: When error is large positive or negative, i.e. "far" set membership-value is close to one, i.e $\alpha \approx 0$ and $u \approx u_2$, a fuzzy controller. When the error e nears the zero value (positive or negative), $\alpha \approx 1$ and control $u \approx u_1$, a predominantly PID controller.

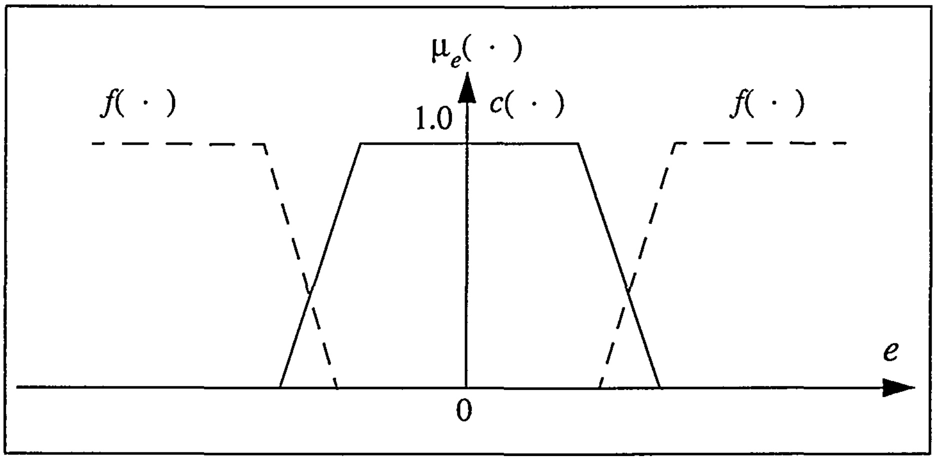


Figure 6-7: Fuzzy sets representing notions of *close* and *far* for error *e*.

This scheme of hybrid control can also be used between fuzzy logic controller and any other conventional schemes such as, *sliding mode*, *optimal*, etc.

6.2.2 Decentralized Fuzzy Control

Another approach to introduce fuzzy logic control in large-scale complex systems is to apply a set of fuzzy rules for each decentralised local controller as discussed in Chapter 5 of Jamshidi (1996). Consider a linear discrete-time large-scale system

$$x(k+1) = Ax(k) + \sum_{i=1}^{N} B_i u_i(k) \tag{6-6}$$

with ith output equation

$$y_i(k) = c_i x(k) \tag{6-7}$$

for $i = 1, \ldots, N$. Assume now that each local output can be used to activate the ith local fuzzy controller given by a structure and a set of rules given below:

$$u_i(k) = \mathrm{FLC}_i(y_i) \tag{6-8}$$

$$R^i\text{: IF } y_i(k) \text{ is } A^i \text{ and } \Delta y_i(k) \text{ is } B^i \text{ THEN } u_i(k) \text{ is } C^i \tag{6-9}$$

where k can be changed from k to k-*1*, k-*2*, ...

Figure 6-8 shows a decentralised fuzzy control architecture. Here each local control signal will be the fuzzified output of rules of type given by (6-9).

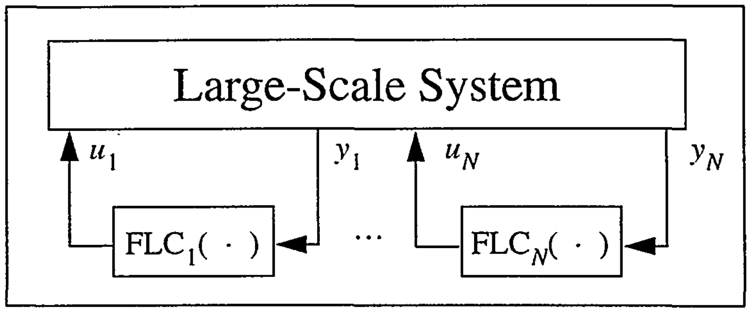


Figure 6-8: A decentralized fuzzy control architecture

An alternative architecture for a decentralised fuzzy control of a large scale system is shown in Figure 6-9. Here, typical local controller would be described by the following relation

$$u(t) = K_p e(t) + K_I \int e(\tau)d\tau + K_D \dot{e}(t)$$

and the fuzzy tuner's typical rules can be as

IF e is A^l and Δe is B^l THEN K_P is Small and K_I is Medium and K_D is Big

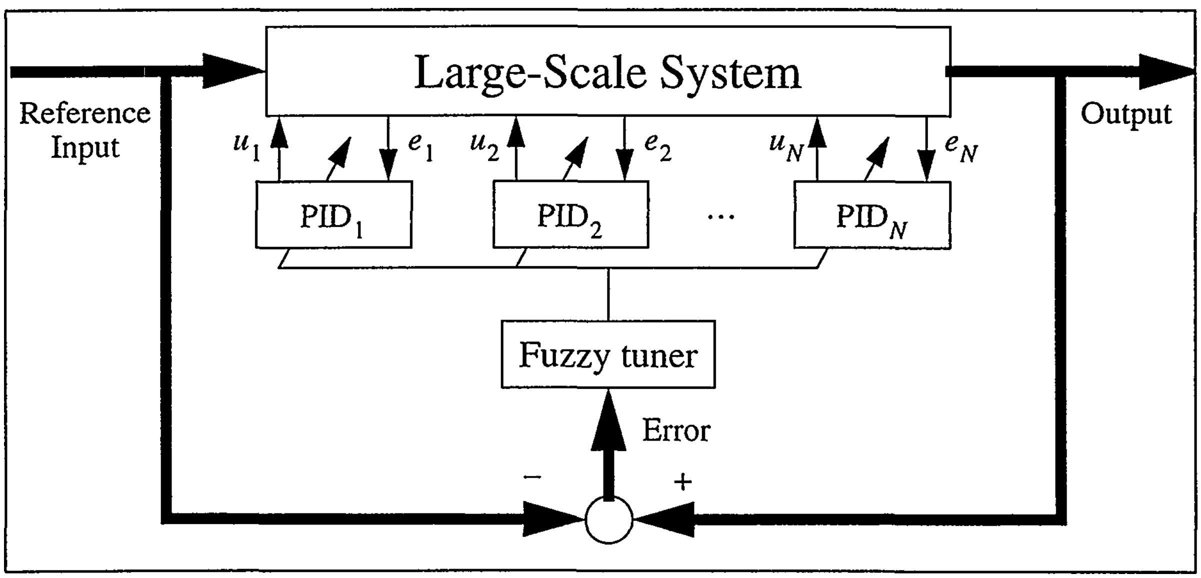


Figure 6-9: An architecture for fuzzy-tuned decentralised PID control of large-scale system.

6.3 Interaction Predicted Fuzzy Control

One of the most attractive approaches for hierarchical control has been the interaction prediction method (see Section 4.3 of Jamshidi, 1996, and Chapter 4 of Jamshidi, 1983). One reason for the attractiveness of this method is the simplicity of the coordinator's problem, while reducing the overall system's order to a finite number of sub-problems. In spite of these favourable attributes the fact remains that at the low-level, a series of computationally intensive optimal control problems must be solved.

The object of this section is to introduce fuzzy logic into the interaction prediction approach of hierarchical control. The reader is cautioned that the content of this section is merely a proposal for further detailed research and investigation.

Consider a large-scale discrete-time linear system

$$x(k+1) = Ax(k) + Bu(k) \tag{6-10}$$

$$x(0) = x_0 \tag{6-11}$$

where all the terms are defined before through the usual optimal control theory (Jamshidi, 1996). Assume that the order of (6-10) is very large, i.e. $n \gg 1$ and let it be decomposed into N subsystems each being described by,

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k) + z_i(k) \tag{6-12}$$

$$x_i(0) = x_{i0} \tag{6-13}$$

where the order of (6-12) is now n_i with constraint $n = \sum_{i=1}^{N} n_i$. The interaction term $z_i(k)$ is assumed to follow the relation

$$z_i(k) = \sum_{\substack{j=1 \\ i \neq j}}^{N} A_{ij} x_j(k) \tag{6-14}$$

and the overall Hamiltonian of the problem is given by:

$$H_i(k) = \frac{1}{2} x_i^T(k) Q_i x_i(k) + \frac{1}{2} u_i^T(k) R_i x_i(k) + \alpha_i^T z_i(k) \tag{6-15}$$

$$- \sum_{j \neq i}^{N} \alpha_i^T A_{ji} x_i(k) + P_i^T(k)\,(A_i x_i(k) + B_i u_i(k) + z_i(k))$$

and $\alpha_i(k)$ is the $n \times 1$ vector of Lagrange multipliers corresponding to the interconnection constraints in (6-14). The remaining terms are self explanatory. It is also well known (Jamshidi, 1983) that the coordinator's policy (interaction balance) is given by

$$\begin{bmatrix} z_i(k) \\ \alpha_i(k) \end{bmatrix}^{l+1} = \begin{bmatrix} -P_i^T(k) \\ \sum_{j=1}^{N} A_{ji} x_j(k) \end{bmatrix}^{l} \tag{6-16}$$

The subsystem problem consists of a discrete-time Riccati solution to (6-12) with known $z_i(k)$ and a quadratic cost function (Jamshidi, 1983), i.e.

$$u_i(k) = -F_i(k) x_i(k) \tag{6-17}$$

where $F_i(k)$ is related directly to $K_i(k)$ which is the symmetric $n_i \times n_i$ positive definite Riccati matrix in discrete-time form.

In order to solve the low-level control problem with fuzzy logic, let us introduce the following set of rules for the ith subsystem:

$$R_i\text{: IF } x_i(k) \text{ is } A_1^i \text{ and } x_i(k) \text{ is } A_2^i \text{ .... THEN } u_i(k) = -F_i(k) x(k) \tag{6-18}$$

where A^i_j, $j = 1, \ldots$ are the ith subsystem's vectors of fuzzy labels and $F_i(k)$ is an $m_i \times n_i$ feedback matrix which can be determined arbitrarily for now. Note that the consequents of (6-18) could also be linguistic such as those rules in (6-1). Then $u_i(k)$ is *NM*, where *NM* stands for Negative Medium. Note also, that the rule set (6-18) represents n_i rules for the ith subsystem. Having observed this if there are, say 5 linguistic labels per fuzzy set A^i_j, then the size of the rule set would be 5^{n_i}, where n_i is usually a large number, e.g. 5, 10, or even 15. This indicates that the number of rules per sub-system can become too large. Figure 6-10 shows a block diagram for the proposed interaction prediction fuzzy control system. The coordinator's problem can be simply the upper half of the relation (6-16). As an alternative one can use fuzzy rules in place of the above relation, e.g

$$\begin{array}{ll} \text{IF } (x_1 \cap x_2 \ldots \cap x_N) \text{ is } A_1 \text{ THEN } & z_1^{l+1} = z_1^l \\ \vdots & \vdots \\ \text{IF } (x_1 \cap x_2 \ldots \cap x_N) \text{ is } A_N \text{ THEN } & z_N^{l+1} = z_N^l \end{array} \tag{6-19}$$

Note that the large-scale hierarchical fuzzy control architecture in Figure 6-10 provides a 3-directional rule-based reduction and simplification of control implementation - decomposition, sensory fusion, and rule-base hierarchy.

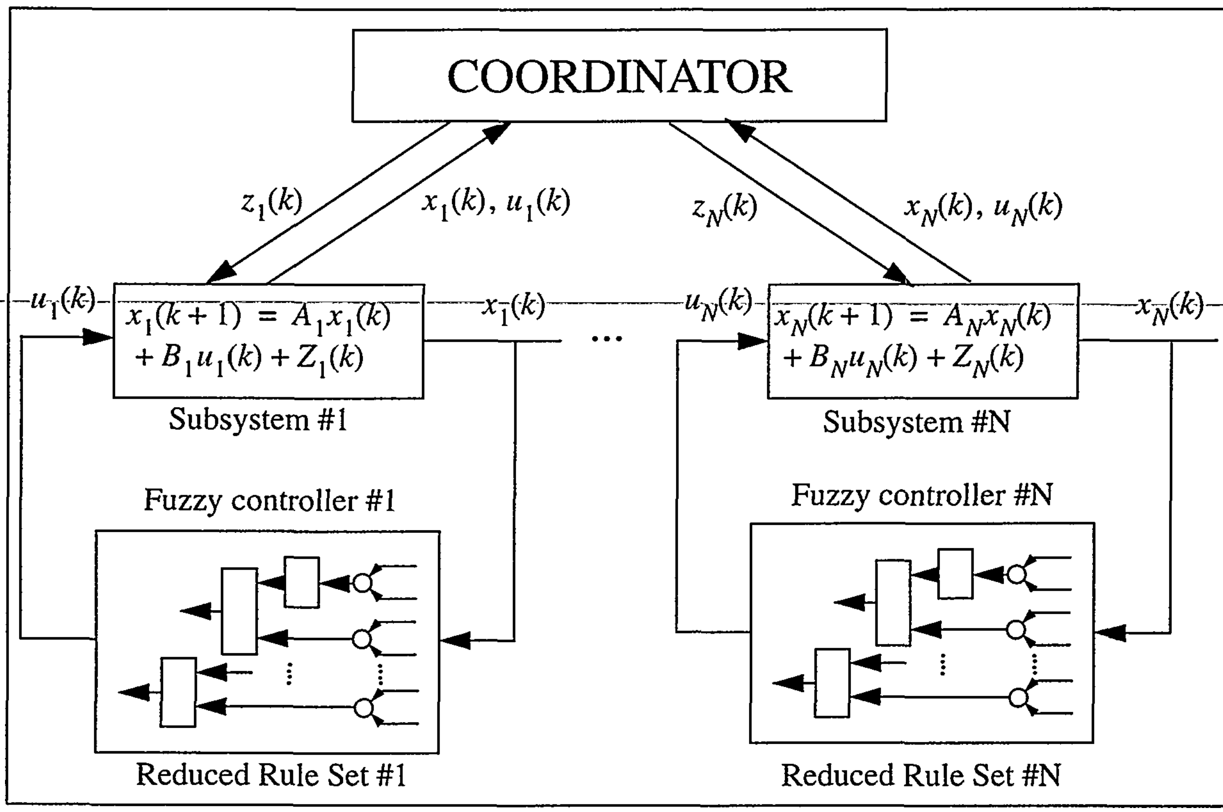


Figure 6-10: A proposed interaction predicted hierarchical fuzzy control architecture

7 Discussions and Conclusions

In this study a water desalination plant's brine heater subsystem under conventional and fuzzy control has been investigated. Our goals in this study was to implement a stable FLC, and evaluate performance by comparing it with three conventional control strategies. We will in this chapter use the performance criterion, Integral of Absolute Error (IAE), to do a fair comparison. Then we give some recommendations on further work from this study. Final remarks ends this chapter and the report.

7.1 Fuzzy vs Conventional

'Best' performance is a subjective measurement, so we choose the numerical criterion IAE = $\int |e(\tau)| d\tau$ to be able to compare the control strategies. (Babcock Inc., 1994) conclude with that the cascade controller is the best performing strategy, however they did not use any performance criteria to reach this conclusion, but based their conclusions on heuristic argumentation. By comparing the performance criterion IAE, in Figure 7-1, our conclusion is that the PID controller is the best performing of the three conventional strategies.

The comparison between the FLC and conventional methods is not meant to be taken 'literally', but rather as to be able to assess the performance of the FLC. No attempts has been made to optimise the conventional strategies, these are taken 'as is'. Still, the comparison can said to be fair, as the FLC was not optimised either and has one degree of freedom less than the PID. Our conclusion is then, from looking at Figure 7-1 (with IAE on y-axis), that **the FLC performs in most simulation cases better or equally well as the conventional strategies.** The superior performance is most clear in Simulation D.

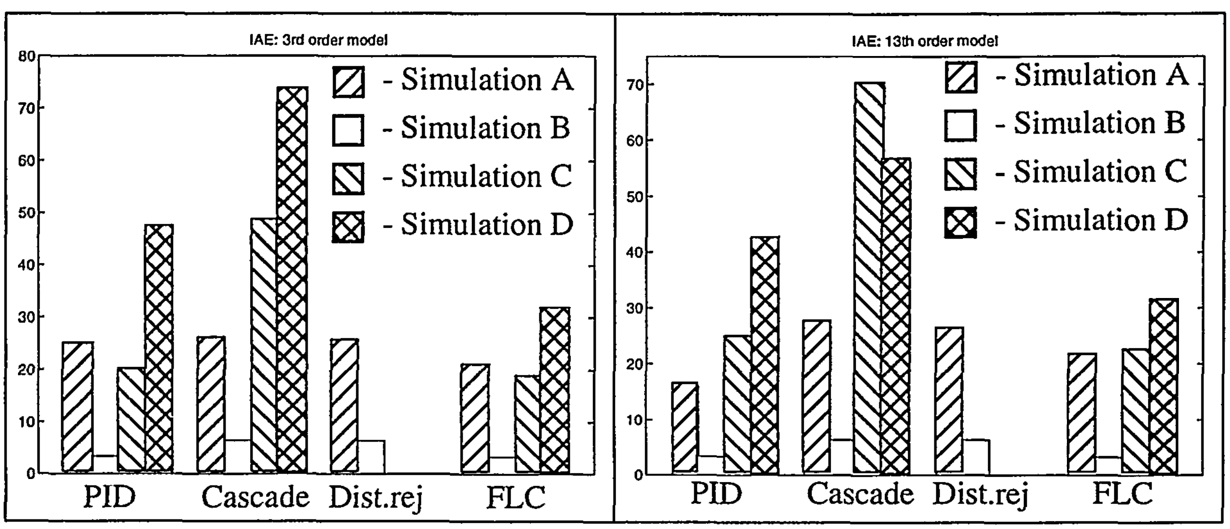


Figure 7-1: Performance criteria IAE for 4 controllers.

7.2 Stability

We have shown (in Section 5.4 'Stability of the Fuzzy Control System' on page 40) that the fuzzy control system is stable. Looking at the Nyquist plot one could be misled to believe that the controller has almost infinite gain margin and a phase-margin of about 45 degrees. But in fact, none of these properties can be deduced from the Nyquist plot as in the linear case. This is because changing the gain of the controller (K_u) will lead to a different X_{mod} to the model. We will then have a different SIDF of the model, which in turn leads to a different L_{OLS}, which can be unstable. We can therefore only conclude that **the fuzzy control system is stable for the chosen TSFLC with the studied amplitudes and frequencies.**

7.3 Recommendations

We will here come with some suggestions on what could be done in a further study.

7.3.1 Model

- The validity of using a 12th order model vs. a 3rd order model should be assessed.
- Model parameters must be evaluated and probably changed to obtain a model which can be simulated over larger area of operating conditions. The model should work in the operating regime as described in Section 2.3.4 'Operating Conditions' on page 11.
- The model does not take into consideration the main property that distinguishes the brine heater from a normal heat-exchanger, namely the dependency between the brine heater inlet temperature and the TBT. In our opinion this is clearly a deficiency, as it changes the overall dynamic behaviour of the system, and thus also the performance of the controllers. The possibility of including a relation showing this dependency, should be investigated.
- Work can be done to simulate the controller strategies with a total non-linear model of an MSF plant to get more realistic responses.
- Model identification and verification could be done with actual plant measurements.

7.3.2 Control

Conventional

- Optimization of the PID control parameters (the PID and cascade control scheme) is clearly needed. The open-loop methods are just meant to be a starting point for further controller tuning.
- In (Babcock Inc., 1994) they discuss the poor performance of the disturbance rejection scheme, and names 1) the lack of time delays and inertias in the brine heater model, and 2) the chosen disturbance rejection function. One could there-

fore find a better disturbance rejection function than the linear used in (Babcock Inc., 1994), and/or introduce reasonable time delays in the model.

- The problem with the cascade control loop seems to be the integral term of the inner loop. Because one has saturation in the valve actuator, the integrator will continue integrating, and the outer loop will produce a still higher setpoint due to the lack of response from the inner loop.This phenomena is known as integrator wind-up. An anti wind-up controller can be used, solving part of this problem.

Fuzzy

- Optimization of scaling factors and tuning of fuzzy sets on existing controller. The controller could also be expanded with more fuzzy sets. A better performance could be expected if one would tune the rules and the membership functions to better reflect the non-linearities in process. This could for instance be done using Genetic Algorithms as a search engine to find a global optimum.
- Hybrid approach. In the cascade control loop, one could investigate the use of an FLC in the secondary loop. The non-linearities and uncertainties lies mainly in the valve characteristics (Babcock Inc., 1994).

In this study we have only looked at one operating point. A good control strategy should be able to work in a larger range of operation. The non-linearities of the process is more predominant, and demands more robustness of the control, when all of the operating regime is considered. Therefore several operating points should be included in a further study (as mentioned before, this requires a model which works for the operating regime).

7.4 Final Remarks

Fuzzy logic offers an elegant method of transferring knowledge of a process into a non-linear controller. One interesting aspect, which (to the authors best knowledge[1]) has not been investigated, is the following:

- A conventionally obtained control law can be transferred to a fuzzy system with any desired degree of accuracy, since fuzzy systems are universal approximators (Kosko, 1992). The resulting fuzzy rules then describes in a natural, linguistic way the underlying control law. The rulebase could then be modified, using heuristic process and control knowledge, enhancing the performance of the controller.

Fuzzy control is still immature in many aspects, e.g stability, design and tuning. One reason for this has been the reluctance from the control community. Hopefully, with increased acceptance from control researchers, some of the drawbacks of fuzzy control will be resolved. Fuzzy control should not be seen upon as a replacement of classic control theory, but rather as an extension in the available tools for a control engineer.

1. in (Dubois *et al.*, 1994) they also mention this idea.

Bibliography

M. R. Akbarzadeh and M. Jamshidi, 1994:"Fuzzy hierarchical control of flexible arms". *Proc. Fuzz-IEEE*, Orlando, Florida.

D. M. K. Al-Gobaisi, A. Hassan, G. P. Rao, A. Sattar, A. Woldai, and R. Borsani, 1994a: "Towards improved automation for desalination processes, Part 1: Advanced control". *Desalination*, **97**, pp. 469-506.

D. M. K. Al-Gobaisi, 1994b: "A quarter-century of seawater desalination by large multistage flash plants in Abu Dhabi". *Desalination*, number unknown.

J. Aracil, A. Garcia-Cezero, A. Barreiro and A. Ollero, 1988: "Stability Analysis of Fuzzy Control Systems. A Geometrical Approach." Ed. C. A. Kulikowski, R. M. Huber (Hrsg). *AI, Expert Systems and Languages in Modelling and Simulation.* North-Holland, pp. 323-330.

J. Aracil, A. Garcia-Cezero and A. Ollero, 1989: "Stability Indices for the Global Analysis of Expert Systems". *IEEE Transactions on Systems, Man, and Cybernetics*, **19**, pp. 998-1007.

J. Aracil, A. Garcia-Cezero, A. Barreiro and A. Ollero, 1991: "Fuzzy control of dynamical systems: stability analysis based on the conicity criterion". *Proc. IFSA'91.* Vol. Engineering, Brussel, pp. 5-8.

R. Babuska, 1994: "Designing fuzzy control systems with MATLAB and SIMULINK". *Proc. EUFIT-94*, Aachen, Germany.

Babcock Inc., 1994: *Optimisation for maintenance and operation of desalination plants in Abu Dhabi, Annex M: Instrumentation and control concepts.* Report made for Water and Electricity Department in Abu Dhabi, the United Arab Emirates.

A. Barriero and J. Aracil, 1992: "Stability of uncertain dynamical systems". *Proc. IFAC Symp. on AI in Real-Time control*, Delft, The Netherlands, pp. 177-182.

F. Bouslama and A. Ichikawa, 1992: "Application of limit fuzzy controllers to stability analysis". *Fuzzy Sets and Systems*, **49**, pp. 103-120.

M. Braee and D. A. Rutherford, 1978: "Selection of parameters for a fuzzy logic controller". *Fuzzy Sets and Systems*, **2**, pp. 185-199.

M. Braee and D. A. Rutherford, 1979: "Theoretical and linguistic aspects of the fuzzy logic controller". *Automatica*, **15**, pp. 553-577.

G. Bretthauer and H. P. Opitz, 1994: "Stability of Fuzzy Systems". *Proc. EUFIT-94*, Aachen, Germany.

H. Buhler, 1993: "Stabilitätsuntersuchung von Fuzzy-Regelungsystemen". *Proc. 3. Workshop Fuzzy Control des GMA-UA.* 1. 4. 1., Dortmund, Germany, pp. 1-12.

R. Böhm, 1992: "Ein ansatz zur stabilitätsanalyse von fuzzy-reglern". *Forschungsberichte Universität Dortmund, Fakultät fur Elektrotechnik, Band Nr. 3,2. Workshop Fuzzy des GMA-UA.* 1. 4. 2., Dortmund, Germany, pp. 24-35.

C. L. Chen, P. C. Chen and C. K. Chen, 1993: "Analysis and design of a fuzzy control system". *Fuzzy Sets and Systems*, **57**, pp. 125-140.

Y. Y. Chen, 1987: "Stability analysis for fuzzy control - A Lyapunov approach". *Proc. IEEE Annual Conference Systems, Man, and Cybernetics.* **3**, pp. 1027-1031.

Y. Y. Chen, 1988: "The analysis of fuzzy dynamic systems using cell-to-cell mapping.". *Proc.*

IEEE Annual Conference Systems, Man, and Cybernetics.

Y. Y. Chen and T. C. Tsao, 1989: "A description of the dynamical behaviour of fuzzy systems.". *IEEE Transactions on Systems, Man, and Cybernetics*, **19**, pp. 745-755.

G. Conte, M. Bonifazio and L. Silvestri, 1994: "Fuzzy supervised PID controllers". *Proc. EUFIT-94*, Aachen, Germany.

Comsol Inc.: *MC user's guide*. Comsol Inc., Stockholm, Sweden. (Year unknown)

B. Demaya, S. Boverie and A. Titli, 1994: "Stability Analysis of Fuzzy Controllers via cell-to-cell root locus analysis". *Proc. EUFIT-94*, Aachen, Germany.

D. Dubois, H. Prade and M. Grabisch, 1994: "Chapter 7. Gradual Rules and the Approximation of Control Laws". In *Theoretical Aspects of Fuzzy Control*, ed. H. T. Nguyen et al. John Wiley & Sons. ISBN unknown.

D. Driankov, H. Hellendoorn and M. Reinfrank, 1993: *An Introduction to Fuzzy Contro*l. Springer Verlag, Berlin, ISBN 3-540-56362-8.

M. H. A. El-Saie and M. H. E. Hafez, 1994: "Selecting and tuning the control loops of MSF desalination for robustness". *Desalination*, **97**, pp. 529-540.

D. Franke, 1993: "Fuzzy control with Lyapunov Stability". *Proc. European Control Conference*, Groningen, Germany.

J. Gertler and H. W. Chang, 1986: "An instability indicator for expert control". *IEEE Transactions on Control Systems*, Heft, pp. 14-17.

H. S. Han and J. G. Lee, 1994: "Necessary and sufficient conditions for stability of time-varying discrete interval matrices". *Int. Journal of Control*, **59**, pp. 1021-1029.

W. T. Hanbury, T. Hodgkiess and R. Morris, 1993: *Desalination Technology '93: An Intensive Course*. Porthan Ltd., Glasgow, U. K.

C. J. Harris and C. Moore, 1990: "Real time fuzzy based self-learning predictors and controllers". *Proc. IFAC 11th Triennial world congress*, Tallinn, Estonia, USSR.

T. Hojo, T. Terano and S. Masui, 1986: "Stability analysis of Fuzzy Control Systems", *IEEE Transactions on Systems, Man, and Cybernetics*, **16**, pp. 576-582.

C. D. Hornburg, A. Radif and H. Sulton: "A Large Single Train MSF Plant. Part I: Design". *Desalination & Water Reuse Quarterly*, **2**, pp. 22-30, (Year unknown)

A. Husain, K. V. Reddy, A. Woldai and D. M. K. Al-Gobaisi, 1995: "Modelling, simulation and optimization of an MSF desalination plant". To be published.

A. Husain, A. Woldai, A. Al-Radif, A. Kesou, R. Borsani, H. Sultan and P. B. Deshpandey, 1994: "Modelling and simulation of a multistage flash (MSF) desalination plant". *Desalination*, **97**, pp. 555-586.

G. C. Hwang and S. C. Liu, 1992: "A stability approach to fuzzy control design for nonlinear systems". *Fuzzy Sets and Systems*, **48**, pp. 279-287.

IEEE Control Systems Magazine, 1993: Letters to the editor. *IEEE Control Systems Magazine*, **13**.

M. Jamshidi, 1983: *Large-Scale Systems - Modelling and Control.* Elsevier, New York.

M. Jamshidi, M. Tarokh and B. Shafai, 1992: *Computer-aided Analysis and Design of Linear Control Systems.* Prentice-Hall, Englewood Cliffs, U. S. A.

M. Jamshidi, N. Vadiee and T. J. Ross, 1993: *Fuzzy logic and control. Software and hardware* *applications. Volume 2*. Prentice-Hall, Englewood Cliffs, U. S. A, (1993). ISBN 0-13-368986-7.

M. Jamshidi and A. Titli, 1995: "Stability of fuzzy control systems via interval matrix method". *Proc. 3rd IEEE Mediterranean Symp. on New Directions in Control and Automation*, Cyprus.

M. Jamshidi, 1996: *Large-scale systems-Modelling, Control and Fuzzy logic*. Prentice-Hall, Englewood Cliffs, U. S. A. (To appear).

T. A. Johansen, 1995: "On the optimality of the Takagi-Sugeno-Kang Inference Mechanism". *IEEE Symposium on Fuzzy Systems*. Yokohama, Japan.

H. Kang, 1992: "Stability and Control of Fuzzy Dynamic Systems via Cell-State Transitions in Fuzzy Hypercubes". *IEEE Transactions on Fuzzy Systems,* **1**, pp. 267-279.

A. H. Khan, 1986: *Desalination processes and multi stage flash distillation practice*. Elsevier Science Publishers, Amsterdam. ISBN 0-444-42563-2.

W. J. M. Kickert and H. R. van Nauta Lemke, 1976: "Application of a fuzzy controller in a warm water plant". *Automatica*, **12**, pp. 301-308.

W. J. M. Kickert and E. H. Mamdani, 1978: "Analysis of a fuzzy logic controller". *Fuzzy Sets and Systems,* **1**, pp. 29-44.

P. J. King and E. H. Mamdani, 1977: "The application of fuzzy control systems to industrial processes". *Automatica*, **13**, pp. 235-242.

J. B. Kiszka, M. M. Gupta and P. N. Nikiforuk, 1985: "Energistic Stability of Fuzzy Dynamic Systems, *IEEE Transactions on Systems, Man, and Cybernetics*, **15**, pp. 783-792.

B. Kosko, 1992: "Fuzzy systems as universal approximators". *Proc. of the First IEEE conf. on Fuzzy Systems*, San Diego, USA. pp. 1153-1162.

R. Kruse, J. Gebhardt and F. Klawonn, 1994: *Foundations of fuzzy systems*. John Wiley & Sons, Chichester West Sussex. ISBN 0-471-94243-X.

G. Langari and M. Tomizuka, 1990: "Stability of fuzzy linguistic control systems". *Proc. IEEE Conf. Decision and Control*, Hawaii, pp. 2185-2190.

P. M. Larsen, 1980: "Industrial application of fuzzy logic control". *Intl. J. Man. Mach. Studies*, **12**, pp. 3-10.

C. C. Lee, 1990: "Fuzzy logic in control systems: fuzzy logic controller-Part I,II". *IEEE Transactions on Systems, Man, and Cybernetics*, **20**, pp. 404-435.

E. H. Mamdani, 1974: "Application of Fuzzy Algorithm for Control of Simple Dynamic Plant". *Proc. IEE*, **121**, pp. 1585-1588.

J. P. Marin and A. Titli, 1994: "Comparative analysis of stability methods for fuzzy controllers". *Proc. EUFIT-94.* Aachen, Germany.

J. P. Marin and A. Titli, 1995: "On the use of describing functions to analyse and design fuzzy controllers". To appear at *IFSA-95*, Sao-Paulo, Brazil.

The MathWorks Inc., 1993: *SIMULINK User's Guide*. The MathWorks, Inc., U. S. A.

H. P. Opitz, 1993: "Fuzzy Control. Teil 6: Stabilität von Fuzzy-Regelungen". *Automatisierungstechnik*, **41**, A21-24.

H. P. Opitz, 1994: "Stability analysis and fuzzy control". *Proc. Fuzzy Duisburg '94, Intern. Workshop on Fuzzy Technologies in Automation and Intelligent Systems*, Duisburg, Germany.

D. W. Patterson, 1990: *Introduction to artificial intelligence and expert systems.* Prentice-Hall International Inc.,Englewood Cliffs, U. S. A. ISBN-0-87692-777-0.

W. Pedrycz, 1993: *Fuzzy Control and Fuzzy Systems.* Research Studies Press, Taunton, England.

G. P. Rao, D. M. K. Al-Gobaisi, A. Hassan, A. Kurdali, R. Borsani and M. Aziz, 1994: "Towards improved automation for desalination processes, Part 2: Intelligent control". *Desalination*, **97**, pp. 507-528.

G. U. Raju, J. Zhou and R. A. Kisner, 1991: "Hierarchical fuzzy control". *Int. Journal of Control*, **54**, pp. 1201-1216.

K. S. Ray and D. D. Majumder, 1984: "Application of circle criteria for stability analysis associated with fuzzy logic controller". *IEEE Transactions on Systems, Man, and Cybernetics*, **14**, pp. 345-349.

K. S. Ray, S. G. Ananda and D. D. Majumder, 1984: "L-Stability and the related design concept for SISO linear systems associated with fuzzy logic controller". *IEEE Transactions on Systems, Man, and Cybernetics*, **14**, pp. 932-939.

B. Sayyarrodsari and A. Homaifar, 1995: "On the hierarchy of fuzzy control systems". submitted to *IEEE Transactions on Fuzzy Systems.*

J. S. Shamma and M. Athans, 1990: "Analysis of gain scheduled control for nonlinear plants". *IEEE Transactions on Automatic Control*, **35**, pp. 898-907.

F. Sheikholeslam, 1994: *Stability analysis of nonlinear and fuzzy systems.* M. Sc. Thesis, Department of EECS Isfahan University of Technology, Isfahan, Iran.

M. Sugeno and M. Nishida, 1985: "Fuzzy control of model car". *Fuzzy Sets and Systems*, **16**, pp. 103-113.

T. Takagi and M. Sugeno, 1985: "Fuzzy identification of systems and its application to modelling and control". *IEEE Transactions on Systems, Man, and Cybernetics*, **4**, pp. 116-132.

K. Tanaka and M. Sugeno, 1992: "Stability analysis and design of fuzzy control systems". *Fuzzy Sets and Systems*, **45**, pp. 135-156.

V. Tahani and F. Sheikholeslam, 1994: "Stability of Autonomous Fuzzy Control Systems", *Proc. EUFIT-94*, Aachen, Germany.

L. X. Wang, 1993: "Stable adaptive fuzzy control of nonlinear systems". *IEEE Transactions on Fuzzy Systems*, **1**, pp. 146-155.

L. X. Wang, 1994a: "A supervisory controller for fuzzy control systems that guarantees stability". *Proc. IEEE Conf. SMC*, pp. 1035-1039.

L. X. Wang, 1994b: *Adaptive Fuzzy Systems and Control.* Prentice Hall, Englewood Cliffs, NJ.

P. Z. Wang, H. M. Zhang and W. Xu, 1990: "Pad-analysis of fuzzy control stability". *Fuzzy Sets and Systems*, **38**, pp. 27-42.

S. Yasunobu and T. Hasegawa, 1985: "Automatic train operation by predictive fuzzy control". In *Industrial Application of Fuzzy Control.* M. Sugeno ed. Amsterdam, North-Holland.

Z. M. Yeh, 1994: "A performance approach to fuzzy control design for nonlinear systems". *Fuzzy Sets and Systems*, **64**, pp. 339-352.

L. A. Zadeh, 1965: "Fuzzy sets". *Information and control*, **8**, pp. 338-353.

L. A. Zadeh, 1973: "Outline of a new approach to the analysis of complex systems and decision processes". *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-3**, pp. 28-44.

L. A. Zadeh, 1994: "The Role of Fuzzy Logic in Modelling, Identification, and Control". *Modelling, Identification and Control*, **15**, pp. 191-203.

L. A. Zadeh, 1995: "Soft Computing and Fuzzy Logic". To be published.

Z. Y. Zhao, M. Tomizuka, and S. Isaka, 1993: "Fuzzy gain scheduling of PID controllers". *IEEE Transactions on Systems, Man, and Cybernetics*, **23**, pp. 1392-1398.

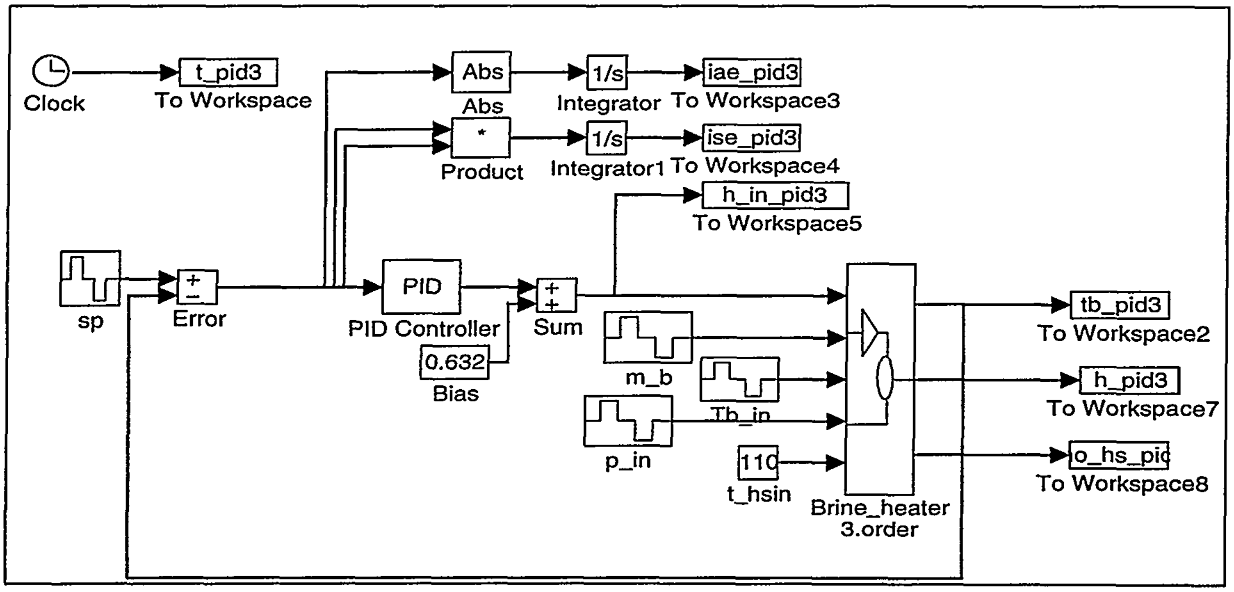
K. Zinser and R. Schreiber, 1993: "Fuzzy logic-an innovative technology holding great promise for automation systems". *ABB Review*, **3**, pp. 13-20.

K. J. Åström, J. J. Anton and K-E. Årzén, 1986: "Expert control". *Automatica*, **22**, pp. 277-286.

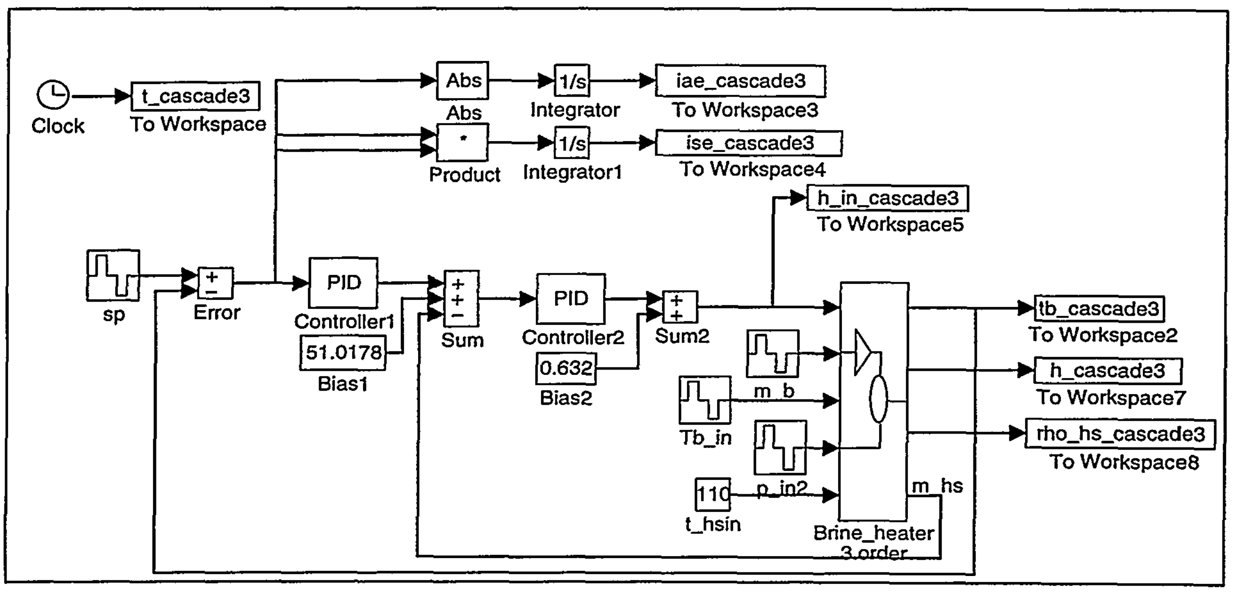
K. J. Åström and T. Hägglund, 1988: *Automatic tuning of PID controllers*. Instrument society of America, ISBN-1-55617-081-5.

K. J. Åström and B. Wittenmark, 1989: *Adaptive control*. Addison-Wesley, Reading, MA, ISBN 0-201-09720-6.

A.1 brine_h_pid.m[1]

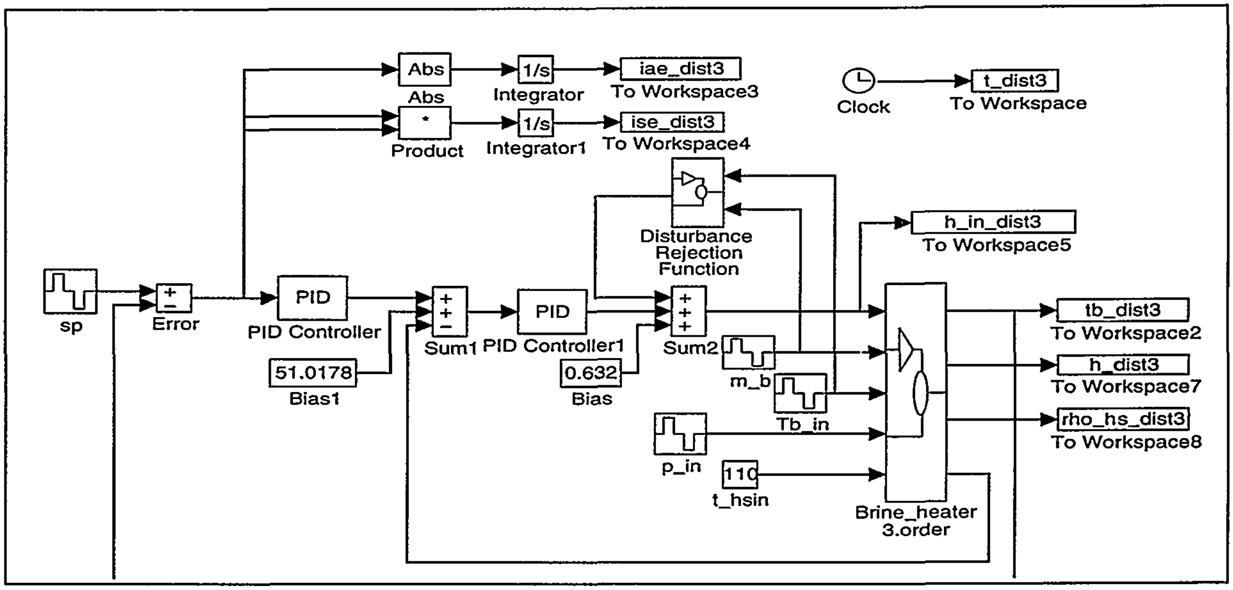


A.2 brine_h_cascade.m[2]



1. **brine_h12_pid.m** is the same, except it uses the 12th order **brineheatermex.c.**
2. **brine_h12_cascade.m** is the same, except it uses the 12th order **brineheatermex.c.**

A.3 brine_h_dist.m[1]



A.4 sugeno.m[2]

```
function out=sugeno(x1,x2,Mem_x1,Mem_x2,Mem_u,rules)
%This is the Takagi-Sugeno type FLC.
%Inputs are:
%       x1, x2: Crisp inputs
%       Mem_x1,
%       Mem_x2: Membership functions of the fuzzy sets
%               for the inputs.
%       Mem_u:  crisp constant for output consequence
%       rules:  rulebase matrix

%determine the number of input sets for x1 and x2
[m,n]=size(rules);

%Find membership value for all fuzzy sets for
%the input x1
for i=1:m,
        a=Mem_x1(i,1);
        b=Mem_x1(i,2);
        c=Mem_x1(i,3);

        if (x1==b),
                ux1=1;
        elseif ((x1>=b) & (x1<=c)),
                ux1=(x1-c)/(b-c);
        elseif ((x1<b) & (x1>=a)),
                ux1=(x1-a)/(b-a);
        else
                ux1=0;
        end;
        u1(i)=ux1;
end;

%Find membership value for all fuzzy sets for
%the input x2
for i=1:n,
        a=Mem_x2(i,1);
        b=Mem_x2(i,2);
        c=Mem_x2(i,3);

        if (x2==b),
                ux2=1;
```

1. **brine_h12_dist.m** is the same. except it uses the 12th order **brineheatermex.c**
2. The **sugeno.c** file obtained from the translation with MC is not included here as it is not very readable

```
        elseif ((x2>=b) & (x2<=c)),
                ux2=(x2-c)/(b-c);
        elseif ((x2<b) & (x2>=a)),
                ux2=(x2-a)/(b-a);
        else
                ux2=0;
        end;
        u2(i)=ux2;
end;

%Initialise sums
SigUijCij=0;
SigUij=0;

%for all fuzzy sets for x1
for i=1:m,
        %for all fuzzy sets for x2
        for j=1:n,
                %use min-operation for fuzzy inference
                %and save in array
                if u1(i)<u2(j),
                        u(i,j)=u1(i);
                else
                        u(i,j)=u2(j);
                end;

                %calculate the firing strength
                SigUijCij=SigUijCij+Mem_u(rules(i,j))*u(i,j);
                SigUij=SigUij+u(i,j);
        end;
end;

%find weighted output
out=SigUijCij/SigUij;
```

A.5 mem3.m

```
%Error has three fuzzy sets defined as:
mem_func_e=[    -1 -1  0        % NEG
                -1  0  1        % ZE
                 0  1  1];      % POS

%The fuzzy sets of e_dot are the same as e
mem_func_edot=mem_func_e;

%Three singletons for output function
mem_func_output=[-1 0 1];       %[NEG, ZE, POS]

%Rulebase
rulebase=[      1 1 2
                1 2 3
                2 3 3];

%Scaling factors
Ku=0.0375;
Ke=1/0.8004;    %1.25
Kedot=1/0.15;   %6.67
```

A.6 brineheatermex3.c

```
#include <math.h>
#include "cmex.h"

#define NSTATES 3
#define NOUTPUTS 4
#define NINPUTS 5
#define NEEDINPUTS 0

#define pi      3.14
#define rho_b   1000    /*Density of brine*/
#define cp_b    4.18    /*Heat capacity of brine*/
#define n_tube  2700    /*Number of tubes in heat exchanger*/
#define d_i     0.0293  /*Internal diameter of tubes*/
#define L       17.33   /*Length of tubes*/
#define k       1.942   /*Heat transfer coefficient*/
#define eps     0.001   /*Valve constant*/
```

```
#define rho      0.7      /*Density of supply steam*/
#define k_vs     7        /*Valve coefficient*/
#define H_v      2230     /*Evaporation enthalpy*/
#define R        0.00461  /*Ideal gas constant*/
#define c2K      273      /*From Celsius to Kelvin*/

static double T_cond;
static double m_hs;

void derivatives(t,x,u,dx)
double t,*x,*u;
double *dx;
{
double h_in,m_b,Tb_in,p_in,T_hsin;
double h,rho_hs,A_i,T_b,T_ln,d_Tb;
double phi,V_hs,V_b,Tb,tmp,p_sh,Q_hs,m_cond;
int i;

        /*Inputs*/
        h_in=u[0];       /*Control signal to steam valve*/
        m_b=u[1];        /*Recycle flow*/
        Tb_in=u[2];      /*Inlet temperature*/
        p_in=u[3];       /*Steam supply pressure to valve */
        T_hsin=u[4];     /*Temperature of saturated steam*/

        /*States*/
        Tb=x[0];         /*TBT*/
        h=x[1];          /*Valve opening*/
        rho_hs=x[2];     /*Density of steam*/

        /*Steam pressure in heater*/
         p_sh=rho_hs*R*(T_cond+c2K);

        /*Steam condensation temperature*/
        T_cond=(-1750.268/(log10(p_sh/1.332e-3)-8.1077)-235);

        /*Heat Transfer surface*/
        A_i=pi*n_tube*d_i*L;

        /*Volume of brine*/
        V_b=0.25*pi*n_tube*L*d_i*d_i;

        /*Logarithmic temperature, T_ln*/
        T_ln=(Tb-Tb_in)/log((T_cond-Tb_in)/(T_cond-Tb));

        /*Q_hs*/
        Q_hs=k*A_i*T_ln;

        /*Energy balance*/
        d_Tb=(Q_hs+m_b*cp_b*(Tb_in-Tb))/(rho_b*cp_b*V_b);

        dx[0]=d_Tb;

        /*-------------------------------*/
        /*Valve equations                */
        /*-------------------------------*/

        if (fabs(h-h_in)<eps){
                dx[1]=0;                /*d_h=0;*/
        }
        else{
                if (h<h_in){
                        dx[1]=0.02;     /*d_h=0.02;*/
                }
                else{
                        dx[1]=-0.02;    /*d_h=-0.02;*/
                }
        }
        /*Saturation limit for valve opening */
        if(h>1){
                dx[1]=-0.02;
                h=1;
        }
        /*-------------------------------*/
        /*Steam density equations        */
        /*-------------------------------*/

        /*PHI-equal percentage valve equation*/
        /*The ratio of k_vo/k_vs is about 0.04*/
        phi=0.04*exp(3.22*h);

        /*PHI-linear characteristic valve equation*/
        /*phi=0.04+0.96*h;*/

        /*Flow characteristic*/
        /*The value of c_G is 519 */
        m_hs=k_vs*519*sqrt(fabs(p_sh*(p_in-p_sh)*rho/T_hsin))*phi;
```

```
        /*Flow of condensate */
        m_cond=Q_hs/H_v;

        /*Steam density*/
        V_hs=4.07*V_b;
        dx[2]=(m_hs-m_cond)/V_hs;

}

void init_conditions(x0)
double *x0;
{
        x0[0]=95;
        x0[1]=0.632;     /*For equal percentage valve*/
        x0[2]=0.706;
        T_cond=105.4;
        m_hs=51.04;
}

void outputs(t,x,u,y)
double t,*x,*u;
double *y;
{
        y[0]=x[0];       /*TBT*/
        y[1]=x[1];       /*Valve opening*/
        y[2]=x[2];       /*Steam density*/
        y[3]=m_hs;       /*Steam flow*/
}
#include "simulink.h"
```

A.7 brineheatermex.c

```
#include <math.h>
#include "cmex.h"

#define NSTATES 12
#define NOUTPUTS 13
#define NINPUTS 5
#define NEEDINPUTS 0

#define pi      3.14
#define rho_b   1000    /* Density of brine */
#define cp_b    4.18    /* Heat capacity of brine */
#define n_tube  2700    /* Number of tubes in heat exchanger */
#define d_i     0.0293  /* Internal diameter of tubes */
#define L       17.33   /* Length of tubes */
#define k       1.942   /* Heat transfer coefficient */
#define n_seg   10      /* Number of segments */
#define eps     0.001   /* Valve constant */
#define rho     0.7     /* Density of supply steam */
#define k_vs    7       /* Valve coefficient */
#define H_v     2230    /* Evaporation enthalpy */
#define R       0.00461 /* Ideal gas constant */
#define c2K     273     /* From Celsius to Kelvin */

static double T_cond;
static double m_hs;

void derivatives(t,x,u,dx)
double t,*x,*u;
double *dx;
{
double h_in,m_b,Tb_in,p_in,T_hsin;
double h,rho_hs,A_i,T_b,T_ln,Q_hs;
double phi,m_cond,V_hs,SUM_Q_hs,p_sh,V_b,Tb;
int i;

        /*Inputs*/
        h_in=u[0];       /* Control signal to steam valve */
        m_b=u[1];        /* Recycle flow */
        Tb_in=u[2];      /* Inlet temperature */
        p_in=u[3];       /* Steam supply pressure to valve */
        T_hsin=u[4];     /* Temperature of saturated steam */

        /* Steam pressure in heater */
        rho_hs=x[11];    /*Density of steam*/
        p_sh=rho_hs*R*(T_cond+c2K);

        /* Steam condensation temperature */
        T_cond=(-1750.268/(log10(p_sh/1.332e-3)-8.1077)-235);
```

```
        /* Heat Transfer surface */
        A_i=(pi*n_tube*d_i*L)/n_seg;

        /* Volume of brine */
        V_b=(0.25*pi*n_tube*L*d_i*d_i)/n_seg;

        /* Inlet temperature */
        Tb=x[0];

        /* Logarithmic mean temperature, T_ln */
        T_ln=(Tb-Tb_in)/log((T_cond-Tb_in)/(T_cond-Tb));

        /* Heat flow */
        Q_hs=(k*A_i*T_ln);
        SUM_Q_hs=Q_hs;

        /* Energy balance */
        dx[0]=(Q_hs+m_b*cp_b*(Tb_in-Tb))/(rho_b*cp_b*V_b);

        /* Calculate each segment */
        for (i=1;i<n_seg;i++){
                /* Logarithmic temperature, T_ln */
                Tb=x[i];        /* Temperature output for segment */
                Tb_in=x[i-1];   /* Inlet temperature for segment */
                T_ln=(Tb-Tb_in)/log((T_cond-Tb_in)/(T_cond-Tb));

                /*Q_hs*/
                Q_hs=(k*A_i*T_ln);
                SUM_Q_hs=SUM_Q_hs+Q_hs;

                /*Energy balance*/
                dx[i]=(Q_hs+m_b*cp_b*(Tb_in-Tb))/(rho_b*cp_b*V_b);
        }

        /*------------------------------*/
        /* Valve acuator equations      */
        /*------------------------------*/
        /* Valve opening */
        h=x[10];

        if (fabs(h-h_in)<eps){
                dx[10]=0;               /*d_h=0;*/
        }
        else{
                if (h<h_in){
                        dx[10]=0.02;    /*d_h=0.02;*/
                }
                else{
                        dx[10]=-0.02;   /*d_h=-0.02;*/
                }
        }

        /* Saturation limit for valve opening */
        if(h>1){
                dx[10]=-0.02;
                h=1;
        }

        /*------------------------------*/
        /*Steam density equations       */
        /*------------------------------*/

        /* PHI-equal percentage valve equation */
        /* The ratio of k_vo/k_vs is about 0.04 */
        phi=0.04*exp(3.22*h);

        /* PHI-linear characteristic valve equation */
        /*phi=0.04+0.96*h;*/

        /* Steam flow characteristic */
        /* The value of c_G is 519 */
        m_hs=k_vs*519*sqrt(fabs((p_sh*(p_in-p_sh)*rho)/T_hsin))*phi;

        /* Flow of condensate */
        m_cond=SUM_Q_hs/H_v;

        /* Steam volume */
        V_hs=4.07*(V_b*n_seg);

        /* Change of steam density */
        dx[11]=(m_hs-m_cond)/V_hs;

}
void init_conditions(x0)
double *x0;
{
```

```
%with the integrator
for i =1:length(Aa),
        for j=1:length(Ww),
                LI(i,j)=L(i,j)*I(i,j);
        end;
end;
```

A.9 Lsys.m

```
function [Lsystem,Xmod]=Lsys(LI,A,W)
%This functions finds the SIDF for the 3rd order
%brine heater model. The inputs are:
%       LI:     The SIDF of the controller
%       A:      The original amplitude vector
%       W:      Frequency vector

%Variables passed to the Simulink system
global Asys Wsys

%Size of integration step used in simulation
stepsize=0.05;

%Find modified input amplitudes, Xmod
mod=abs(LI);
for i=1:length(A),
        Xmodmax(i)=max(abs(LI(i,:)))*A(i);
        Xmodmin(i)=min(abs(LI(i,:)))*A(i);
end;
Xmin=min(Xmodmin);
Xmax=max(Xmodmax);
Xmod=logspace(log10(Xmin),log10(Xmax),length(A));

%For all amplitudes
for i=1:length(A)

        %For all frequencies
        for j=1:length(Xmod)

                %Input amplitude
                Asys=Xmod(i);

                %Input frequency
                Wsys=W(j);

                %Set final time to 30 periods
                Tf=30*pi*2/Wsys;

                %Simulate Simulink system
                rk45('lsys',Tf,[],[1e-03 stepsize stepsize 0 0 2]);

                %Extract last period of TBT
                y=tb(length(tb)-1/Wsys*2*pi/stepsize:length(tb));
                %Take average value
                y=y-mean(y);

                %Extract last time period
                t=t(length(t)-1/Wsys*2*pi/stepsize:length(t));
                %Make period start at time 0
                t=(t-t(1));

                %Calculate discrete integral

                int1=(Wsys/(Asys*pi))*sum(y.*sin(t*Wsys)*stepsize);
                int2=(Wsys/(Asys*pi))*sum(y.*cos(t*Wsys)*stepsize);

                %Store SIDF in array
                Lsystem(i,j)=int1+sqrt(-1)*int2;
        end;
end;
```

A.10 controller.m

```
function [xprime]=controller(wt,x)
%This function returns the SIDF of the FLC.
%Written by J.P.Marin, LAAS
```

```
%Inputs:
%       wt:     frequency
%       x is a dummy state vector
%       x(1)=X
%       x(2)=W
%       x(3)=RE(L(X,W))
%       x(4)=Im(L(X,W))
%       x(5)=Ke
%       x(6)=Kde
%       x(7)=Ku

% controller parameters
global Mem_e Mem_de Mem_u Tableij

% e=K_e*X*sin(wt)
e=x(5)*x(1)*sin(wt);

% edot= K_edot*X*w*cos(wt)
de=x(6)*x(1)*x(2)*cos(wt);

%Maximum allowable value for the inputs is 1
if ( abs(e)>=1  ) e=sign(e); end;
if ( abs(de)>=1  ) de=sign(de); end;

% u=K_u*FLC(e,edot)
output = x(7)*sugenomex ( e, de, Mem_e, Mem_de,Mem_u, Tableij );

%output
%dummy variables
xprime(1)=0;
xprime(2)=0;

%Calculate integral
xprime(3)=2/(pi*x(1))*output * sin(wt);
xprime(4)=2/(pi*x(1))*output * cos(wt);

%dummy variables
xprime(5)=0;
xprime(6)=0;
xprime(7)=0;
```

A.11 STABSIDF.m

```
%Initialization
init_table;
global Asys Wsys

%Find the SIDF of the controller
[W,X,Lfuzzy,LIfuzzy]=SIDF;

%Find the SIDF of the system
[Lsystem,Xmod]=Lsys(LIfuzzy,X,W);

%perform the linear interploation to match
%the amplitude and frequency pairs
for i=1:length(X),
        for j=1:length(W),
                Xmodreel(i,j)=X(i)*abs(LIfuzzy(i,j));
                Xmodre=ones(length(X),1)*Xmodreel(i,j);
                diffe=Xmod'-Xmodre;
                [k]=min(find(diffe>-eps));

                if k==1,
                        Lsys(i,j)=Lsystem(1,j);
                else
                        indicemin=k-1;
                        indicemax=k;
                        Lsys(i,j)=(Xmodreel(i,j)-Xmod(indicemin))/(Xmod(indicemax)-Xmod(indi-
cemin))*Lsystem(indicemax,j)...
        +(-Xmodreel(i,j)+Xmod(indicemax))/(Xmod(indicemax)-Xmod(indicemin))*Lsystem(indi-
cemin,j);

                end;
        end;
end;

%The SIDF of the Open Loop System is:
Lols=LIfuzzy.*Lsys;

%save work
%save tot_stab;
```