Analysis of Surge Phenomena in Axial Flow Compressors

By
Muhammad Kamran Khan Tareen

Research Centre for Modeling and Simulation (RCMS),
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.
2014

Analysis of Surge Phenomena in Axial Flow Compressors

Muhammad Kamran Khan Tareen
2011-NUST-MS-CS&E-33

This work is submitted as MS thesis in partial fulfillment of
the requirement for the degree of
MS in Computational Science and Engineering

Supervisor Name: Dr Mukaram Khan

Research Centre for Modeling and Simulation (RCMS),
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.
2014

IN THE NAME OF ALLAH THE MOST GRACIOUS THE MOST BENEFICIENT

Dedicated to my parents, wife and kids who are always source of energy for me.

Acknowledgements

I am deeply in debt to my supervisor, Dr. Mukaram Khan for his guidance, suggestions, constructive criticism and encouragement. His patience and understanding helped me sail through the hard times. I am grateful to Dr. Jamil Ahmad, Sir Sikandar Hayat Mirza and Mr. Tariq Saeed, for taking personal interest and efforts to encourage me to complete this work.

I would like to thank my wife who encouraged me to do MS. She managed with my hectic schedules and her support always gave me a new energy to finish my work. Last but not the least, I am grateful to my parents whose love and prayers are always a source of encouragement for me.

Abstract

The use of Neural Network coupled with Fuzzy Logic in control systems is on increase due to its capabilities of adapting to the plant dynamics variations, dynamic environmental effects or working in ill-defined environment. In contrast to classical methods, these techniques do not require a mathematical model for designing the controllers. The accuracy of mathematical models can be questioned in the case of dynamic, nonlinear and complex plants / processes. Neuro-Fuzzy systems are based on the experts' knowledge and trained judgment of skilled workers, covering the whole dynamics of the system through learning process and are better suited for dynamic environments. This work is focused towards the use of adaptive neurofuzzy inference system in control systems for the estimation and control of the error due to plant model mismatch and process uncertainties in a Super-saturation Controlled Batch Crystallization process producing crystals with desired properties (target size distribution). As a result of this research, a neuro-fuzzy controller is proposed which is capable of compensating uncertainties related to the plant dynamics including stirring speed, model inaccuracy and measurement errors for the control of temperature trajectories for super-saturation controlled industrial batch crystallization process. We are using systemic direct design approach (producing optimal temperature trajectories for super-saturation set point with time) for crystallization of Potash Alum concentration in water, yielding desired target Crystal Size Distribution at the end of a batch. The validation results are very encouraging and prove the efficacy of fuzzy neural networks in the designing of dynamic control systems.

Table of Contents

List of Figures

Abbreviations

NN	Neural Network
ANN	Artificial Neural Network
ANFIS	Adaptive Neuro Fuzzy Inference System
QMOM	Quadrature method of Moments
CSD	Crystal size Distribution
MOCH	Method of Characteristics
PBE	Population Balance Equation
ODE	Ordinary Diffrential Equation
PD	Product Difference
MSZW	Meta Stable Zone Width
MLP	Multi Layer Perceptrons
FNN	Neuro Fuzzy System
FL	Fuzzy Logic

Chapter 1

Introduction

1.1 Intelligent Control system

An Intelligent Control system is capable of operating with minimum amount of human interaction in ill-defined environments, learning significant information in a stable manner and placing minimum number of restrictions on the dynamics of the plant **[**17**].**This system is capable of making decisions in place of humans. Intelligence systems adapt themselves using some example situations (inputs &output of a system) and they correct decisions automatically for future situations after going through a required amount of trainings [17].Human learning process elements i.e. correlating, inference from a lot of data, learning from examples etc are the parts of intelligent system. Now days, the focus of research in control system design is shifting to empowering machines with these qualities [18].

1.2 Adaptive Neuro Fuzzy System

1.2.1 Fuzzy Logic

Fuzzy Logic provides morphology that enables approximating human reasoning capabilities like quantitative expressions. With fuzzy sets, human concept such as small, young, big can be translated to a form usable by computers and is used as a tool to infer decisions under vagueness [16].

1.2.2 Artificial Neural Network

Artificial neural networks are models inspired by computational properties of neurons cells in human brain**[**18**].**It consist of a large number of simple, highly interconnected processing elements called "neurons" that can compute values from inputs by feeding information through the network. Artificial Neural Networks (ANN) are capable of simulating powerful functionality of human brain such as learning, adaptive fault tolerance, parallelism and generalization [17].

1.2.3 Neuro Fuzzy System

The combination of fuzzy systems with ANN is called neuro-fuzzy systems. Iintegrated neuro-fuzzy systems can combine human-like knowledge representation and explanation abilities of fuzzy system with the parallel computation and learning abilities of neural networks. Due to fusion of these techniques, neural networks become more transparent, while fuzzy systems are capable of learning[18].An Artificial Neural Fuzzy Inference System (ANFIS) is trained, not programmed; so through learning it adapts to the new changes. As a result its performance improves every times it undergoes training [9].

1.2.4 Use of Adaptive Neuro-Fuzzy System

Use of fuzzy logic and ANN for control systems has become a popular research topic in recent years [16], as this technique does not require a mathematical model for designing. The accuracy of mathematical models is quite questionable in the case of nonlinear, dynamic and complex plants / processes; as a result the performance of the conventional controllers usually degrades. As neuro fuzzy systems are based on the knowledge of expert and trained judgment of skilled workers, they are able to cover most of the dynamics of the system through learning process and are better suited for such environments [10].In addition to control problems, FNN has also successfully been applied to other engineering fields, such as system identification [18], and system uncertainty estimation [18].

1.3 Crystallization Process in Industry

Batch Crystallization is highly complex nonlinear process in chemical industry and is commonly used for purification due to its simplicity, flexibility, and less requirement for process development / investment as compared to other separation / purification techniques [1]. There are a number of issues associated with its control design due to complex dynamics and uncertainties related to governing phenomena. The major concern in the process has been to optimize consistency and quality of the final product while improving productivity [1, 2].

Batch crystallization processes are normally carried out by manipulating the temperature trajectories. The state variables for crystallization process include solution concentration, process temperature and jacket temperature [4].Due to non-availability of measurement devices for all process state variables (i.e. super saturation measurement sensors are at prototype stage and are often not available for industrial scale use) and non-linear dynamic nature of the process make it difficult to design practically useable fully automatic controllers [4].

1.4 Control of Crystallization Process

Currently control system for crystallization process is designed either by using direct design or a model base approach. In direct design approach processes follow predetermined temperature profiles in time which are chosen arbitrarily or by trial and error experiments [4]. These depend upon the dynamics of the plant. While the operating conditions in model-based approach are optimized by using measureable data and predictions of immeasurable data based on model. Main problem is the need for computing the solution of the optimization problem within the sampling period.

1.5 Challenges in Crystallization Process

The main focus in industrial sector is in understanding the process and controlling it. The aim is to achieve the desired end properties of the crystals e.g.to get uniform and reproducible Crystal Size Distribution (CSD). This is getting popularity in chemical industry, especially in the pharmaceutical industry, as most of the product acceptance standards are directly related to the shape and the size of distribution. The shape of CSD, obtained from crystallization process, also strongly affect the efficiency of downstream operations such as filtration, drying and washing [3].

1.6 Latest Advancements

Latest developments in crystallization modeling process provide a solution to obtain the temperature profiles in the time domain, corresponding to a desired target CSD, providing a systematic direct design approach for practical applications [3-5]. This approach is based on the solution of Population Balance Equation by combination of Quadrature Method of Moments (QMOM**)** and Method of Characterization (MOCH**)**[3].

1.7 Research Proposal

This work presents a study that demonstrates the use of ANFIS to control the crystallization process based on a systemic direct design approach in order to compensate for the errors due to plant-model mismatch and unmeasured process disturbances, and getting the advantages of online dynamic optimization technique with less computation resources.

1.8 Research Objectives

To design an ANFIS based controller for controlling industrial crystallization process which is:

- Easy to implement with minimal hardware changes.

- Computationally efficient and fault tolerant.
- Does not depend on the dynamics / structure of the plant.
- Improves in performance by learning and adapting to the new environmental conditions.

1.9 Organization of the Thesis

This thesis is organized in five chapters. The second chapter describes the basic concepts related to neural networks and fuzzy logic. Third chapter explain how fuzzy logic and neural networks engineering can be combined to optimize their benefits in overcoming uncertainties in a dynamic environment. In fourth chapter concept of crystallization process and different techniques to control the crystallization process in industry are summarized. In this chapter systematic design approach to obtain temperature trajectories in time domain is explained. Our study has been carried out by using this systematic design approach. In the fifth chapter the design of control system to control temperature trajectories is presented. This controller uses ANFIS to estimate and compensate for the uncertainties related to all types of error. In sixth chapter design is tested again noisy data and results are discussed. The last chapter summaries the research and presents some important conclusions from the results. We also offer some suggestions for the future work in this chapter.

Chapter 2

Artificial Neural Networks and Fuzzy Logic

2.1 Neurons

Neurons are neural cells, and network of these cells is called Neural Network. Human brain's powerful capabilities in remembering, recalling, correlating, interpreting, reasoning and inference have always been a candidate for modeling and simulation to understand these phenomena in detail. Initially researchers have been focusing towards studying stimulus response relationships of biological neurons. Later computational properties of neurons attracted the attention of researchers to use model of neuron to solve various engineering problems [10].

2.2 Artificial Neuron

An artificial neuron is a device /computational element with number of inputs and one output [17]. It is also called a node or unit. It receives input from some other neurons or from an external source. Each input has a weight *a*ssociated with it. This weight can be modified so as to model synaptic learning. The unit computes some function f of the weighted sum of its inputs [17]:

$$y_i = f\left(\sum_j w_{ij} y_j\right)$$

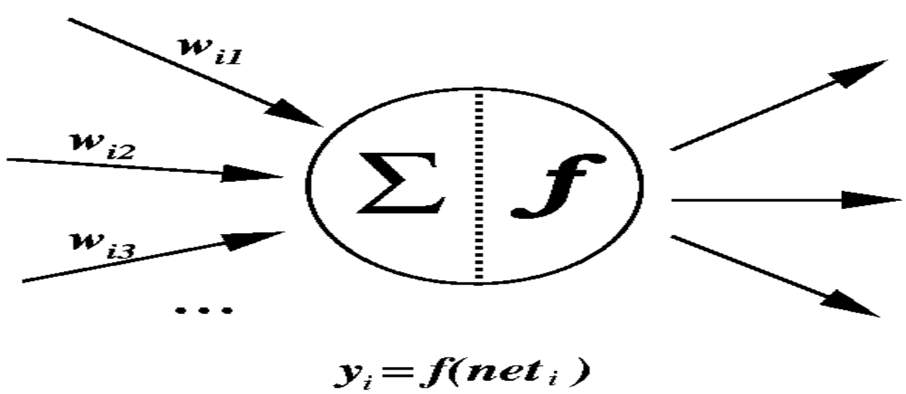


Figure 2.1: Model of Artificial Neuron [17].

Where $\sum_j w_{ij} y_j$ is the weighted sum, called the **net input** to unit i, often written net_i and w_{ij} refers to the weight from unit j to unit i (not the other way around). The function f is the unit's **activation function**. In the simplest case, f is the identity function, and the unit's output is just its net input[17].

2.3 Computational Process with Artificial Neurons

Artificial neuron receives input from a number of others neurons or from an external stimulus, a weighed sum of these inputs constitutes the argument to an activation function. The resulting value of activation function is the output of the artificial neuron. This output is distributed along weighted connections to other neurons.

2.4 Artificial Neural Networks

An artificial neural network is a computational system consisting of a large number of simple, highly interconnected processing elements (artificial neurons) in an architecture inspired by the structure of the cerebral cortex of the brain[17].ANN maybe physical devices, or simulated on conventional computers. From a practical point of view, an ANN is just a parallel computational system consisting of many simple processing elements connected together in a specific way in order to perform a particular task.

2.4.1 Modes of operations of ANNs

- The training mode (and testing).
- The usage mode.

2.4.1.1 Training mode

The operation in which synoptic weights are adjusted to get desired result is known as training mode. During training mode a set of data is presented to network, network produces an output. The output of network is compared with the desired output. If this output does not match with the desired output then weights are adjusted, and once again input patterns are presented to network and the output obtained is compared with desired output. This process continues until desired output with accepted error is obtained or the required number of training cycles are met [17].

2.4.1.2 Usage mode

Once ANN is trained, it can be used for the operation for which it has been trained. The data is presented to the input layer neurons which output as per their training.

2.4.2 Types of Training Modes for ANNs

2.4.2.1 Supervised Training

In this training mode an input stimulus is provided to ANN, which produces the output according to its internal design. The produced output is compared with desired output and difference if any is noted. In next epoch the neurons' weights are adjusted to bring the output closer to the desired value. This process continues till the output with accepted accuracy is obtained. After the training, output response of a specific input is always the same. This is shown in figure 2.2.Supervised learning is commonly used in MLP [17].

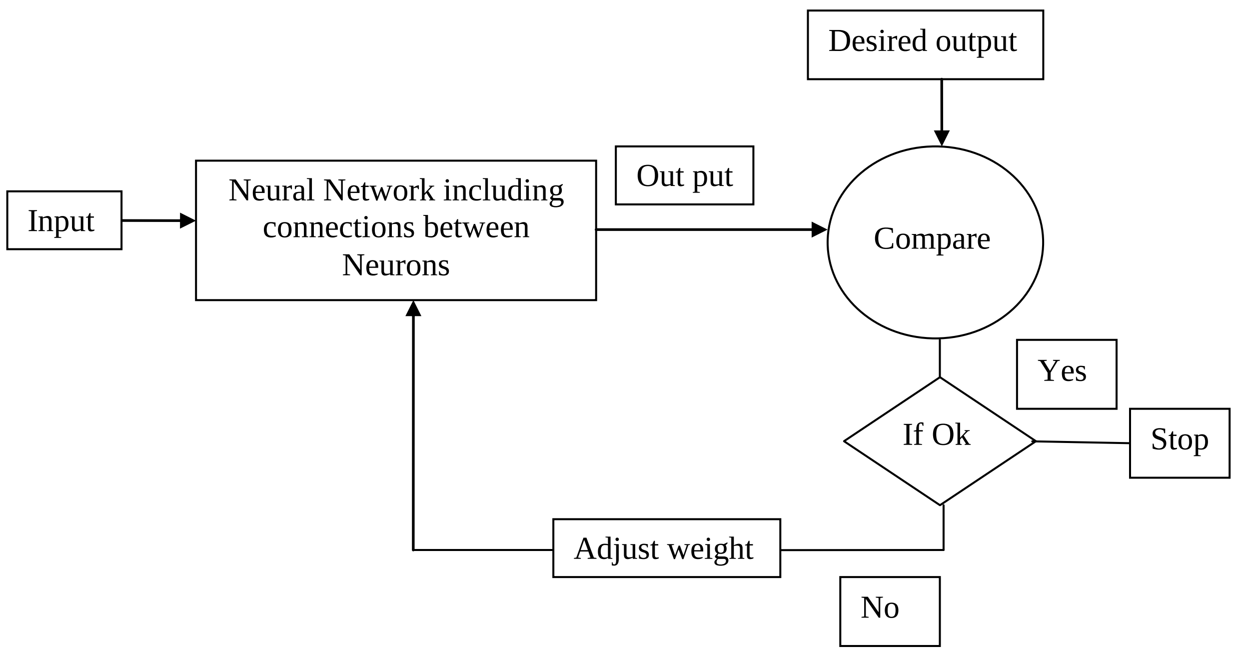


Figure 2.2: Supervised Training Mechanism [17].

2.4.2.1 Unsupervised Training

In unsupervised training, no desired output is provided and network simply reorganized itself to each input stimulus according to internal structure. The output response for a specific input cannot be determined. Kinds of application with unsupervised networks are different than those with supervised networks. Khonen's network is an example of such network [17].

2.4.2.1 Graded / Reinforced

This training is similar to the supervised training, however, instead of desired output only a grade on the network is provided [17].

2.4.3 Static and Dynamic ANNs

ANN is static if for a given input its output remains same and is said to be dynamic if its output varies.

2.5 Essential ingredients of ANNs based Computational System

- Architecture (how neurons are connected).
- Threshhold Function.
- Learning Rule.

The system can be implemented by using digital, optical or analog computing elements as building blocks. For example device that behaves like neuron is analog operational amplifier.

2.5.1 Architecture of ANN

Neural network can be modeled or categorized depending on their architecture. Depending on architecture neural networks can be categorized into different models such as Perceptron, Hopfield, Feed Forward, Back propagation etc.

2.5.1.1 Layers in ANN

ANN consists of neurons which are divided in groups. Neurons in each group are arranged in columns or rows and are referred to as layer of neurons in the network. ANN may have one, two or more than two layers. Each ANN has an input and output layer and can have a number of intermediate layers, also called hidden layers.

2.5.1.2 Neural Operation in a Layer

In a single layer network the input signals are received at the same layer, processing is done by its neurons and output is generated in the same layer. When more than one layer are present, neurons at first layer receives the input signals and supply it to the neurons in the next layer which computes the output and transmit this output as input to the next layer. Every network has input layer but in most of the cases this layer is used to transmit the information to next layer.

2.5.1.2 Connections between layers

Feed Forward: The network in which information flows only in one direction i.e. from input layer to output, through the intermediate layers in any. It consists of a (possibly large) number of neurons, organized in *layers*. Neurons in each layer are connected to all or some of the neurons in the previous layer. These connections may have a different strength or *weight*. Weights represent the level of conductivity between strength of connection between two neurons. The weights on these connections encode the knowledge of a network. Data enters the inputs layer and passes through the network, layer by layer, until it arrives at the outputs layer. During normal operation, that is when it acts as a classifier, there is no feedback between layers. This is why they are called feed forward neural networks. In the following figure we see an example of a 2-layered network with, from top to bottom, an output layer with 5 units and a *hidden* layer with 4 units, respectively. The network has 3 input units[17].

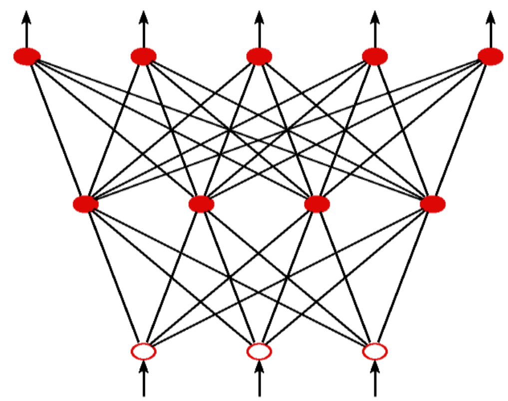


Figure 2.3: Feed forward Neural Network [17].

Feed back or Recurrent: In this type of network information flows in both direction and some processing may be carried out at input layer. A recurrent neural network (RNN) is a class of neural network where connections between units form a directed cycle. This creates an internal state of the network which allows it to exhibit dynamic temporal behavior [20]. Unlike feed forward neural networks, RNNs can use their internal memory to process arbitrary sequences of inputs. This makes them applicable to tasks such as un segmented connected handwriting recognition, where they have achieved the best known results. An example is Hopefield NN.

2.5.2 Threshhold Function

Output of a neural network may be in a bit pattern, binary function value or in some analog signal form. An output depends on the type of mapping intended for inputs. Threshold function maps the final activation of input neuron to the network output.

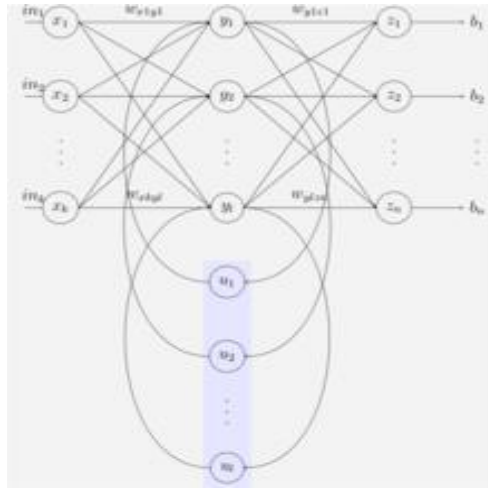


Figure 2.4: Feedback or recurrent Neural Network [17].

Output of neuron is the result of threshholding of its internal activations[20]. Internal activation is the weighted sum of the neuron's inputs. Thresholding is done for the scaling down to map it into meaningful output or to add bias. Some of threshold functions are explained below[20].

2.5.2.1 Sigmoid Function

Sigmoid functions have more than one form which differs in their formulas and in their ranges. They all have a graph similar to a stretched letter **s**. Two examples of sigmoid functions are given below. The first is the hyperbolic tangent function having values in range (–1, 1). The second is the logistic function and has values between 0 and 1. Choice of the function depends upon the required range.

a. $f(x) = tanh(x) = (e^{x} - e^{-x})/(e^{x} - e^{-x})$

b. $f(x) = 1 / (1+ e^{-x})$

2.6 Multilayer Perceptron (MLP)

MLPs consist of multiple layers of neurons. First layer is input layer and last layer is output layer while internal layers are known as hidden layers. Neurons in each layer are fully interconnected to all the neurons in the next layer. Connections are characterized by weights. Each neuron adds up the product of weight and corresponding output carried by all the connections to the neuron. In MLP, Sigmoid function$f(x) = 1/(1 + e^{-x})$ is used to calculate the output. The theme behind the MLP is that for each input pair there exist a set of weights that can map the input to output. During the learning MLP updates the connection weights for achieving any desired input output mapping. More than one input output mapping can be represented by the weights in the networks. Hidden layers act as a feature detectors or filters for some type of inputs[17].

2.6.1 Learning in MLPs

There are two phases in its learning process, in first phase the input pattern is presented to the MLP, while in the second phase weights are adjusted to get the required output. During the first phase input is presented to obtain the output. This output is compared with the required output to obtain the error signal. Error signal is back propagated to the hidden layer(s).Weights of the connections are adjusted on the basis of the squared error, or some other metric. Error is reduced in each cycle to finally minimize within a tolerance range [20].

During supervised learning MLP is presented with a finite set of input and relevant output patterns. Input pair is presented to neurons at first layer which transmits it to the neurons in the next layer which computes the output by using the bias and threshold function and transmit this output to the next layers. These hidden layer outputs become inputs to the next hidden or output layer neurons. The output of the network is obtained by the activations of neurons from the output layer. The network output and the required patterns /output are compared. Based on the error, weights of connections between the hidden layer and the output layer are adjusted. On the similar basis, based on the error in the output layer, connection weights between the input and hidden layers are adjusted. The process is repeated with each pattern of input pair assigned for training the network. Presenting each pattern of input and adjusting connection weight for training is called a cycle or an epoch. The procedure is then repeated i.e. training cycles continue until the error is within a prescribed tolerance. Back propagation uses gradient descent in order to minimize the squared error between the network output and the target values for these outputs.

2.6.2 Back propagation Algorithm for MLP [20]

- Initialize the weight to small numbers.
- Until satisfied, Do
    - For each training example Do
        - Input each training example to network and compute the network outputs
        - For each output unit k

$$\delta_{k \leftarrow O_k(1-O_k)(t_k-O_k)}$$

- For each hidden unit h

$$\delta_{h \leftarrow O_h(1-O_h)} \sum_{k \in outputs} \delta_k w_{h,k}$$

- Update each network weight w_{ij}

$$w_{i,j} \leftarrow w_{i,j} + \Delta w_{i,j}$$

Where

$$\Delta w_{i,j} = \eta \delta_j x_{i,j}$$

2.6.3 Learning in MLP

Activation of Neurons

- **Hidden Layers**

Find weighted sum of the input on each neuron in hidden layer [20]

$$\sum_{i=1}^{n} w_i x_i$$

Finding activation based on the weighted input and threshold

$$f(x) = 1 / (1 + e^{-x})$$

This will become the output of hidden layer neurons and input to the output neurons

- **Output Layer**

Same process as for the hidden layer neurons

- **Compute Error**[20]
  - Compare the output with the target
  - Difference from Target $O1 = t1 - O1$
  - Compute Error at $O1 = O1*(1-O1)(T1-O1)$
  - Back propagate the error to the hidden layer neurons
  - Compute Error at hidden layer with the help of back propagated error value as

    $H1 = O_{h1} * (1-O_{h1}) * \sum w_{h,k}\delta_k$
  - Find adjustment in threshold at output neurons as

    $E_{O1} = \eta$*computed error*
  - Find adjustment in threshold at hidden layer neurons as

$$E_{H1} = \eta * computed\ error$$

- Find adjustment in weights at output neurons as

$$\Delta W_{O1} = \eta * \delta_k * x_{i,j}$$

- Find new weights as $W_{new} = W_{old} + \Delta W$
- Find new threshold as $\theta_{new} = \theta_{old} + \Delta\theta$

2.6.4 Termination of Training[20]

The choice of termination condition is important because

- Too few iterations can fail to reduce error sufficiently.
- Too many iterations can lead to over fitting the training data.

2.6.5 Termination Criteria

- After a fixed number of iterations (epochs).
- Once the error falls below some threshold.
- Once the validation error meets some criterion.

2.7 Fuzzy Logic

Ordinary or mathematical logic deals with true and false i.e. a preposition can be true (have a value '1') or false (have a value'0'). Ordinary logic does not admit degree of truth in between these two extremes. In other words ordinary logic does not admit imprecision in truth [16].

In contrast fuzzy logic deals with prepositions having any value between'0' and '1' i.e. preposition can be true to a certain degree. Preposition's truth value indicates the degree of certainty about which the preposition is true. With fuzzy logic, human concept like small, young, big, can be translated to a form usable by computers. It provides morphology that enables approximate human reasoning capabilities to be applied to knowledge base and quantitative expressions.

2.7.1 Fuzzy Set

"Fuzzy set" is an extension of the definition of a classical set in which the characteristic function is permitted to have any values between 0 and 1. A "fuzzy set" A in X can be defined as a set of ordered pairs:

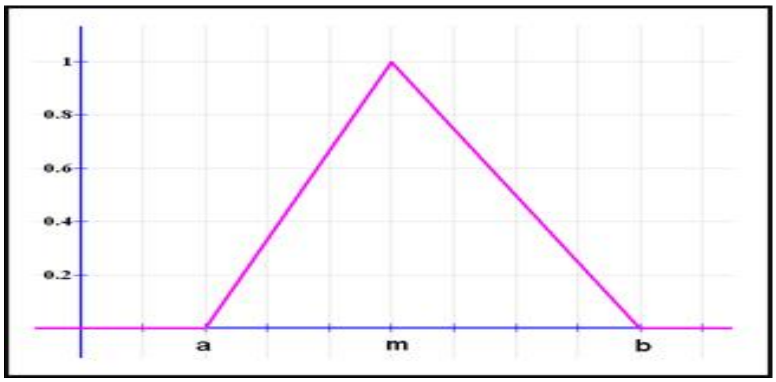
$$A = \{(x, \mu_A(x)) | \ x \in X\}$$

Where $\mu_A(x)$is called membership function for the fuzzy set *A*. It maps each x to a membership grade between 0 and 1. Examples of some membership functions (Triangular, Trapezoidal and Gaussian) are described below.

2.7.1.1 Triangular function

Defined by a lower limit **a**, an upper limit **b**, and a value **m**, where **a < m <b[16]**.

$$\mu_A(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{m-b} & a < x \leq m \\ \frac{b-x}{b-m} & m < x < b \\ 0 & x \geq b \end{cases}$$



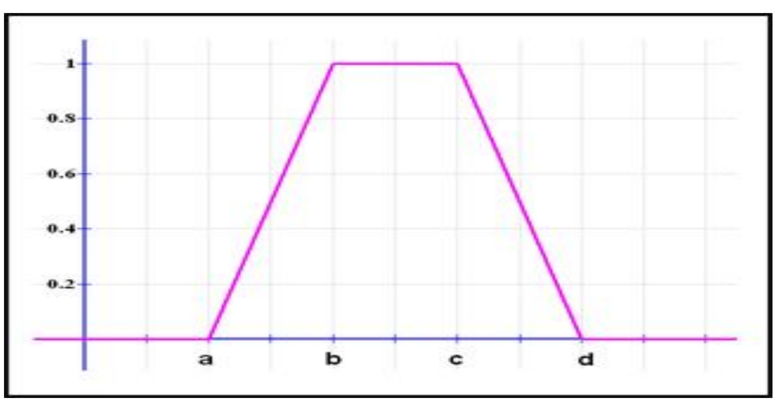
Figurer 2.5: Triangular Membership[16].

2.7.1.2 Trapezoidal function:

Defined by a lower limit **a**, an upper limit **d**, a lower support limit **b**, and an upper support limit **c**, where **a < b < c < d [16]**.

$$\mu_A(x) = \begin{cases} 0 & (x < a) or (x > d) \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \frac{b-x}{b-m} & c \leq x \leq b \end{cases}$$

14

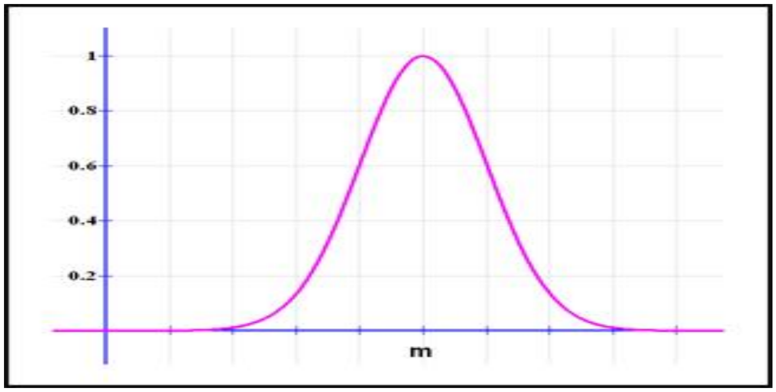


Figurer 2.6: Trapezoidal Membership[16].

2.7.1.3 Gaussian function

Defined by a central value **m** and a standard deviation **k** > **0**. The smaller k is, the narrower the “bell” is[16].

$$\mu_A(x) = e^{-\frac{(x-m)^2}{2k^2}}$$



Figurer 2.7: Gaussian Membership[16].

2.8 Steps to Design Fuzzy Controller

- Define problem (normally complex) in detail.
- Identify all important variables and their range.
- Determine membership profile for each variable range.
- Determine rule.
- Select defuzzification methodology

2.8.1 Structure of Fuzzy Control System

Structure of a fuzzy control system is shown in Figure2.3. It is composed of the following four elements [16]:

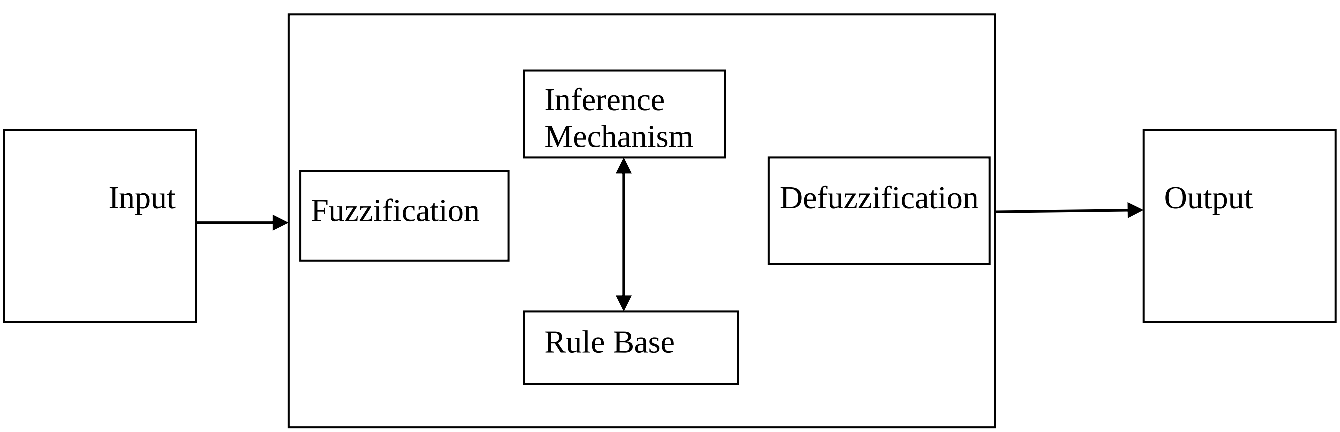


Figure 2.8: Structure of Fuzzy Control System [16]

a. **Rule Base** (rules and parameters for membership functions), which contains a fuzzy logic quantification of the expert's linguistic description / knowledge and skill of how to achieve desired performance /result in the form If-Else statements.

b. **Inference Mechanism** or Decision Unit, which follows the expert's decisions about how best to control the plant. It evaluates which rules are relevant to current situation and what will be the input to the plant.

c. **A Fuzzification Interface,** the process of transforming crisp values into grades of membership for linguistic variables of fuzzy sets. The membership function is used to associate a grade to each linguistic term.

d. **A Defuzzification Interface,** which converts the conclusions of the inference mechanism into actual inputs for the plant.

2.8.1.1 Fuzzification Phase

This phase transforms the real value into fuzzy sets. Triangular membership functions are employed to quantify the meanings of linguistic description for input and output variables in this study. These are selected due to their computational efficiency and better control performance. Guidelines that should be considered to determine the range of the fuzzy variables as related to the crisp inputs are:

- Symmetrically distribute the fuzzified values across the universe of discourse.
- Use an odd number of fuzzy sets for each variable so that some set is assured to be in the middle. The use of 5 to 7 sets is fairly typical.
- Overlap adjacent sets (by 15% to 25% typically).

2.8.1.2 Fuzzy Rule Base

Fuzzy rules serve to describe the quantitative relationship between variables in linguistic terms. There are four methods, used in defining of a rule base:

- Control knowledge
- Modeling the operator's behavior,
- Fuzzy modeling
- Fuzzy controller with self learning.

Defining the rule according to modeling the operator's behavior and control engineering knowledge are ideal methods. The other methods are known as deterministic methods. The deterministic method was used by Takagi and Sugeno and the heuristic method was used by Mamdani [16].The knowledge based on IF-THEN rules is the commonly used method to define the rule base as it needs more engineering skills and experience than plant information.

2.8.1.3 Fuzzy Inference

An inference engine produces a new fuzzy set based on inference. Inference engine calculates the overall value of the fuzzy control output based on the individual contributions of each rule in the rule base. Each contribution on the inference engine represents the value of the fuzzy control output as computed by a single rule.

2.8.1.4 Defuzzification

Defuzzification converts the output of inference into a real value which could be used as a fuzzy control input i.e. the set of output values from inference is converted into a single point value. Several methods [126, 130]can be applied in the defuzzification to obtain the fuzzy control signal, which are given as:

2.8.1.4.1 The Center of Gravity Method

This widely used method generates a center of gravity (or center of area) of the resulting fuzzy set of a control action. According to the centre of gravity method, the crisp value of the fuzzy control output is given as[20]:

$$u=\frac{\sum_{i=1}^{n} r_i\mu_i}{\sum_{i=1}^{n} \mu_i}$$

where n is the number of quantization levels, ri is the amount of control output at the quantization level i and $\mu_i$represents its membership value.

2.8.1.4.2 The Mean of Maximum Method

The mean of maxima method generates a crisp control action by averaging the support values which their membership values reach the maximum. In the case of discrete universe the output is given as[20]:

$$u= \sum_{i=1}^{l} \frac{r_i}{l}$$

where l is the number of the quantized r values, which reach their maximum Memberships.

2.8.1.4.3 The Weighted Average Method

This method is used when the fuzzy control rules are the functions of their inputs [161] as shown in the Figure 4.2. For this type of fuzzy reasoning, in general, the consequent part of the rule is [18]:

$$u = f(x, y)$$

If *wi* is the firing strength of the rule *i*, then the crisp value is given by:

$$u=\frac{\sum_{i=1}^{n} w_i f(x,y)}{\sum_{i=1}^{n} w_i}$$

where *n* is the number of rules.

Center of gravity method is used for defuzzification. To get there al control input (*u*) the fuzzy output should be de normalized by using scaling factor.

2.9 Fuzzy Model

Fuzzy model consists of rules that describes action taken or each combination of control variables and normally appear in IF-Then form e.g. If temp is cool and pressure is weak then throttle action is Positive large. Two important Fuzzy models are given below:

2.9.1 Mamdani Model

This is Linguistic Model which describes the system behaviour by associating a set of coordinated fuzzy predicates in the form of if-then rules:

IF {Antecedents} THEN {Consequent}.

2.9.2 T- Sugeno Model

T-Sugeno or Takagi-Sugeno-Kang method of fuzzy inference was introduced in 1985 [7], It is similar to the Mamdani method in many respects. The first two parts of the fuzzy inference process, fuzzifying the inputs and applying the fuzzy operator, are exactly the same. The main difference between Mamdani and Sugeno is that the Sugeno output membership functions are either linear or constant [7]. In Takagi and Sugeno method each rule's output is a linear combination of input variables. The crisp output is the weighted average of each rule's output A typical rule in a Sugeno fuzzy model has the form [16]:

If Input 1 = x and Input 2 = y , then Output is z = ax + by + c

For a zero-order Sugeno model, the output level Z is a constant (a=b =0). The output level z of each rule is weighted by the firing strength w_i of the rule. For example, for an AND rule with Input 1 = x and Input 2 = y, the firing strength isw_i = And Method (F_1

(x), F_2(y))$F_{1,2}$(.) are the membership functions for Inputs 1 and 2. The final output of the system is the weighted average of all rule outputs, computed as [16]:

$$\textit{Final Output} = \frac{\sum_{i=1}^{N} w_i z_i}{\sum_{i=1}^{N} w_i}$$

2.10 Summary of Chapter

Artificial Neural Networks and Fuzzy Logic are two distinct disciplines. Artificial neuron represents the mathematical model of human neural cell, and is used to simulate the computational capabilities of human brain. Artificial neural network are applied to different Engineering problems due to its capabilities such as learning, fault tolerance, parallelism, flexibility and adaptability.

On the other hand Fuzzy logic deals with prepositions which have certainty value between ‘0’ and ‘1’.With fuzzy sets human concept can be translated to a form, usable to computers, Goal of fuzzy logic is to make machines which can respond like human brain. When fuzzy system is applied to control system, its characteristics show a faster and smoother response then those with conventional systems. Fuzzy systems are more suitable to complex systems, where conventional systems do not behave well.This chapter introduces the terms related to neural network its training, working and method to design the controller by using Fuzzy logic.

Chapter 3

Adaptive Neuro Fuzzy System

3.1 Fuzzy inference system

Fuzzy Inference System is a mechanism / technique that maps input data / characteristics to input member function, memberships to rules, rules to a set of output data / characteristics, output member function to an output or a decision (s) associated with the output. Member functions are chosen arbitrarily and are normally fixed [18]. Fuzzy inference system is applied normally to a system where rule structure is essentially pre determined by the user's interpretation based on his experience and skill [14].

3.2 Neuro Fuzzy System

Neural network can deal with imprecise data; on the other hand fuzzy logic is a tool for modeling vagueness associated with human reasoning [18]. The fusions of these techniques have improved their potential to use for complex problems and practical situations [12].

Fuzzy system can store banks of common sense/expert knowledge, if we incorporate modeling and learning capabilities of artificial neural network in fuzzy system. These two systems share with humans the abstract property of model free function estimation which is the most important characteristics of an intelligent system. Two popular techniques to fuse these technologies are as:

- Fuzzy inference systems could be constructed / incorporated with the properties like learning functions and non linearities of the neural network [18].
- On the other hand, neural networks could be constructed according to If-Then fuzzy inference rule.

Here in this study we select ANFIS with a structure similar to that of Neural Network which maps input through input member functions and associated parameters, and then through output member functions and associated parameters to output is used to interpret input/output map. Member function parameters are tuned (adjusted) using either a back propagation algorithms or in combination with a least square type method. Its parameters associated with member functions will change through learning process.

3.3 Learning and adaptive Mechanism

ANN goes through the learning and adaption process in order to improve its performance. This is achieved by weight modification or synaptic modification. In the neuro fuzzy system, in addition to synaptic modification, somatic modifications also take place. i.e neuron's structure also changes. In neuro fuzzy system, weights simply serve as mapping function that transform or modify each fuzzy input into other fuzzy sets until the training results are satisfactory [7].During learning or training process neuro fuzzy system changes its body structure i.e. changing the rule, changing the member functions assigned to the fuzzy terms in the rules, and changing the ways of representing the rules.

3.4 ANFIS Modeling Approach

ANFIS modeling approach is similar to many system identification Techniques. In first step parameterized model structure (relating inputs *to* membership functions to rules to outputs to memberships and so on) is hypothesized [18]. In the next step input/output data is collected in a form that is suited to problem and is used for training of ANFIS. Then ANFIS is trained by presenting trained data to it, i.e. during training process member function parameters are modified according to selected error criterion. Model works well if the training data presented to ANFIS for training (estimation) membership functions is a good representative of the data for which ANFIS is intended to model. However in some cases representative data is not available. In such a case approximate data is used for training.

3.5 ANFIS Model Validation

In model validation process input data from input/output data sets on which the ANFIS was not trained, is presented to the trained model, to see how well the ANFIS model predicts the corresponding data set output values. The idea behind model checking is that after training up to number of cycles the output of networks over fitting the training data set. The error tends to decrease as the training takes place up to the point that over fitting begins and then the network error for the checking data increases sharply [18].

3.6 Structure of Adaptive Neuro Fuzzy System

In Adaptive neuro fuzzy inference system, T-Sugeno model is used for generating fuzzy rules. In T-Sugeno model in each rule output can be a linear combination of input variables and a constant term or can be only a constant term [7]. The final output is obtained as the weighted average of each rule's output. For two inputs x and y and one output z basic ANFIS architecture is shown in Figure. The T-Sugeno model can be expressed as:

- Rule1: If x is A_1 and y is B_1, then $f = p\ x_1 + q\ y_1 + r$
- Rule2: If x is A_2 and y is B_2then $f = p\ x_2 + q\ y_2 + r$

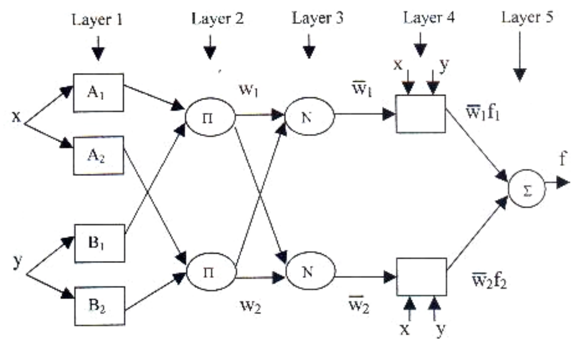


Figure 3.1: Basic structure of ANFIS [7].

ANFIS has input and output layers, and three hidden layers that represents membership functions and fuzzy rules. Theworking of input and output of each layer is summarized as follows:

- *Layer 1:* Every node i in this layer is an adaptive node with a node output defined by

$$O_{1,f} = \mu_A(x) \quad \text{for i=1,2 or} \qquad (3)$$

$$O_{1,i} = \mu_{B_{i-2}}(y) \quad \text{for i=3,4} \qquad (4)$$

Where x is the input to node i, and A_i(or B_{i-2}) is a linguistic label (such as"Small" or "large") associated with this node. Any appropriate membership function, such as the Triangular or Gaussian membership functions can be used for inputs. Parameters in this layer are called as "premise parameters".

- *Layer 2*: Every node in this layer is a fixed node labeled π, at which incoming signals are multiplies to give product at output. Each output node represents the firing strength of a r*u*le.

$$O_{2,i} = w_i = \mu_{A_i}(x)\mu_{B_i}(y), \quad \text{i=1,2}$$

- *Layer 3*: Every node in this layer is a fixed node labeled N. The *i*th node calculates the ratio of the rule's firing strength to the sum of all rules' firing strengths. Outputs of this layer are called "normalized firing strengths".

$$O_{3,i} = w_i = w_i/(w_1 + w_2) \quad \text{i= 1,2}$$

- *Layer 4:*Every node *i* in this layer is an adaptive node with a node function as:

$$O_{4,i} = \overline{w_i} f_i = \overline{w_i}(p_i x + q_i y + r_i)$$

Where w_i is a normalized firing strength from layer 3 and $\{p_i,\ q_i\ ,w_i\}$ is the parameter set of this node. Parameters in this layer are referred to as "consequent parameters".

- *Layer 5:* The single node in this layer is a fixed node labeled Σ that computes the overall output as the summation of all incoming signals:

$$\text{Overall output} = O_{5,i} = \sum_i \overline{w_I} f_i = \frac{\sum_i w_i f}{\sum_i w_i}$$

In the ANFIS training algorithm both antecedent parameters and consequent parameters are optimized. The consequent parameters are adjusted in the forward pass while the antecedent parameters remain fixed. The antecedent parameters are tuned in the backward pass while the consequent parameters are kept fixed.

3.7 ANFIS learning algorithm

From the proposed ANFIS architecture above, the output *f* can be defined as:

$$f = \overline{w_1} f_1 + \overline{w_2} f_2 =$$

$$(\overline{w_1} x) p_1 + (\overline{w_1} y) q_1 + (\overline{w_1} r_1) + (\overline{w_2} x) p_2 + (\overline{w_2} y) q_2 + (\overline{w_2} r_2)$$

which is linear in the consequent parameters [21].

The methods for updating the parameters are listed as below:

a. **Gradient decent only:** All parameters are updated by gradient decent Back propagation [20].

b. **Gradient decent and One pass of Least Square Estimates (LSE):** The LSE is applied only once at the very beginning to get the initial values of the consequent parameters and then the gradient descent takes over to update all parameters.

c. **Gradient and LSE:** This is the hybrid learning rule. Since the hybrid learning approach converges much faster by reducing search space dimensions than the original back propagation method, it is more desirable. In the forward pass of the hybrid learning, node outputs go forward until layer 4 and the consequent parameters are identified with the least square method. In the backward pass, the error rates propagate backward and the premise parameters are updated by gradient descent.Table1below summarizes the activities in each pass.

Table 1: Two passes in the hybrid learning procedure for ANFIS.

	Forward pass	**Backward pass**
Premise parameters	fixed	gradient descent
Consequent parameters	least squares estimate	fixed
Signal	node outputs	error rates

3.8 Summary of Chapter

Neural network can deal with imprecise data; on the other hand fuzzy logic is a tool for modeling vagueness associated with human reasoning. Fusion of these two approaches is believed to have characteristics of both the technique. Neuro fuzzy system can be constructed by two methods

- Fuzzy system could be constructed / incorporated the properties like learning functions and non linearities of the neural network.
- On the other hand neural networks could be constructed according to If-Then fuzzy inference rule.

In this study we have selected 2^{nd} technique which is easier to apply. This chapter describes how neuro fuzzy inference system is implemented and works based on the second technique.

Chapter 4

Crystallization Process in Industry

Formation of solid particle from a vapor as solidification of a liquid or formation of dispersed solids from a solution is known as crystallization. Crystallization requires a phase change i.e. liquid to solid. Crystallization from solution requires two component System (i.e. a Solute and Solvent) [22].Crystallization is a widely used process. We can find its applications in many industries. In petrochemical industry it is used for purification and separation. In pharmaceutical industry protein crystallization is used for drug design. Similarly in microelectronics industry it is used for silicon production and chemical vapor decomposition for manufacture of semiconductors [3]. Similarly in food industry it is used for the stability and texture issues. Although it is a widely used separation technique when the end product is desired in solid form but there are still some challenges related to the control of the crystal properties and their characteristics [4].

4.1 Related Terms

4.1.1 Solubility

Amount of a substance (solute) that can be dissolved in a given amount of solvent at a given set of pressure and temperature [22].

4.1.2 Saturated Solution

Solution that is in equilibrium, with excess of the solute present in the solution [22].

4.1.3 Super saturation

Under certain conditions (i.e. by cooling the solution, by adding anti solvent) solution can dissolve more solute than in saturated state. This solution is known to be in supersaturation state [22].

4.1.4 Solubility & Nucleation Curve

The phase relationship of the system can be illustrated by a solute concentration versus temperature diagram known as the equilibrium phase diagram [21]. In the equilibrium phase diagram, shown in Figure 4.1, there are two curves: solubility curve and nucleation curve. Graph between Temperature and Concentration of Solute determined by thermodynamics of the system is known as solubility curve. Solubility curve is influenced by the impurities in the system, which may affect the solvent activity. The solution is said to be in saturation equilibrium at the solubility curve. The second curve is the nucleation curve where the spontaneous nucleation starts. The nucleation curve is a region where the nucleation rate increases rapidly rather than a sharp boundary. These two curves divide the phase diagram in three important zones.

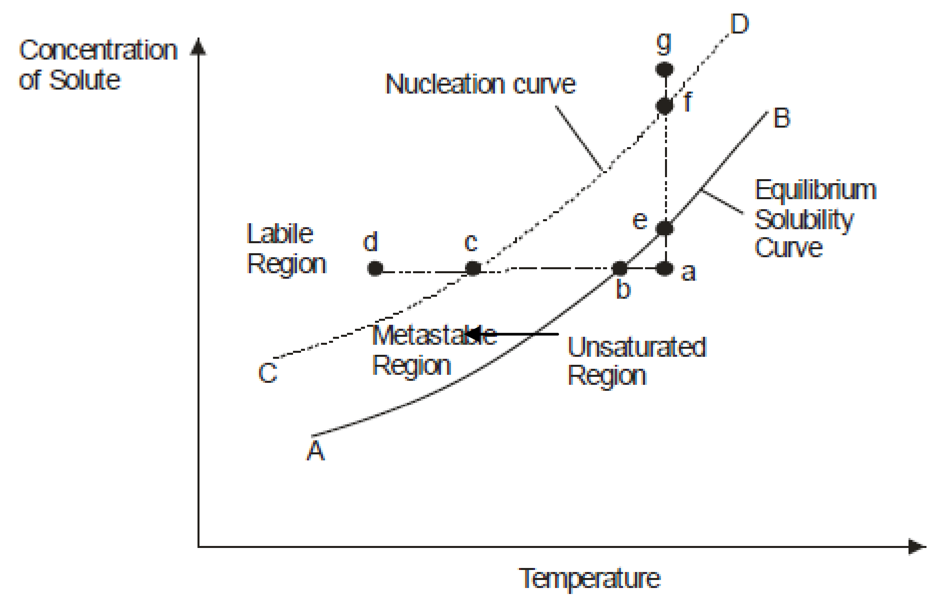


Figure 4.1: Phase Diagram of crystallization processes [21].

Under saturated zone

In this region the solution is in under saturation and crystals will dissolve (region below solubility curve) if present. The dissolution rate of disappearing crystals depends on the degree of under saturation, which is expressed similar to the super saturation [21].

Meta stable zone

This is the region in which crystals will grow (region that lies in between the equilibrium solubility curve and nucleation curve) with a rate defined by the level of super saturation [21]. The meta stable limit is not very well defined [] and depends on a number of parameters such as rate of generating the super saturation, temperature, solution history, impurities and fluid dynamics. The Meta stable Zone Width (MSZW) may vary to different extents for different systems and is said to be the point after which continuous nucleation occurs. Seed crystals would grow within the MSZW but no significant amount of new nuclei should form. MSZW is therefore an important property in assessing the tendency of a system to crystallize and in deciding the crystallization technique.

Labile or unstable zone

This is the region in which spontaneous nucleation starts. In this region crystal formation starts with nucleation.

Driving force for crystallization

Driving Force for Crystallization process is super saturation. Optimum crystallization processes can only be accomplished if the **MSZW** is known and saturation level is controlled during the entire process [21].

4.2 Super saturation Formation

Super saturation can be created in crystallizers by different types. These are briefly defined below.

4.2.1 Cooling Method

The most widely used method is by cooling a solution through indirect heat exchange. This approach is chosen when the solubility of the solute decreases significantly with temperature and hence the solution becomes supersaturated [22]. In Figure 4.1, cooling trajectory is shown by path (abcd). Starting from point 'a' in the under saturated region. The equilibrium solubility curve is crossed at point 'b' and enters into the meta stable region. As soon as the system crosses the solubility curve and enters the meta stable region, the solution becomes supersaturated. In Figure 4.1, crystallization will not start until it has been sub-cooled to a point on the nucleation curve 'c'.

4.2.2 Evaporation Method

Evaporation of solvent is another way to generate super saturation in the system and is used when the solute has weak dependency of solubility on temperature [22]. Concentration of the solute in the solution is increased by gradually removing solvent from the solution by evaporation. In Figure 4.1, evaporation trajectory is shown by (aefg). Starting from point in the under saturated region, the equilibrium solubility curve is crossed at point and enters into the meta stable region by slowly removing the solvent by evaporation. Crystallization will not start until the concentration reaches point 'f'on the nucleation curve by evaporating the solvent.

4.2.3 Addition of Anti Solvent

Another way to create super saturation in crystallization process is to add an extraneous substance, generally known as anti-solvent [21]. The disadvantage of this technique is an added unit for the separation of this extraneous material, which can add complexity to the solution and increase cost.

4.3 Crystallization process Development

Determine the solubility of a substance in solvent i.e. determine the solubility curve for that particular system. For crystallization the solution should be in a supersaturated state at the process temperature. Otherwise, the system is under saturated and solids will dissolve. Select the method to induce super saturation in the system. The most common approach is either to follow a cooling profile in time, to use evaporation of the solvent or the addition of a poor solvent (anti-solvent). Some of these techniques can also be combined together to induce super-saturation in a system. Another attractive option includes chemical reaction (precipitation). Determination of the meta stable zone width in which crystals will grow and nucleation can be avoided.

4.4 Mechanism of Crystallization Process

Following particle formation processes occur during the crystallization:

4.4.1 Nucleation

Nucleation is the primary particle formation process during which particles of a solid come close to each other to form clusters in an arranged manner [22]. Nucleation determines the initial formation of crystals, and crystal growth, which determines the subsequent size [4].

4.4.2 Growth and Dissolution

The subsequent growth of solute particles is called growth. The newly born nuclei grow with time [22]. Two successive steps are required for crystal growth i.e. solute molecules transport from the solution to the crystal surface by diffusion, convection or the combination of both mechanisms and incorporation of the material into the crystal lattice through surface absorption, also described as surface reaction step. The second step is further subdivided into a number of sub steps which are as follows:

- Adsorption of the growth unit on the crystal surface.

- Release of part of its solvation shell, followed by the diffusion of growth unit into the adsorption layer until it is either incorporated into the lattice or leaves the adsorption layer and returns back into the solution [22].

4.4.3 Aggregation and Breakage

Aggregation is a further growth process, which joins fines in an assembly [3-5]. The particle characteristics obtained in the product depend strongly on the mechanism of aggregation. Aggregation results in relatively rapid size enlargement [3-5]. Collision of particles with themselves and the walls of the container and impeller results in breakage of bigger crystals. Crystal formation can also occur via particle breakage processes that start with existing particles and forms new smaller ones of varying sizes.

4.4.4 Dissolution

Dissolution is decomposition of crystals. This is reverse of the crystallization, which occurs in the under saturated region (shown in Figure 4.1) [3-5]. The concentration of the solute increases as dissolution proceeds. If given enough time at fixed conditions, the solute will eventually dissolve up to a maximum solubility where the rate of crystallization equals the rate of dissolution.

4.5 Solubility Curve & Crystallization Process

The set point supersturation is chosen so that there is compromise between fast crystal growth and low nucleation rate to obtain required crystal characteristics [3-5]. If set point is chosen close to the meta stable limit (high super saturation) this results in excessive nucleation, lower purity and longer filtration times. On the other hand if set point is chosen close to solubility curve (low super saturation) this leads to slow growth and long batch times.

4.6 State Variables for Crystallization Process

State variable for crystallization process are as [3-5]:

- Concentration of the solution.
- Process temperature.
- Jacket temperature.
- Crystal Size Distribution (CSD**).**

4.7 Operating Conditions for Crystallization Process

- Total batch time.

- Initial concentration.
- Seed mass size.
- Supersaturated set point [4].

4.8 Modeling of Crystallization Process

Batch crystallization is a highly non-linear process. It contains a large number of time-varying kinetic and transport parameters. Therefore it cannot be modeled over a broad range of operating conditions using linear models [3], and classical linear control theory cannot be used for controller synthesis. Therefore complex nonlinear control techniques and dynamic optimization approaches are required for batch crystallization control and optimization.

4.8.1 Crystallization process is modeled by using

- Population Balance Equation (Solution by Analytic Techniques) [22].
- Conservation Laws [22].
- Kinetic relations (Annex 'A') of physical phenomena occurring in crystallization process [3].

4.9 Systematic direct design approach to obtain Temperature Trajectories

Batch crystallization processes is normally carried out by manipulating the temperature trajectories. A desired shape of crystal size distribution is obtained by optimizing set point super saturation in MSZW (Meta stable zone width)[4]. A solution of population Balancing equation by combined **MOCH-QMOM** provides a direct design approach for super saturation controlled crystallization process [3-5].

$$\frac{\partial f_n(L,t)}{\partial t} + \frac{\partial \left| G(S,L,\theta_g) f_n(L,t) \right|}{\partial L} = B(S,\theta_b)\delta(r_o,L) \qquad (1)$$

In above equation f_n(L,t) is crystal size distribution expressed in number of crystals / Kg slurry, t is time in seconds, *"S"* is absolute concentration and C is solute concentration in Kg solid / Kg slurry, $G(S,L;\theta_g)$ is rate of crystal growth in Kg solid/ Kg slurry, $B(S,L;\theta_b)$ is the rate of nucleation number /s / Kg slurry, C_{sat} is Saturation concentration with T is temp in ℃, $\delta(r_0,L)$ is Kronecker delta, $\theta_g$is the vector of growth kinematics, θ_b is the vector of nucleation kinematics and $r_0$is the size of nuclei.

The solution of equation (1) is an initial value problem with initial condition given by the size distribution of the seed [3],

$$f_{n(L,0)} = f_{n,0}(L_0)$$

For the generic case of size dependent growth, for which the kinetics are given by,

$$G = k_g S^g (1 + \gamma L)^p$$

Where $\theta_g = [k_g, g, \gamma, b]$ is the growth parameter vector.

For seeded crystallization, secondary nucleation is considered as the dominating nucleation phenomenon, which is generally expressed as a function of the supersturation and the volume of the existing crystals, given by the third-order moment of the size distribution. Hence in the model the empirical relationship for secondary nucleation is given by [3],

$$B = k_b S^b \mu_3$$

Where $\theta_b = [k_b, b]$ is the nucleation parameter vector.

Where μ_3 is the third moment. This can be obtained by applying Quadrature Method of Moment QMOM to Population Balance Equation. It employs a quadrature approximation of the distribution function as given by [3]:

$$f_{n(L,t)} \approx \sum_{i=1}^{N_q} w_i(\mathrm{t})\delta\big(L_i(\mathrm{t}), \mathrm{L}\big)$$

Where N_q is number of quadrature points, corresponding weights w_i and abscissas L_i can be determined by Product Difference (PD) Algorithm or through direct solution of differential algebraic DAE system. Moments are given by equation [3]:

$$\mu_j = \int_0^\infty f_n(L) L^j dL \approx \sum_{i=1}^{N_q} w_i L_i^j$$

Applying moment transformation to equation (1) the resulting equations have the form

$$\frac{\boldsymbol{d\mu_0}}{\boldsymbol{dt}} = B(S, \theta_b) \qquad \text{i.e.}$$

$$\frac{\boldsymbol{d\mu_j}}{\boldsymbol{dt}} = \boldsymbol{j} \sum_{i=1}^{N_q} w_i L_i^{j-1} G(S, L, \theta_g) + B(S, \theta_b) r_0^i, \quad \text{j=1,2,3,...}$$

L_i is different than L length of the particles and is used to calculate abscissas only.

The generic PBE equation (1) can be reduced to a system of ODEs by applying the method of characteristics (MOCH). The aim of the MOCH is to solve the PBE by finding characteristic curves in the plane L-t. Applying the MOCH, equation (1) is reduced to the following system of two ODEs [3]:

$$\frac{\boldsymbol{dL}}{\boldsymbol{dt}} = G(S, L, \theta_g) \tag{2}$$

$$\frac{d f_n(L,t)}{dt} = -f_n(L,t)\frac{d|G(S,L,\theta_g)|}{dL} + B(S,\theta_b)\delta(r_o, L) \tag{3}$$

In the case of well controlled constant super saturation, which follows the desired set-point value [3], $S_{sp}$the system (2)-(3) can be solved analytically with the solution given by [3],

$$L = \left(\left((1+\gamma L_0)^{1-p} + k_g S^g t\gamma(1-p)\right)^{\frac{1}{1-p}} - 1\right)/\gamma$$

$$f_n = f_{n,0}(L_0)\left(1 + \frac{k_g S^g t\gamma(1-p)}{(1+\gamma L_0)^{1-p}}\right)^{\frac{p}{p-1}}$$

The optimization problem is formulated as follows:

$$min_{\alpha_T(j),t_f} \sum_{k=1}^{K}\sum_{l=1}^{N_d}\left(f_{v,k}(L_l) - f_{v,k}^{target}(L_l)\right)^2$$

Subjected to

$$\alpha_{T,min} \le \alpha_T(j) \le \alpha_{T,max}\ ,\ j = 0,1,\dots,N_d$$

$0\le t_f \le t_{f,max}$

$$C(t_f) \le C_{f,max}$$

Where, t_f is the total batch time,$\alpha_T(j)$are the elements of the vector containing the slopes (dT/ dt) for the temperature trajectories depending on the implementable heating and cooling capacity of the system, $C(t_f)$is the solute concentration at the end of the batch,$C_{f,max}$is the maximum acceptable concentration at the end of the batch to achieve the required yield, $f_{v,k}$and$f_{v,k}^{target}$ are the values of the simulated and the target volume probability distribution functions at the discrete time steps k =1,2,…K, where measurement data was available corresponding to the discretized sizes L_l, l=1,2…,N_d.with$N_d$being the number of experimental size bins[3].

By this approach super saturation trajectories can be defined in terms of temperature trajectories. Temperature trajectories in time can be designed for a desired super-saturation set point to obtain the required CSD.

4.10 Crystallization Process in Industry

In industry crystallization process is performed by manipulating the super saturation trajectory due to its relation with crystallization phenomena e.g. nucleation, growth etc. The key concern in industrial batch processes is to maximize the production efficiency while improving the quality and consistency of the final products. Crystal growth rate is a key process variable having a close relations with most of product quality aspects. So there is a tradeoff between product quality requirements and maximization of batch throughput [2]. Due to this, it is optimal control problem such that an upper limit is met throughout the batch run.

4.11 Current Control Approaches [21]

- Direct Design Approach.
- Model Based design Approach.

4.11.1 Direct Design Approach.

In direct design approach, crystallization processes is controlled to follow predetermined profiles in time. These profiles are chosen arbitrarily or by trial and error experiments [21].

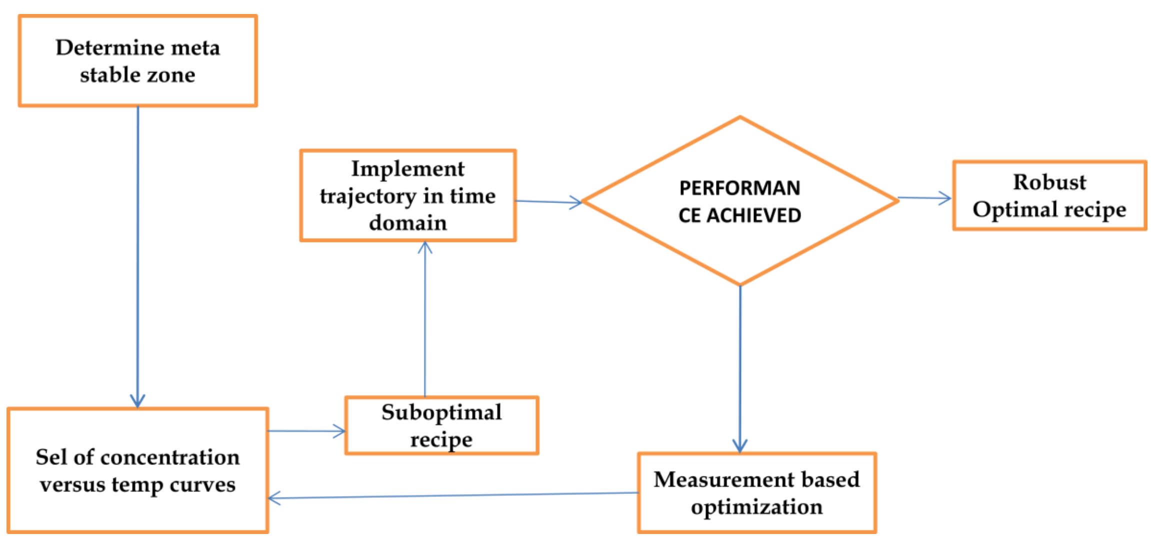


Fig 4.2: Direct Design Approach Representation [21].

4.11.2 Model-based Design Approach

In model-based approach the operating conditions are optimized based on model predictions [21]. Model based approaches can be divided in to two categories:

4.11.2.1 Off Line Model Based Design Approach

In this technique temp versus time or anti solvent addition rate versus time trajectories are optimized off-line. These resulting optimal operating policies (e.g. temperature versus time or anti-solvent addition rate versus time profiles) are then implemented using simple feedback tracking control systems.[2] This is not an effective optimal control strategy as the performance is often deteriorated due to plant-model mismatch, unmeasured process disturbances and irreproducible start-ups, i.e. unknown initial conditions [2].

4.11.2.2 Online Model Base Control Approach

"In this technique the optimal operating policies can be computed in an online mode. An Optimization-based strategy in conjunction with a state estimator is utilized to compute the optimal operating policy in a feedback control framework where the effects of the plant model mismatch and process uncertainties are accounted for by continuous state adaptation. In this method an upper constraint on the crystal growth rate that should be fulfilled at all times during the batch without jeopardizing the product quality is achieved." [2].

4.12 Summary of Chapter

Crystallization is widely used technique in different chemical industries and is often used in pharmaceuticals and fine chemicals. Super saturation derives the crystallization process and crystal properties i.e. growth, size and purity depends upon it. Super saturation is created by different methods such as cooling, evaporation, chemical reaction and anti-solvent addition. Determination of solubility curve, nucleation curve and meta stable zone width is of key importance for development of crystallization process. After achieving super saturation, crystallization is a two-step process. The first step is phase separation known as nucleation and then the growth of nuclei to crystal. Crystallization process is modeled by Population Balance Equation. Population Balance Equation is a material balance that accounts for the distribution in particle size, location and other variables. Population Balance Equation can be solved by number of method. Each method has its own advantages and disadvantages, but no classical method provides the direct relation between the temperature and time for set saturation point to get desired crystal characteristics. In this chapter latest technique to solve the Population Balance Equation is reviewed. This technique uses the combine Quadrature Method of Moments QMOM and Method of Characteristics MOCH. This technique provides a method to design the set point trajectory for the super saturation trajectory so that a target CSD can be obtained, for this saturation trajectory the relation between temp and time is obtained. This relation is used to design to design the controller.

Chapter 5

ANFIS based Control System Design

5.1 Control System for Super-saturation Controlled Crystallization Process by using ANFIS

Main state variable for crystallization process includes solution concentration, process temperature and jacket temperature [3-5]. Due to non-availability of measurement devices for all process state variables (i.e. super saturation measurement sensors are at prototype stage and are often not available for industrial scale use) and non- linear dynamics of process make it difficult to control [4]. Currently control systems for crystallization process are designed either by using direct design or model bases approaches [2]. In direct design approach, process is to follow predetermined temperature profiles in time which are chosen arbitrarily or by trial and error experiments and depend upon the dynamics of the plant [21]. While in model based approach operating conditions are optimized by using measurable data and predictions of unmeasured data based on model [2]. Main problem is the need for computing the solution of the optimization problem within the sampling period [4].

Systematic Design Approach described in Chapter 4 provides a direct relation between time and temperature for the control of crystallization process. Due to the availability of accurate measuring equipment for time and temperature with quick response time, control strategy based on this approach will improve the process control. In order to devise a control system for this strategy, temperature trajectories with respect to time are designed offline by using Systematic Design Approach [4] and ANFIS is used to generate the control actions to compensate for the errors due to the plant model mismatch, measurement errors or errors due to process uncertainties etc [6].The proposed model for the control system with ANFIS is shown in figure 5.1.

5.2 Control Mechanism

- Temp is controlled to maintain supersaturation level throughout process.
- Temperature is measured throughout the process and compared with desired value.
- Error and change in error are the inputs of ANFIS and output gives the control signal to compensate the error.

- Control action for the plant consist of two components i.e Rate of change of Temp dT/dt (the rate at which temperature is changed during the process the rate at which temperature is changed during the process)required for next step & action to compensate the error $e_c$in the last measurement step.
- Total Control Action is as

$$u = \frac{dT}{dt} + e_c \tag{5.1}$$

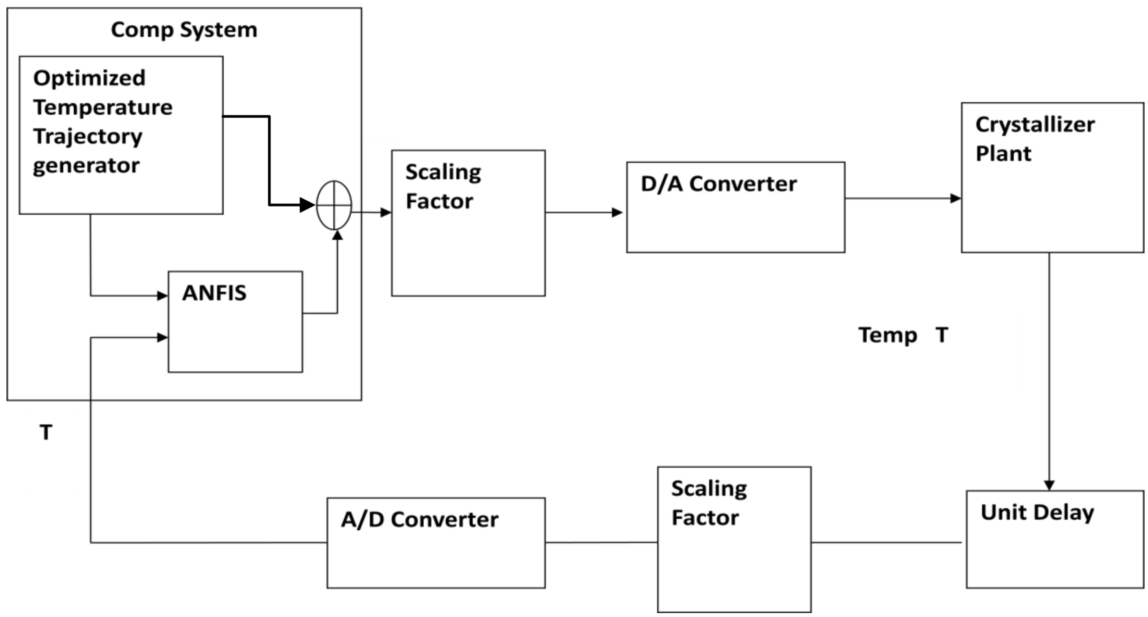


Figure 5.1: Control System for the implementation of desired temperature trajectory for batch crystallisation process.

5.3 Working of Controller

- Plant Temperature is measured by using temperature sensor (PT-100).
- Output of sensor is scaled so that it maps the measured value to universe of discourse.
- Scaled value is converted into digital value by using A/D converter.
- Error and change in error is measured.
- ANFIS calculates the required control action to compensate the error based on error and change in error.
- Control action is scaled to map with control hard ware.

- Scale signal is converted to analog signal by A/ D converter and applied to control the plant.

5.4 Role of ANFIS

- To estimate the error due to
  - Plant – model mismatch
  - Plant uncertainties due to complex nonlinear nature of the Process
  - Plant dynamics including stirring speed, control of temp, measurement uncertainty.
- To generate control action to compensate the error due to all sources.

5.5 Design of ANFIS

To obtain the designed temperature trajectory, temperature is measured throughout the process. Let *'e'* be the error in temp measurement (i.e. difference between desired value and actual value of measured temperature) due to all error sources and 'C_e' is the time rate of change of error. *'e'* and 'C_e' are taken as input of ANFIS. The function of ANFIS is to compensate the error effect. The output of ANFIS is equal to negative of sum of *'e'* and 'C_e' and is a real number.

For **fuzzification** of input values 'e' and 'C_e'i.e. input variables are converted into fuzzy variables or the linguistics variables. The fuzzification maps the 2 input variables to linguistic labels of the fuzzy sets described by linguistic terms as

'e' and 'C_e' ={NB,NS,ZE,PS,PB}

Where NB is negative big, NS is negative small, ZE is zero, PS is positive small and PB is negative big. Triangular member functions are used to identify the meaning of linguistic values. The inputs are fuzzified using the fuzzy sets and are given as input to ANFIS controller. The values of these linguistic terms are selected / adjusted by ANFIS automatically during training process. Triangular member functions for error and change in error are given in figure 5.2 &5.3.

Structure of ANFIS: Structure of ANFIS is similar to structure of neural network. It consists of five layers. Structure is shown in figure 5.4.Inputs are applied to Ist layer, output of this layer maps the input value to member function. 2nd layer has ten nodes equal to number of linguistic variable for both the inputs.

This layer gives the firing strength of a rule. Third layer has twenty five nodes equal to total number of rules. This layer calculates the ratio of rule's firing strength to the sum of all rule's firing strength.

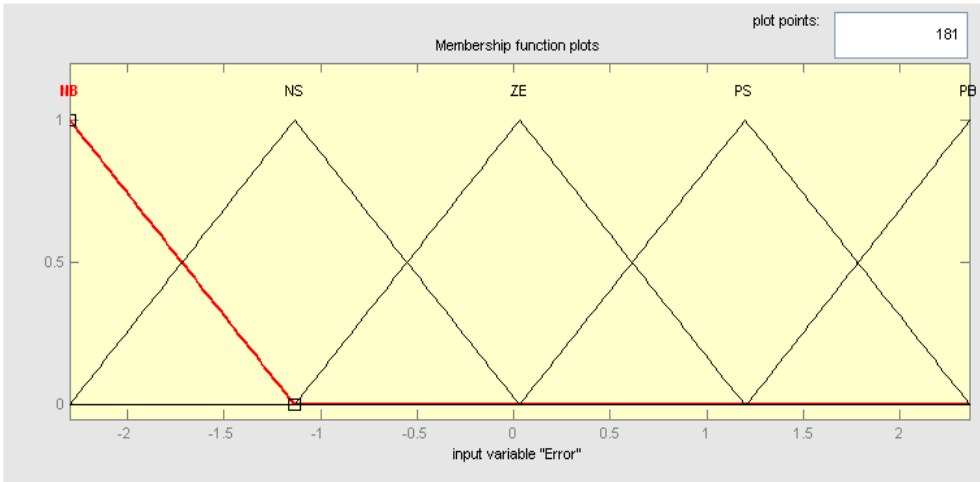


Figure 5.2: Membership Functions for Error

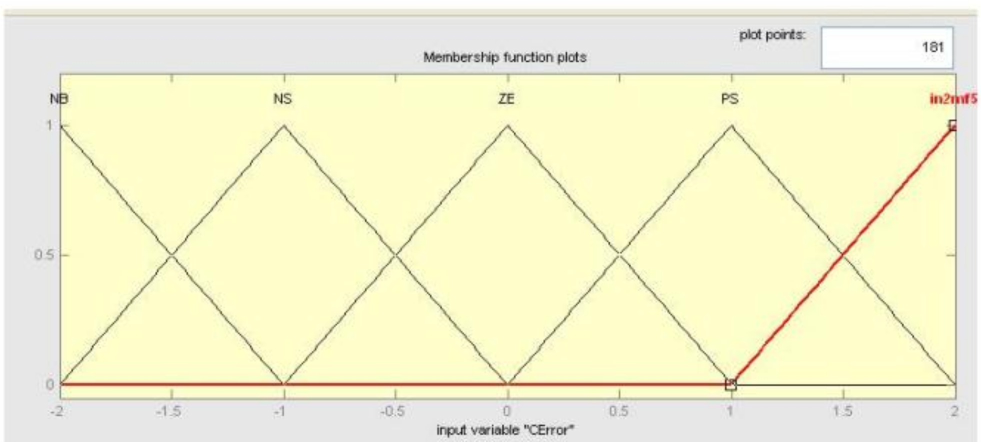


Figure 5.3: Membership Functions for Change in Error

4^{th} layer has also 25 nodes and is known as inference layer .5^{th} layer consists of only one node and produce the output after defuzzification.

Rule Generation: Takagi-Sugenofuzzy model is used for generation of fuzzy rules [7, 8,9]. In Takagi –Sugeno model, the output of each rule can be a linear combination of input variables plus a constant term or can be only a constant term. The final output is the weighted average of each rule's output. The Takagi-Sugeno rule base can be expressed as:

Rule1: If x is $A1$ and y is $B1$, then f= $p\, x_1 + q\, y_1 + r$

Rule2: If x is A_2 and y is B_2, then $f = p\, x_2 + q\, y_2 + r$

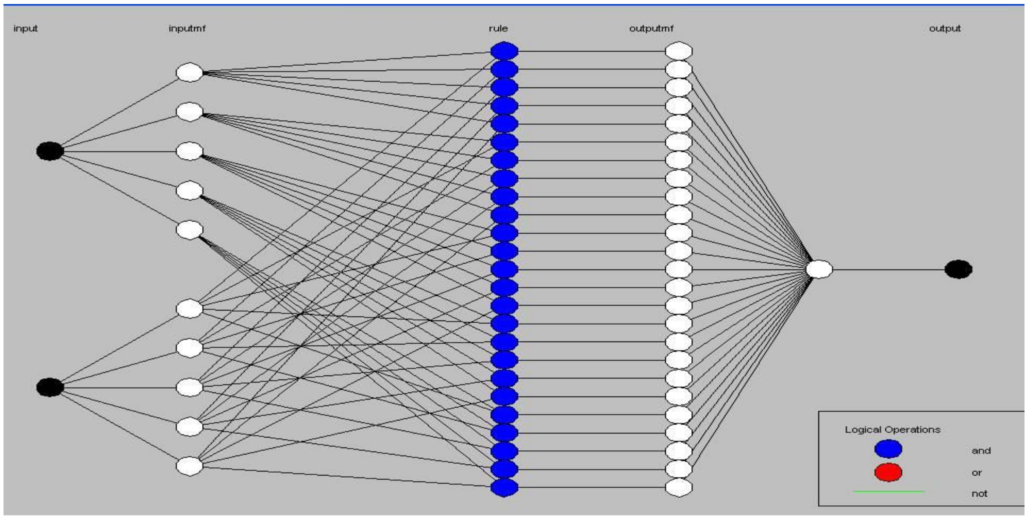


Figure 5.4: ANFIS model structure showing all the 5 layers in the ANN architecture

As in this case output is a real number (constant value) so the Takagi-Sugeno model of zeroth order is chosen for this study.

Inference System: Product (prod) and probabilistic OR (probor) functions are used for inference system.

Defuzzification is the process in which fuzzy value is converted into crisp value. For this study weight average method is used for the defuzzification.

In the ANFIS training, Hybrid Learning (combination of back-propagation and the least-squares method) algorithm is used which optimizes both antecedent parameters and consequent parameters. The consequent parameters are adjusted in the forward pass while the antecedent parameters remain fixed. The antecedent parameters are tuned in the backward pass while the consequent parameters are kept fixed.

5.6 Training of ANFIS

ANFIS is trained by choosing arbitrary values for input variables. During the training ANFIS adjusts the values of linguistic terms for input variables and generate the fuzzy rule base.

5.7 Summary of Chapter

This chapter describes the design and working of ANFIS based controller for control of temperature trajectories for super saturation controlled crystallization process. Neuro fuzzy inference system plays a very important rule in controller it estimates the error

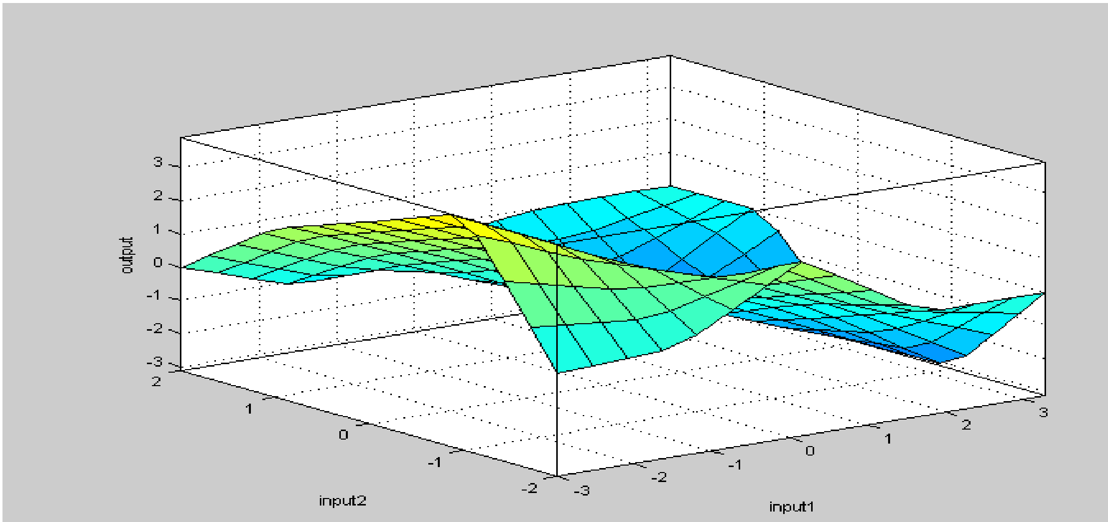


Figure 5.5: Decision surface for the ANFIS

due to all types of uncertainties and generate the control action to compensate the error due all types of errors. Accuracy of the controller depends upon the working of neuro fuzzy inference system. This chapter also covers the design of neuro fuzzy system. In the end the training of ANFIS is also discussed.

Chapter 6

Result and Discussion

The aim of the study is to design ANFIS based controller to control temperature trajectories for set point super saturation crystallization process and to study the performance of controller. ANFIS architecture is used to estimate error due to all type of uncertainties during crystallization process, estimate the control action to compensate the error. In this chapter, simulation results showing the controller performance with ANFIS are presented.

6.1 Generation of Training Data

Training data set should be in the operating input-output data range. If the operating input-output data are outside their training data range, estimator will not operate accurately. As a result, the training data set should possess sufficient operational range including the maximum and minimum values for input-output variables. The data set should include data for each process variable, evenly distributed throughout the range for which estimation is desired.

6.2 Testing of Controller

An optimized temperature profiles (linear, cubic, and bimodal) are generated by using direct design approach and is stored in the controller memory. Noise / error is added in the temperature profiles by generating random numbers from different distributions. Model of the proposed system is checked / verified by performing simulations against the noisy data. The results for different temperature profiles are discussed below.

6.3 Simulation Results and Discussions

6.3.1 Case 1

In this case error is added by generating random numbers in the range (-1 ~ 1) and (2~ 2) in three types of temperature profiles. Noisy data is presented to controller where data is compared with required value / data, as a result the error and change in error is calculated. These error and change in error are given to ANFIS as its input. ANFIS calculate the control action to compensate the error. Overall control action is calculated by using equation 5.1 given below

$$u= \frac{dT}{dt}+ e_c$$

Simulation results for this case are shown in figure 6.1 ~ 6.6. In each figure three types of profiles are shown i.e. Actual Temperature profile (Reference Profile).Noisy temperature profile (input to ANFIS). Result shows that output obtained by using proposed system compensates the effect of error.

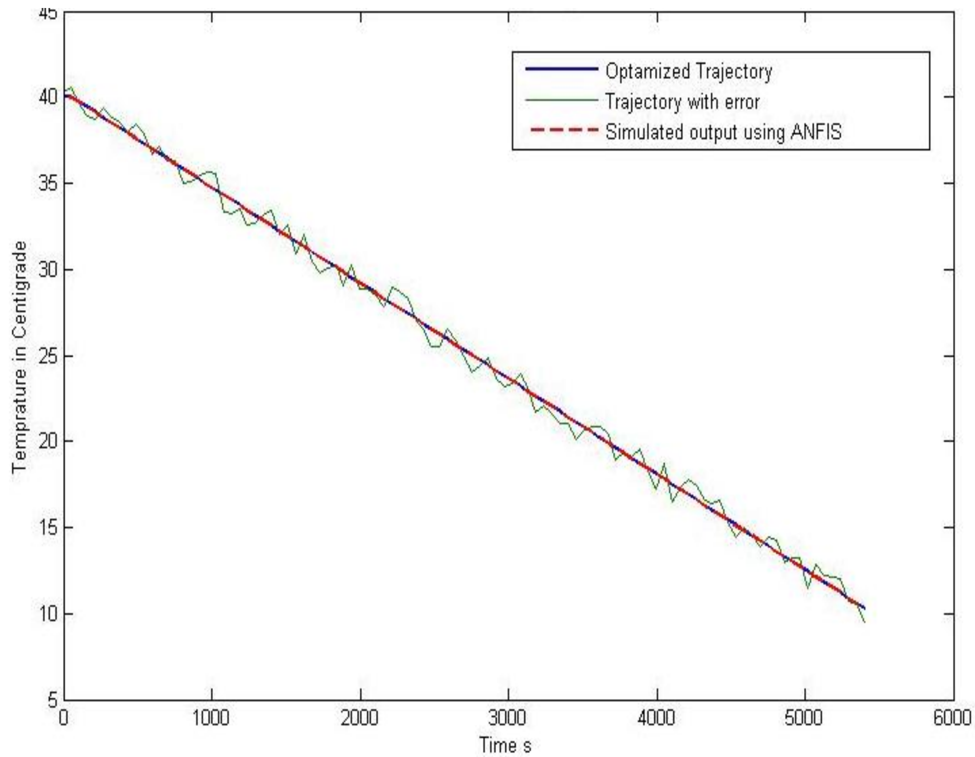


Figure 6.1: Linear temperature profile for bimodal distribution with noise has random values between 1~ -1.

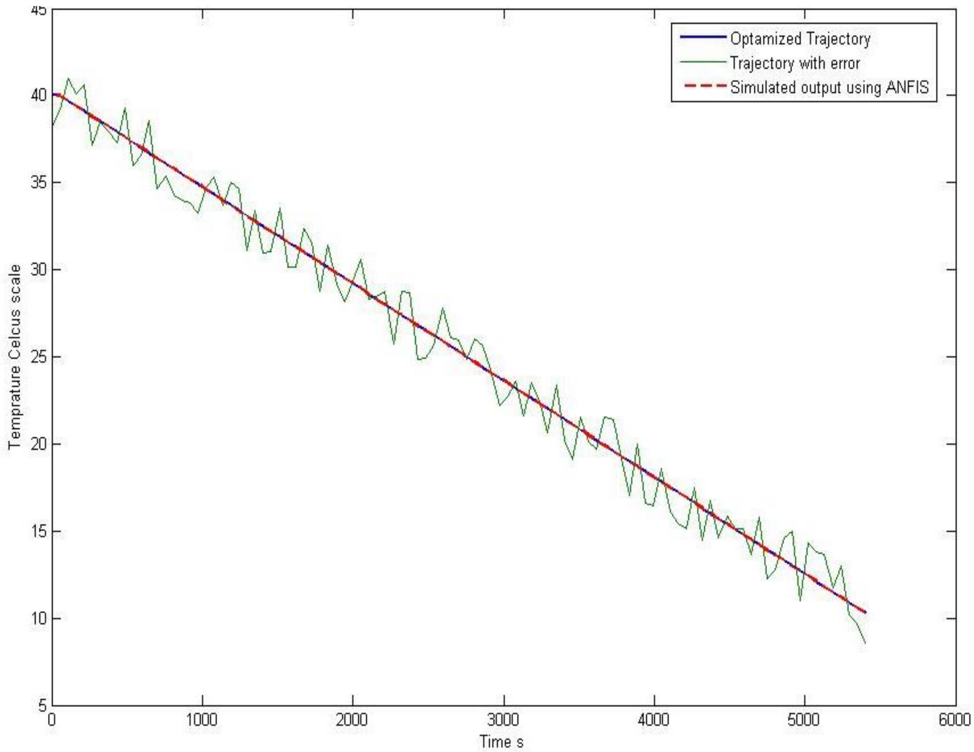


Figure 6.2: Linear temperature profile for bimodal distribution with noise has random values between2~ -2.

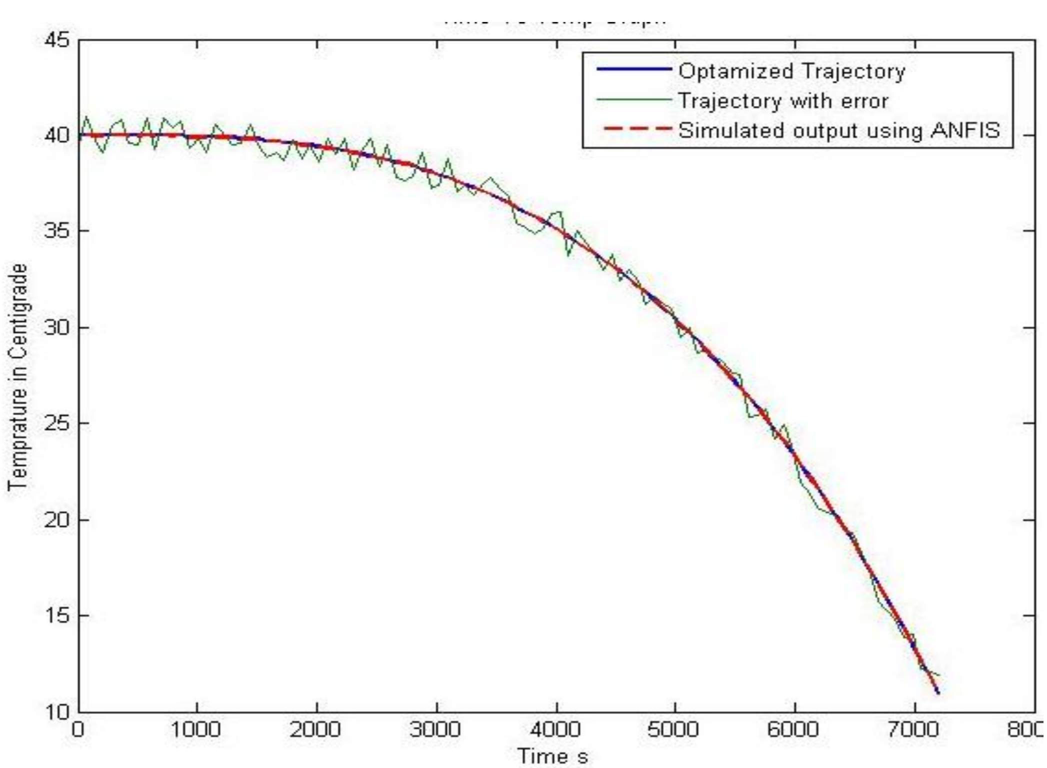


Figure 6.3: Cubic temperature profile for mono-modal distribution with noise has values between 1~ -1.

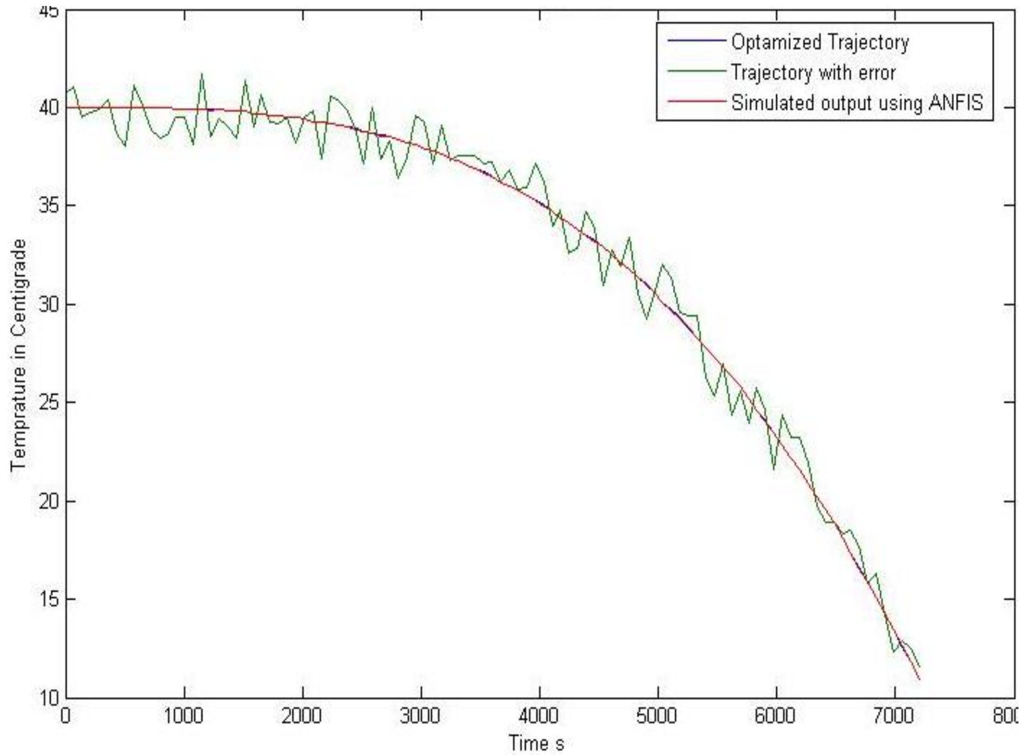


Figure 6.4: Cubic temperature profile for mono-modal distribution with noise has random values between 2~ -2

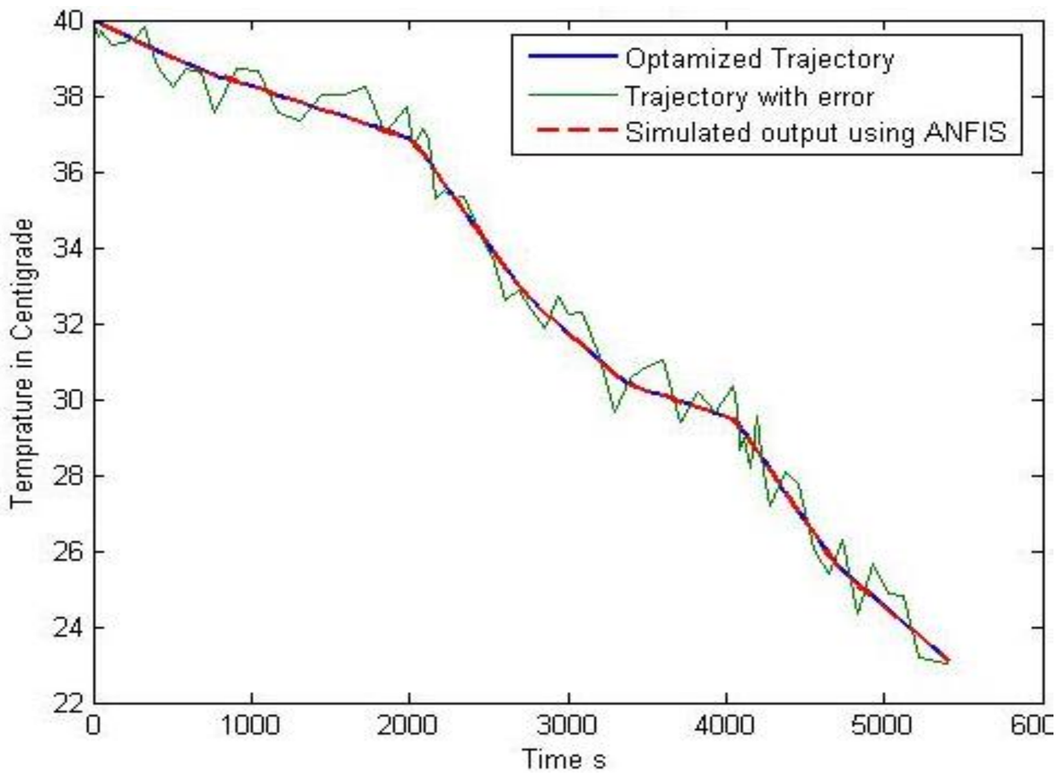


Figure 6.5: Temperature profile for bio-modal distribution with noise has random values between 1~ -1.

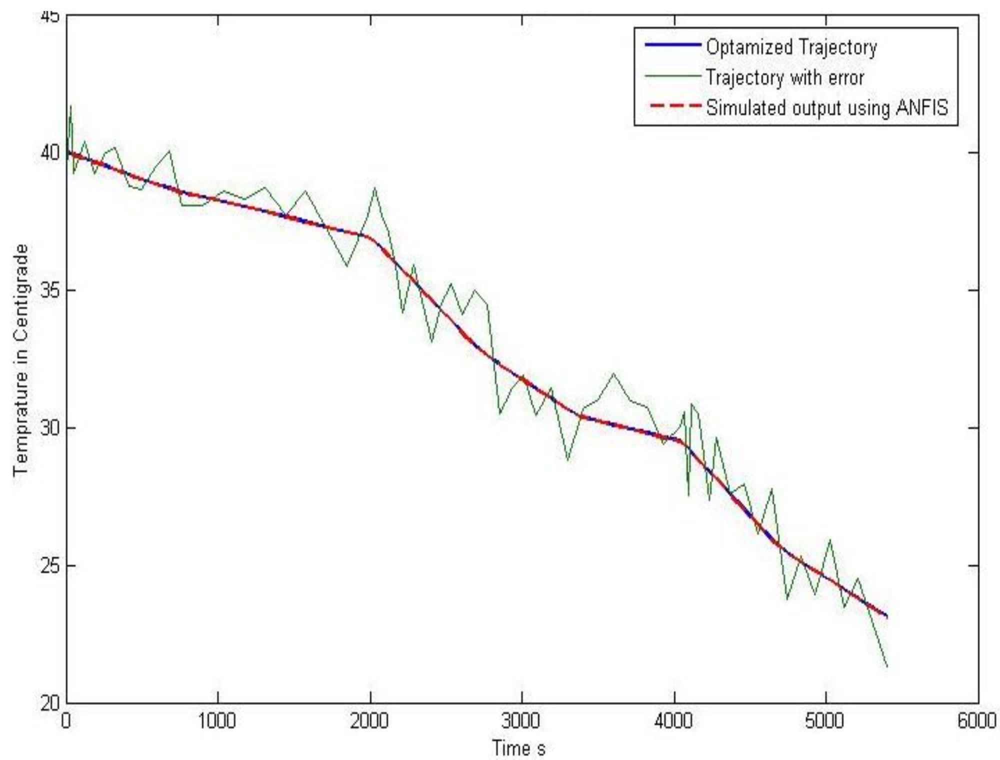


Figure 6.6: Temperature profile for bio-modal distribution with noise has random values between 2~ -2.

6.3.2 Case 2

In this case error is added by generating random numbers from Normal and Uniform Distribution linear and cubic temperature profiles. Noisy data is presented to controller where data is compared with required value / data, as a result the error and change in error is calculated. These error and change in error are given to ANFIS as its input. ANFIS calculate the control action to compensate the error. Overall control action is calculated by using equation 5.1 given below

$$u= \frac{dT}{dt}+ e_c$$

Simulation results for this case are shown in figure 6.7 ~ 6.10. In each figure three types of profiles are shown i.e. Actual Temperature profile (Reference Profile).Noisy temperature profile (input to ANFIS). Result shows that output obtained by using proposed system compensates the effect of error.

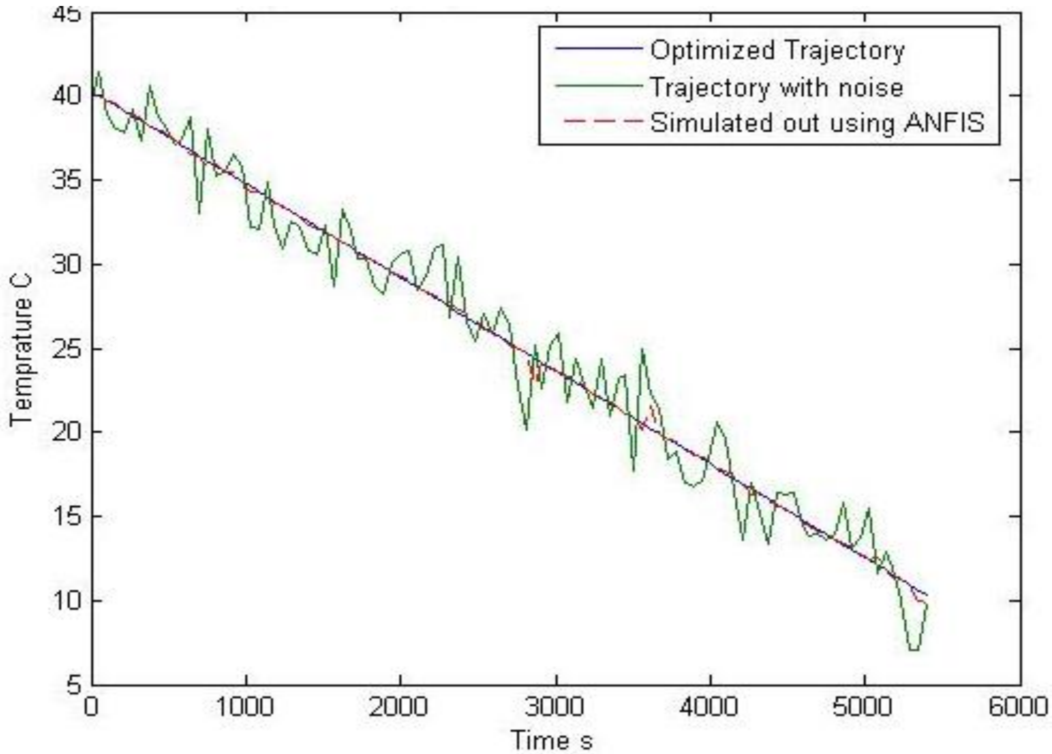


Figure 6.7: Temperature profile for linear model distribution with noise belongs to normal distribution.

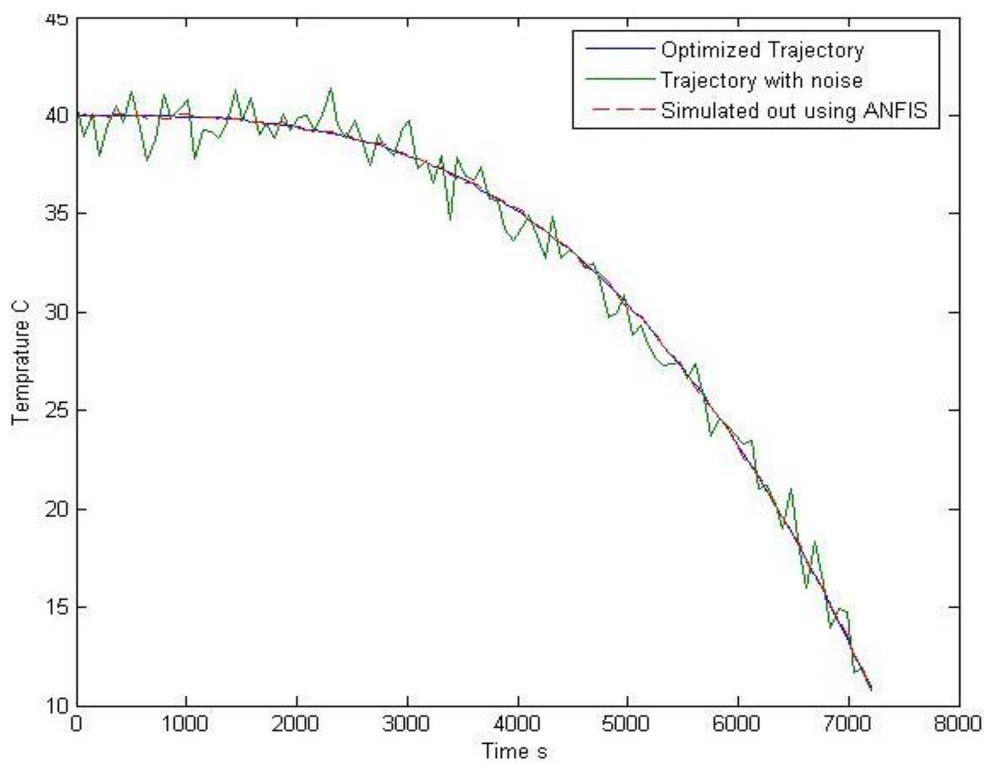


Figure 6.8: Temperature profile for cubic model distribution with noise belongs to normal distribution.

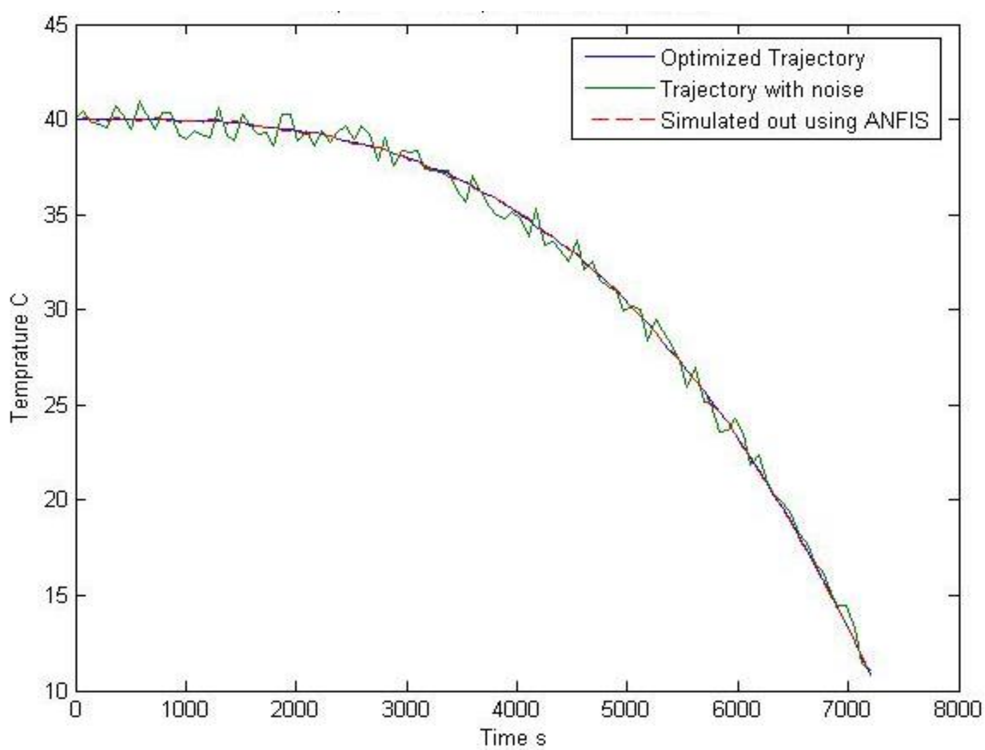


Figure 6.9: Temperature profile for cubic model distribution with noise belongs to uniform distribution.

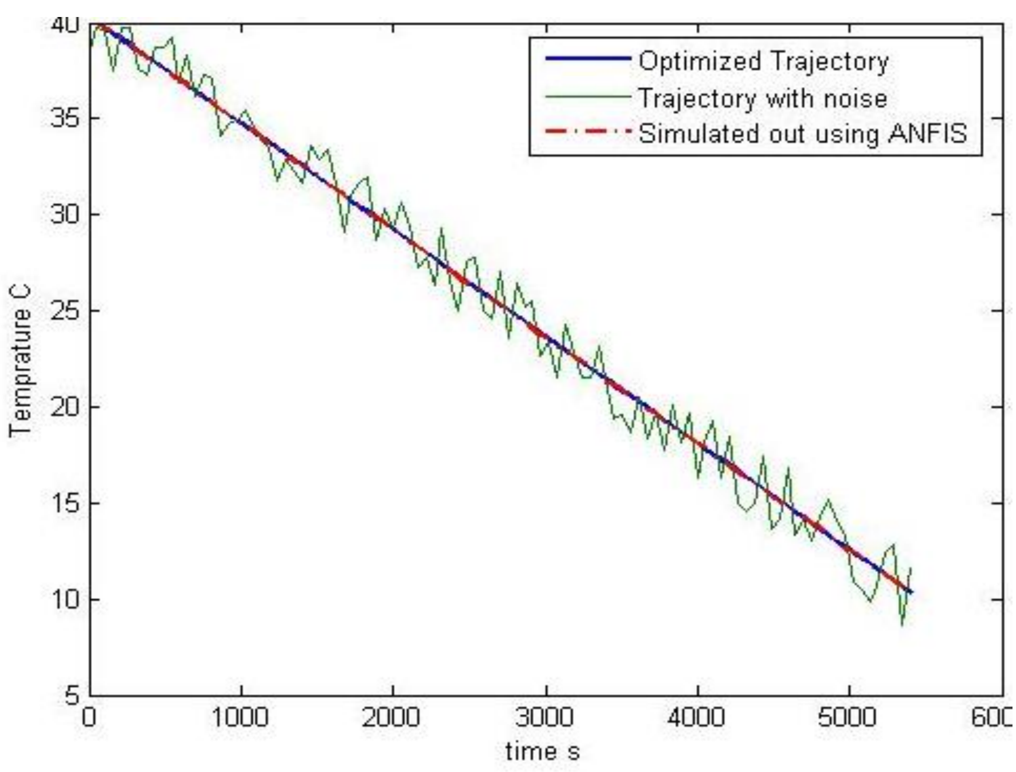


Figure 6.10: Temperature profile for linear model distribution with noise belongs to uniform distribution.

6.3.3 Case 3

In this case error is added by generating random numbers from Poisson distribution linear and cubic temperature profiles.

Noisy data is presented to controller where data is compared with required value / data, as a result the error and change in error is calculated. These error and change in error are given to ANFIS as its input. ANFIS calculate the control action to compensate the error. Overall control action is calculated by using equation 5.1 given below

$$u= \frac{dT}{dt}+ e_c$$

Simulation results for this case are shown in figure 6.11 ~ 6.12. In each figure three types of profiles are shown i.e. Actual Temperature profile (Reference Profile).Noisy temperature profile (input to ANFIS). Result for Bio model distribution did not follow the actual trajectory; this can be improved by adding data points in the training set.

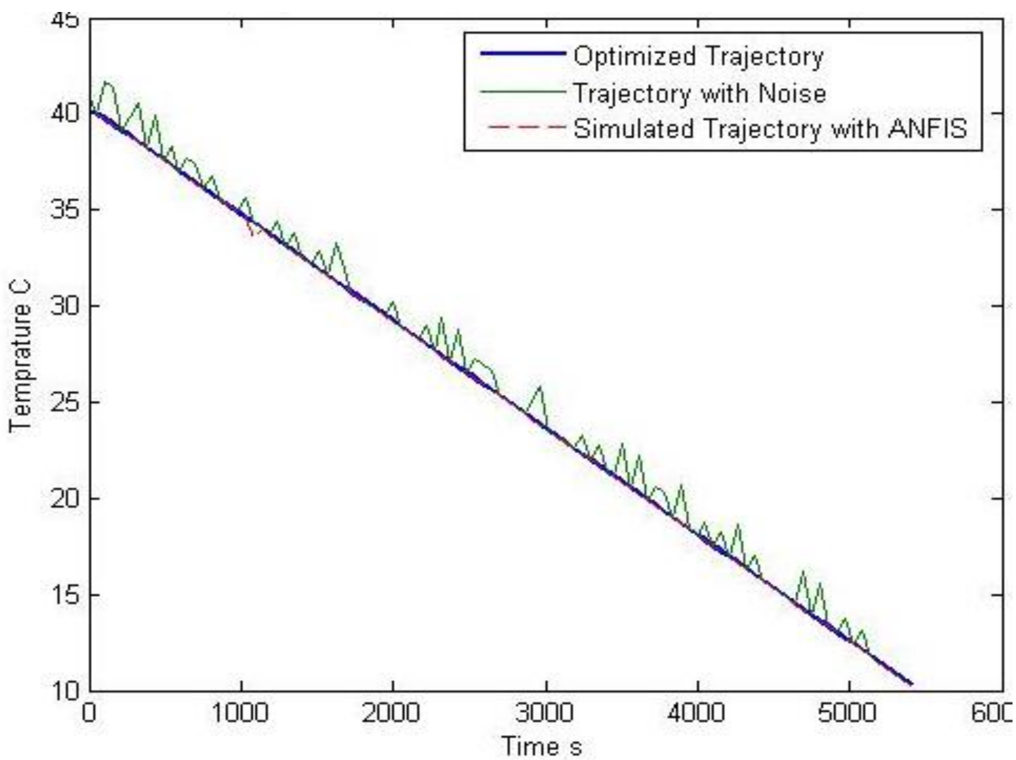


Figure 6.11: Temperature profile for linear model distribution with noise belongs to Poisson distribution.

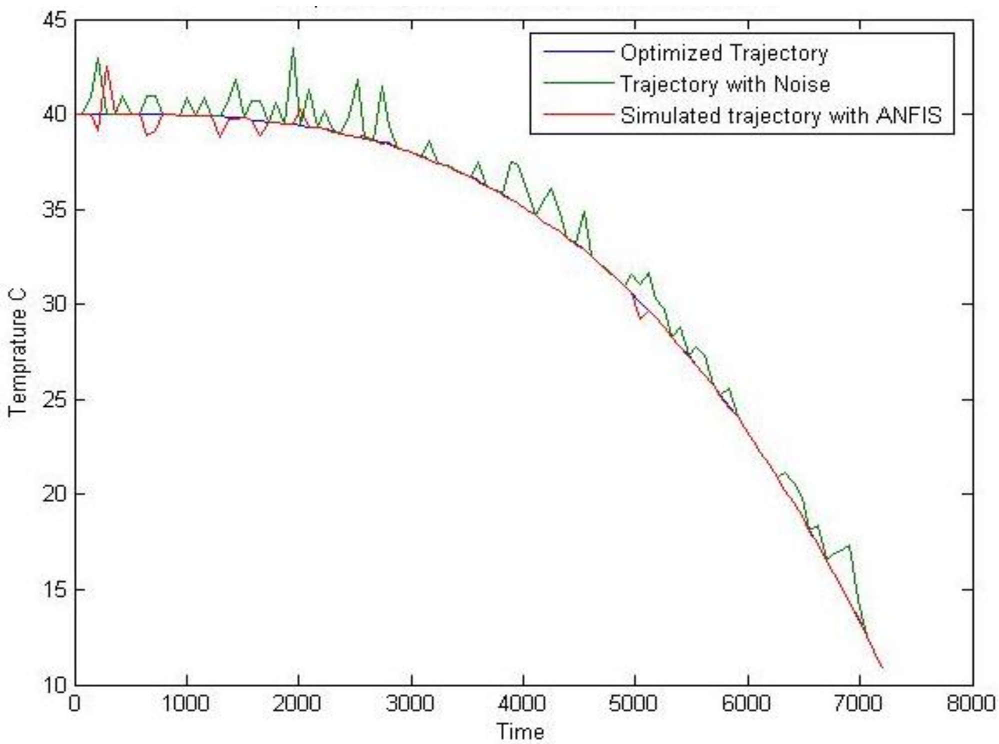


Figure 6.12: Temperature profile for Bio model distribution with noise belongs to Poission distribution.

6.4 Summary of the chapter

This chapter describes the testing of the controller and presented the simulated results obtained during the testing. In order to test the performance of controller, reference trajectories for mono model and bimodal CSDs are designed by using systematic direct design approach discussed in chapter four. Different types of Error / noise are added in these trajectories by generating random number in error range from normal distribution, uniform distribution and Poission distribution. These trajectories are presented to controller as output of the plant, control generate the control action by taking these noisy data and reference data. Simulation results are very encouraging and prove the efficacy of fuzzy neural networks in the designing of dynamic control systems.

Chapter 7

Summary of Research, Conclusion & Recommendations

7.1 Summary of Research

Crystallization is highly non linear process, large numbers of studies are being carried out to understand and interpret the process. Due to its nonlinear behavior this process is difficult to control. Initially the main objective of the process was to obtain the maximum throughput but with the improvement in understanding the crystallization process the focus is shifted to obtain maximum yield and crystals with desired properties. Main state variable of the crystallization are concentration of the solute, Process temperature, Jacket temperature, and the batch time. In industry crystallization is carried out by manipulating the supersturation trajectory due to its relation with crystallization process i.e. nucleation, growth etc. Super saturation trajectories are normally implemented with the help of time & temperature trajectories. These time and temperature trajectories in industry are implemented either by direct design approach or by Model based approach.

For crystallization process in industry all the process parameters cannot be measured accurately, especially concentration of solute for which sensors are at prototype stage and are very costly. Therefore these process parameters are estimated either by using model of the process or by direct design approach. In Direct design approach time & temperature trajectories are obtained experimentally or by hit & trial and do not provide systematic design approach. In the model base approach unmeasured parameters are estimated based on the available experimental data by using model. Basic requirements for the model based approach are to calculate the optimized estimated value for un measured process parameter before the next control step.

As the crystallization process is very complex and highly non linear. So the normal technique direct design and predicative control are either does not produce products with desired properties, or require lot of computational resources. So there is need to develop a control technique which use less computational resources during operation, independent of plant dynamics and provides a better control on the process.

Fuzzy logic has a capability to deal with vagueness and it shows very good result for the control of non linear system. Addition of neural net work induces the learning properties in fuzzy system. So the use of neural fuzzy system in the control system improves the performance of the system.

In this study the neuro fuzzy based controller is designed in which neuro fuzzy system is used to estimate total error due to all type of uncertainties and generate the control action to compensate the error. Neuro fuzzy system has two inputs and one output i.e. error, change in error are inputs and output is the control action to compensate the error and change in error. Error is the difference between the output of the plant and the required value and change in error is the rate of change of error. Both the inputs and output has five fuzzy variables. Whole ranges of values are divided into these five fuzzy variables. Fuzzy rules are generated by using T-Sugeno model. This neuro fuzzy system is trained for a random data within maximum value of error.

In order to test the performance of controller reference trajectories for mono model and bimodal CSDs by using systematic direct design approach. Error / noise is added in these trajectories by generating random number in error range from normal distribution, uniform distribution and poission distribution. These trajectories are presented to controller as output of the plant, control generate the control action by taking using these noisy data and reference data. Simulation results are very encouraging and prove the efficacy of fuzzy neural networks in the designing of dynamic control systems.

7.2 Conclusion

This thesis presents a methodology to control the temperature trajectories of crystallization process. In this approach ANFIS is used, which improves the performance of feedback control system. ANFIS is used to estimate the error due to all types of sources and generate the control action to compensate these errors. Control of the process can be implemented by using feedback system and ANFIS.

Design and working of the Controller is discussed in detail in chapter five. This chapter also describes the design of the ANFIS system. ANFIS system has two inputs and one output. Member ship functions for input and output and training of ANFIS system is also discussed.

Testing of the controller and discussion on the result is discussed in chapter six. Design of controller is tested by following steps:

- ANFIS is trained for dummy set of input and output data (which lies in the expected error range).
- The performance of the Controller is tested by performing simulations against the noisy data.

The data of Potash Alum in water for crystallization is used for verification / testing of controller. Temperature profiles for different distributions for crystallization is generated by direct design approach i.e. solving population balance equation (PBE) by combining the Quadrature Method of Moments (QMOM) and method of characteristics (MOCH) for desired size of crystal size distribution (CSD). Direct design approach will provide temperature profiles in the time domain which leads to constant supersturation corresponding to a desired CSD. These trajectories serve as reference trajectories for crystallization process. In order to test the performance of controller three types of dummy data is presented to controller, which can be obtained as

- Adding error in reference trajectories by generating random numbers in the range (-1 ~ 1) and (2~ 2).
- Adding error in reference linear and cubic temperature profiles by generating random numbers from Normal and Uniform Distribution.
- Adding error in reference linear and cubic temperature profiles by generating random numbers from Poisson Distribution

This noisy data is presented as input to the controller. In the controller ANFIS generate the control action to compensate the error and final control action is obtained by calculated and adding this action in the output. The output is compared with the actual data it shows the exact match of required trajectories.

The validation results are very encouraging and prove the efficacy of fuzzy neural networks in the designing of dynamic control systems. This study shows that Intelligent Controller based on ANFIS can be used to compensate the errors due to plant-model mismatch, unmeasured process disturbances in a crystallization process and can be implemented by using simple feedback tracking control system.

ANFIS has adaptive feature due to learning capabilities (i.e. it can be trained for new universe of discourse) its efficiency will improve every time it undergoes training.

This control system does not require any mathematical model so this can be implemented in any crystallizer having different features. Thus use of ANFIS makes this control system independent of plant without any extra hardware requirements. Training of ANFIS takes relatively more time and is carried out offline; once it is trained it will work in real time and will give the advantages of online dynamic optimization technique with less computation resources.

7.2 Recommendations

The described methodologies have been evaluated by simulations. However this can be implemented experimentally. And can be used to monitor and control the crystal size distribution in a nonlinear model predictive control framework. This could be applied to other systems as well by learning their control system design approach.

This study is carried out by using Sugeno fuzzy model, however, it is suggested this can also be evaluated by using Mamdani model, in which membership functions are assigned by using ANNS.

References

[1] Landlust, J., Mesbah, A., Wildenberg, J., Kalbasenka, A.N., Kramer, H. J.M. and Ludlage, J.H.A., 2008, An industrial model predictive control architecture for batch crystallization, In Proceedings of the 17th International Symposium on Industrial Crystallization Maastricht, The Netherlands, pp. 35–42.

[2] Landlust, J., Mesbah, A., Wildenberg, J., Kalbasenka, A.N., Kramer, H. J.M. and Ludlage, J.H.A, A model-based control framework for industrial batch crystallization processes , in chemical engineering research and design vol 8 8 (2 0 1 0) , pp 1223–1233

[3] E. Aamir, Z. K. Nagy, C.D. Rielly, T. Kleinert, and B. Judat, Combined quadrature method of moments and method of characteristics approach for the efficient solution of population balance model for dynamic modeling and crystal size distribution control of crystallization process". Ind. Eng. Chem. Res. 48, 2009, 8575.

[4] E. Aamir, Z. K. Nagy, C.D. Rielly, Systematic design of super-saturation controlled crystallization processes, in Proceeding of AIChE 2008: Annual Meeting of American Institute of Chemical Engineering, Philadelphia, PA, USA, ISBN 978 978-0-8169-1050-2 , November 20-22, 1-9, 2008.

[5] Z. K. Nagy and E. Aamir Systematic design of super-saturation controlled crystallization Processes for Shaping the Crystal Size Distribution Using an Analytical Estimator".ChemEngSci, 84, 2012, 656-670.

[6] R. Sivakumar, C. Sahana, P. A. Savitha , Design of ANFIS based Estimation and Control for MIMO Systems by International Journal of Engineering Research and Applications Vol. 2, Issue 3, May-Jun 2012, pp.2803-2809.

[7] JafarTavoosi, Majid Alaei, BehrouzJahani ,Temperature Control of Water Bath by using Neuro- Fuzzy Controller in proceedings of 5th Symposium on Advances in Science and Technology Mashhad Iran May 2011.

[8] Simon Hykon, Neural Networks A Comprehensive Foundation, Second Edition , Pearson Prentiace Hall 2005.

[9] Timothy J Ross, Fuzzy Logic with Engineering Applications, Second Edition, John Wiley & Sons 2004.

[10] Hamed. Vezvaei,Sepideh.Ordibeheshti,Mehdi.Ardjmand, Soft-Sensor for Estimation of Gasoline OctaneNumber in Platforming Processes with Adaptive Neuro-Fuzzy Inference Systems (ANFIS), World Academy of Science, Engineering and Technology 59 2011,pp 261~265

[11]Cheng-J L, Chi-Y Lee and Cheng-C Chin (2006), Temperature control using neuro-fuzzy controllers with compensatory operations and wavelet neural networks, Journal of Intelligent and Fuzzy Systems 17. IOS Press. Pages: 145 – 157.

[12]Jassar.S, Lian.Zhao, Zaiyi Liao, Ng, K.L.R (2009), Parameter Selection for Training Process of Neuro-fuzzy Systems for Average Air Temperature Estimation, International Conference on Mechatronics and Automation. China. IEEE. Pages: 1218 – 1222.

[13]Peymanfar.A, Khoei.A, Hadidi.Kh (2010), Design of a general propose neuro-fuzzy controller by using modified adaptive-network-based fuzzy inference system, AEU – International Journal of Electronics and Communications, Volume 64, Pages 433-442.

[14]Fuzzy Control by Kevin M. Passino& Stephen Yurkovich , printed by Addison Wesley company 1997.

[15]Control System By ManjitaSrivastava ,SmritiBhatnagar and others , printed by Tata McGraw-Hill Company New Delhi printed in 2009.

[16]Fuzzy Modeling and Control by AndrzejPiegat,printed by Physica – VerlagA Springer company in 2001.

[17]Artificial Neural Networks Concept and Control Applications edited bvy V. Rao Vermuri, printed by IEEE Computer Society Press Los Vogueros in 1992.

[18]Neuro Fuzzy Control Systems, Proceedings of the National Workshop on Neuro-Fuzzy Control Systems held in 1997printed by Narosa Publishing House New Delhi

[19]Budman, H., Kavchak, M., (1999), "Adaptive neural network structures fornonlinear process estimation and control", Computers and chem. Engng,Vol. 23, pp. 1209-1228.

[20] C++ Neural Networks and Fuzzy Logic by Valluru B. Rao published by M&T Books, IDG Books Worldwide, Inc.ISBN: 1558515526.

[21] Z.K. Nagy, M. Fujiwara , R.D. Braatz , Modelling and control of combined cooling and antisolvent crystallization processes ,Journal of Process Control in 2008.

[22] Wikipedia, the free encyclopedia , website on internet.

Appendix-A

Summary of some commonly used empirical relationships for growth and dissolution.

Physical Phenomena	Empirical Relation
Primary Nucleation	$B = k_b S^b$
Secondary Nucleation	$B = k_b S^b \mu_3$
Absolute Supersaturation	$S = \frac{C - C_{sat}}{C_{sat}}$
Kinetic Parameters	$[k_g, g, k_b, b]$
Material Balance	$\frac{dC}{dt} = -r_0 k_v (3Gm_2 + BL_0^3)$
Growth	$G = k_g S^g$
Relative Supersaturation	$S = C - C_{sat}$

Where

C	**Concentration of Solute**	**B**	**Nucleation**
$\boldsymbol{\rho_c}$	Density of crystal	L_0	Newly Inducted Crystal
$\boldsymbol{K_v}$	Volume Shape Factor	μ_3	Third Moment
$\boldsymbol{G}$	Growth	C_{sat}	Equilibrium Saturation
$\boldsymbol{k_g, g}$	Growth Parameters	k_b, b	Nucleation Parameters

APPENDIX B

SOURCE CODES

```
%% File to Create ANFIS
[System]
Name='show'
Type='sugeno'
Version=2.0
NumInputs=2
NumOutputs=1
NumRules=25
AndMethod='prod'
OrMethod='probor'
ImpMethod='prod'
AggMethod='sum'
DefuzzMethod='wtaver'

[Input1]
Name='Error'
Range=[-3 3.3]
NumMFs=5
MF1='NB': 'trimf',[-4.575 -3.00099441030916 -1.42870079670306]
MF2='NS': 'trimf',[-3.00026165742545 -1.42943625589944 0.132795032715903]
MF3='ZE': 'trimf',[-1.41704732135703 0.146326299629818 1.72350810935573]
MF4='PS': 'trimf',[0.160805455127888 1.72546616860901 3.29988351295232]
MF5='PB': 'trimf',[1.73032176054372 3.30111528545192 4.875]

[Input2]
Name='ErrorChange'
Range=[-2 2]
NumMFs=5
MF1='NB': 'trimf',[-3 -2.00000044802443 -1.00005444700847]
MF2='NS': 'trimf',[-1.99999434215936 -1.00336510755098 0.00203989557253881]
MF3='ZE': 'trimf',[-0.999704603552994 0.00456044852034993 0.999290961051927]
MF4='PS': 'trimf',[0.00491358336992784 1.00773901975173 1.99999809201911]
MF5='PB': 'trimf',[1.00000760184099 2 3]

[Output1]
Name='output'
Range=[-3.1 3.3]
NumMFs=25
MF1='out1mf1':'constant',[0]
MF2='out1mf2':'constant',[4.05774584041382]
MF3='out1mf3':'constant',[2.98250975235983]
MF4='out1mf4':'constant',[2.00025509553064]
MF5='out1mf5':'constant',[0]
```

MF6='out1mf6':'constant',[0.203362989414018]
MF7='out1mf7':'constant',[2.35807064923576]
MF8='out1mf8':'constant',[1.43646681192726]
MF9='out1mf9':'constant',[0.457529167064248]
MF10='out1mf10':'constant',[-1.20482316377456]
MF11='out1mf11':'constant',[2.16848784634203]
MF12='out1mf12':'constant',[0.781497260495468]
MF13='out1mf13':'constant',[-0.152516848825596]
MF14='out1mf14':'constant',[-1.24510692372757]
MF15='out1mf15':'constant',[-0.443240933323978]
MF16='out1mf16':'constant',[0]
MF17='out1mf17':'constant',[-0.668126029050253]
MF18='out1mf18':'constant',[-1.77159587457718]
MF19='out1mf19':'constant',[-2.37664000383969]
MF20='out1mf20':'constant',[0]
MF21='out1mf21':'constant',[0]
MF22='out1mf22':'constant',[-3.06187656687053]
MF23='out1mf23':'constant',[-3.10960155918724]
MF24='out1mf24':'constant',[0]
MF25='out1mf25':'constant',[0]

[Rules]
1 1, 1 (1) : 1
1 2, 2 (1) : 1
1 3, 3 (1) : 1
1 4, 4 (1) : 1
1 5, 5 (1) : 1
2 1, 6 (1) : 1
2 2, 7 (1) : 1
2 3, 8 (1) : 1
2 4, 9 (1) : 1
2 5, 10 (1) : 1
3 1, 11 (1) : 1
3 2, 12 (1) : 1
3 3, 13 (1) : 1
3 4, 14 (1) : 1
3 5, 15 (1) : 1
4 1, 16 (1) : 1
4 2, 17 (1) : 1
4 3, 18 (1) : 1
4 4, 19 (1) : 1
4 5, 20 (1) : 1
5 1, 21 (1) : 1
5 2, 22 (1) : 1
5 3, 23 (1) : 1
5 4, 24 (1) : 1
5 5, 25 (1) : 1

%% Code to Generate Simulation For Linear Modal

```
Opt1=1.0e+003 *[0    0.0400
    0.0540    0.0400
    0.1080    0.0397
    0.1620    0.0394
    0.2160    0.0391
    0.2700    0.0388
    0.3240    0.0385
    0.3780    0.0382
    0.4320    0.0379
    0.4860    0.0376
    0.5400    0.0373
    0.5940    0.0370
    0.6480    0.0367
    0.7020    0.0364
    0.7560    0.0361
    0.8100    0.0358
    0.8640    0.0355
    0.9180    0.0352
    0.9720    0.0349
    1.0260    0.0346
    1.0800    0.0343
    1.1340    0.0340
    1.1880    0.0337
    1.2420    0.0334
    1.2960    0.0331
    1.3500    0.0328
    1.4040    0.0325
    1.4580    0.0322
    1.5120    0.0319
    1.5660    0.0316
    1.6200    0.0313
    1.6740    0.0310
    1.7280    0.0307
    1.7820    0.0304
    1.8360    0.0301
    1.8900    0.0298
    1.9440    0.0295
    1.9980    0.0292
    2.0520    0.0289
    2.1060    0.0286
    2.1600    0.0283
    2.2140    0.0280
    2.2680    0.0277
    2.3220    0.0274
    2.3760    0.0271
```

2.4300	0.0268
2.4840	0.0265
2.5380	0.0262
2.5920	0.0259
2.6460	0.0256
2.7000	0.0253
2.7540	0.0250
2.8080	0.0247
2.8620	0.0244
2.9160	0.0241
2.9700	0.0238
3.0240	0.0235
3.0780	0.0232
3.1320	0.0229
3.1860	0.0226
3.2400	0.0223
3.2940	0.0220
3.3480	0.0217
3.4020	0.0214
3.4560	0.0211
3.5100	0.0208
3.5640	0.0205
3.6180	0.0202
3.6720	0.0199
3.7260	0.0196
3.7800	0.0193
3.8340	0.0190
3.8880	0.0187
3.9420	0.0184
3.9960	0.0181
4.0500	0.0178
4.1040	0.0175
4.1580	0.0172
4.2120	0.0169
4.2660	0.0166
4.3200	0.0163
4.3740	0.0160
4.4280	0.0157
4.4820	0.0154
4.5360	0.0151
4.5900	0.0148
4.6440	0.0145
4.6980	0.0142
4.7520	0.0139
4.8060	0.0136
4.8600	0.0133
4.9140	0.0130

```
    4.9680    0.0127
    5.0220    0.0124
    5.0760    0.0121
    5.1300    0.0118
    5.1840    0.0115
    5.2380    0.0112
    5.2920    0.0109
    5.3460    0.0106
    5.4000    0.0103
];

x=Opt1(:,1);
y=Opt1(:,2);
dt=diff(y);
dt1=[dt;0];
a=-2;
b=2;
r = a + (b-a).*rand(101,1);
ye=y+r;
a=-.01;
b=.01;
ce = a + (b-a).*rand(101,1);

cd=[r,ce];

fismat = readfis('show');

out = evalfis(cd,fismat);

y1=ye+dt1+out;
y2=[y(1);y1];
yc=y2(1:101,1);
plot(x,y,'b',x,ye,'r',x,yc,'g')
```

%% Code to Generate Simulation ForCubic Modal with noise from Uniform %%Distribution

```
Opt1=1.0e+003 *[0    0.0400
    0.0720    0.0400
    0.1440    0.0400
    0.2160    0.0400
    0.2880    0.0400
    0.3600    0.0400
    0.4320    0.0400
    0.5040    0.0400
    0.5760    0.0400
    0.6480    0.0400
```

0.7200	0.0400
0.7920	0.0400
0.8640	0.0400
0.9360	0.0399
1.0080	0.0399
1.0800	0.0399
1.1520	0.0399
1.2240	0.0399
1.2960	0.0399
1.3680	0.0398
1.4400	0.0398
1.5120	0.0398
1.5840	0.0397
1.6560	0.0397
1.7280	0.0396
1.8000	0.0396
1.8720	0.0395
1.9440	0.0395
2.0160	0.0394
2.0880	0.0393
2.1600	0.0393
2.2320	0.0392
2.3040	0.0391
2.3760	0.0390
2.4480	0.0389
2.5200	0.0388
2.5920	0.0387
2.6640	0.0386
2.7360	0.0385
2.8080	0.0384
2.8800	0.0382
2.9520	0.0381
3.0240	0.0379
3.0960	0.0378
3.1680	0.0376
3.2400	0.0374
3.3120	0.0373
3.3840	0.0371
3.4560	0.0369
3.5280	0.0367
3.6000	0.0365
3.6720	0.0362
3.7440	0.0360
3.8160	0.0358
3.8880	0.0355
3.9600	0.0353
4.0320	0.0350

```
    4.1040    0.0347
    4.1760    0.0344
    4.2480    0.0341
    4.3200    0.0338
    4.3920    0.0335
    4.4640    0.0332
    4.5360    0.0329
    4.6080    0.0325
    4.6800    0.0321
    4.7520    0.0318
    4.8240    0.0314
    4.8960    0.0310
    4.9680    0.0306
    5.0400    0.0301
    5.1120    0.0297
    5.1840    0.0293
    5.2560    0.0288
    5.3280    0.0283
    5.4000    0.0278
    5.4720    0.0273
    5.5440    0.0268
    5.6160    0.0263
    5.6880    0.0258
    5.7600    0.0252
    5.8320    0.0246
    5.9040    0.0241
    5.9760    0.0235
    6.0480    0.0228
    6.1200    0.0222
    6.1920    0.0216
    6.2640    0.0209
    6.3360    0.0202
    6.4080    0.0196
    6.4800    0.0189
    6.5520    0.0181
    6.6240    0.0174
    6.6960    0.0166
    6.7680    0.0159
    6.8400    0.0151
    6.9120    0.0143
    6.9840    0.0135
    7.0560    0.0126
    7.1280    0.0118
    7.2000    0.0109
];

x=Opt1(:,1);
```

```
y=Opt1(:,2);
dt=diff(y);
dt1=[dt;0];
fori=1:101 ;
A(i) = unifrnd(-2,2);
end
r = A'
ye=y+r;
fori=1:101 ;
B(i) = unifrnd(-.2,.2);
end
ce = B'
cd=[r,ce];

fismat = readfis('show');

out = evalfis(cd,fismat,101);

y1=ye+dt1+out;
y2=[y(1);y1];
yc=y2(1:101,1);
plot(x,y,x,ye,x,yc)
```