Application of Fuzzy Logic to Detection of Internal Leakage Fault in Hydraulic Systems

A THESIS

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL

OF THE UNIVERSITY OF MINNESOTA

BY

Manali Ashwinikumar Kulkarni

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF SCIENCE

Dr. Seraphin Chally Abou, Dr. Marian Stachowicz

April 2011

Acknowledgement

No words can express my thanks to my advisor, Dr. Seraphin Abou for his valuable advice, continuous guidance and encouragement.

I would like to express my sincere gratitude to my co-advisor, Dr. Marian Stachowicz for his guidance in conducting this research. I would also like to thank Prof. Dr. Richard Lindeke for taking time and efforts to review this thesis.

I am grateful to Dr. Nariman Sepehri for his contributions on this research project and I would like to specially thank him. The research on internal leakage fault diagnosis was done at University of Manitoba, Winnipeg, Canada under the supervision of Dr. Nariman Sepehri.

I would like to thank University of Minnesota Duluth, Mechanical and Industrial Engineering department and Graduate school for providing me research and teaching assistantships. I would also like to thank University of Manitoba, Winnipeg, Canada for inviting me to take part in their research and for providing test rig facility.

Even physically separated from my family I had their love and support to complete my education here. I am in deep gratitude to my mother Arati and father Ashwinikumar who never lost faith in my capabilities and for their unlimited support. I would also like to thank my sister Sayali for her love and support.

Most of all I would like to express my deepest appreciation to my husband, Vinod, for his love, friendship and many sacrifices all of which made this effort possible. I also want to thank my parents-in-law Rajani and Dilip. Last but not the least I would like to thank my Landlord Mary Ann, who became a part of my life in Duluth for all her help.

Last but not the least I would like to thank all my friends who are seeking precise answers for fuzzy problems.

Dedication

This thesis is dedicated to ‘my family’

Abstract

Detection of fault as early as possible before it leads to any financial loss or even catastrophic failure is very important. Like other systems, faults are inevitable part of hydraulic system. Due to its high power transmission capacity, the usage of hydraulic systems is high in today's industry and so is the need for its fault diagnosis.

Fault in a hydraulic system can arise due to numerous reasons like change in environmental conditions, change in bulk modulus or viscosity of the hydraulic fluid or drop in supply pressure. A faulty sensor or malfunction of any actual component in the whole system can also lead to failure. Using fuzzy logic a yes/no decision of fault is upgraded to a percent fault severity at the output in first part of this study. When a system deviates from its normal residuals are generated which are the measure of amount of fault present in the system. These residuals are evaluated using fuzzy logic. The performance of system is successfully evaluated and final output is fault severity which ranges from 0-100%. This approach combines fuzzy logic approach which is a knowledge based technique with an already developed model based technique.

Out of these numerous faults that can potentially arise, the second part of this study focuses on detection of internal leakage fault in hydraulic actuators. An algorithm which successfully detects occurrence of internal leakage in hydraulic system is developed using fuzzy logic. This research uses the available knowledge according to which "changes in pattern are observed in the second level wavelet transform of the pressure signal measured at one end of the chamber". Combining this knowledge with measured data, membership functions and heuristic rules are developed which mimic human mind (logical reasoning) in order to make conclusions regarding occurrence of internal leakage fault. Two different algorithms are developed and both are repeated taking into consideration data in two different time intervals. These four approaches are compared based on results obtained on different sets of data at the end of this dissertation.

The method proposed here is a knowledge based method which primarily uses fuzzy logic and acts as an extension to already developed model based techniques. Thus the final diagnosis made is a combination of model based and knowledge based technique.

Table of Contents

List of Tables

List of Figures

1. INTRODUCTION

1.1 Outline

A fault is defined as any disruption to expected or normal behavior of a system. Thus, a fault is an error which leads to malfunction of the system and in some cases to catastrophic failure. If no fault is present, the system is said to be in a healthy or normal condition. In case of a fault the system starts deviating from its normal or healthy state. Faults can be categorized into two types: ‘partial fault failure’ where the system can continue to operate in presence of a small scale fault and ‘total fault failure’ where the system needs to shut down or ceases to function for safety purpose.

Fault detection is an integral part of most of the systems especially in applications like aircraft control where faults can lead to a life threatening situation. It is also crucial in high precision processes like semiconductor industry, where system faults can cause significant financial loss if they went undetected. In these cases it is necessary that the system works reliably all the time. Fault detection is an extremely important and at the same time challenging task. The amount of fault depends on where the system is used. For example, if a hydraulic system being used in an aircraft and it is undergoing a small amount of internal leakage, the fault cannot be fixed till the aircraft lands which might be several hours later. On the other hand, if the hydraulic system is used in a semiconductor manufacturing facility, huge financial loss can be avoided in the upcoming semiconductor wafers by detecting the fault within a few minutes. The design of the fault detector depends on the amount of fault the system can tolerate. For example the upper and the lower thresholds are relatively close when the system has low amount of fault tolerance while it increases as the system’s tolerance to fault increases.

Like any other system, faults are an inevitable part of a hydraulic system. A hydraulic system uses hydraulic fluid under pressure to generate force or motion as its primary driving media. Hydraulic systems are widely used in industry due to their capacity to transmit high power. The actuator makes use of the power transferred by the hydraulic fluid through fixed lines or flexible tubes to generate motion. Due to its capacity to deliver high power, hydraulic systems are used in heavy-duty equipment including mechanisms to control aircraft flight surfaces or in excavators to move heavy objects. In all of these situations, the consistent and reliable operation of the hydraulic system is necessary and expected.

A hydraulic system is prone to different types of faults. Changes in environmental conditions including temperature or humidity can cause a change in the system parameters like bulk modulus or viscosity of the hydraulic fluid causing a considerable deviation from the expected system output. A drop in the supply pressure can also lead to an unexpected output. A faulty sensor generating incorrect data to the operators often results in misleading control information. Fault can also arise due to actual malfunction of any component within the system.

The leakage of hydraulic fluid is one of the major causes of fault. There are two types of leakage present in the system depending on their location. When the hydraulic fluid leaks from one chamber of the actuator cylinder to another it is called internal leakage and when it leaks out of the cylinder, it is called external leakage. Internal or external leakage or both can cause a substantial drop in hydraulic pressure and eventually decrease the velocity or controllability of the output shaft. Comparing detection of internal and external leakage one finds that external leakage is visible to the operator and hence is easily detectible. This study concentrates on the detection of internal leakage in the hydraulic system. The main cause of internal leakage is the wear and tear of piston seals which allows the hydraulic fluid to escape from one chamber to another without contributing to power output.

One or more of these faults mentioned above cause the system to deviate from its expected output. Reference [1] uses a mathematical modeling approach to detect fault in the system. The output of the system is predicted using non-linear observers while the actual output of the system is known through the sensors. The difference between the predicted output and the actual output yields a 'residual'. These residuals are a measure of the amount of fault present in the system. They are added over a period of time to compute 'cumulative residuals'. The final decision regarding the occurrence of fault is made by evaluating these 'cumulative residuals' using the threshold concept. In [1], anything outside the upper and lower threshold is considered as a system in fault while anything in-between is not. This means that a cumulative residual just above the threshold will be considered a fault while the one which is just below is not considered as a fault. This "hard cutoff" might, potentially, lead to missing alarms or give false triggers to the operator.

In this study this situation is upgraded to decide on the severity of fault at the output instead of a simple yes/no decision. This was done using fuzzy logic where the binary yes/no decision regarding fault was converted to a smooth transition from no to yes in terms of severity of fault. The performance of the system is successfully evaluated and the final output is the fault severity which ranges from 0-100%. Fuzzy logic is more effective when it is combined with a model based technique, like the non-linear observers in this case, to make the final decision.

Out of the different types of faults that can potentially arise, this study concentrates on the internal leakage fault in the hydraulic system. A difference is observed in the second level wavelet transform pattern of the pressure signal measured at one end of the (potentially leaky) actuator [2]. The amplitude of spikes and their frequency observed in this second level wavelet transform before introducing leakage is higher than that after leakage is introduced.

The fuzzy logic system developed in this study makes use of this information and the other available data to develop rules and make inferences regarding the occurrence of fault. In this online fault detection system, fuzzy logic is applied at three different levels to the input data. Rule based inferences are made at each level by developing membership functions and rules which map the input data to the desired output. The final control output is the percent fault severity which is converted to a digital yes/no decision for the presence/absence of internal leakage fault. The 'number based approach' and the 'ratio based approach' are the two different approaches for fault detection discussed in the later chapters of the thesis. The fuzzy logic system is developed for these two approaches first considering data present in a 20 second data interval and then considering the data present in the 60 second data interval. The algorithm is successfully tested and the results are compared for different sets of data.

The methodology is mainly based on the data and the linguistic information available (with very little mathematical modeling) [7]. In situations where the amount of uncertainty is high and it is difficult and time consuming to perform mathematical modeling, a diagnostic system based in fuzzy logic provides a flexible, cost effective tool for fault detection which is reliable to some extent depending on its design and application. In this thesis, the percentage of success for the available data ranges from 60% (Method I which uses 20 second data interval) to 90% (Method II which uses 60 second data interval) for the available data sets.(As shown in the last chapter under 'Results' section)

Continuous online monitoring of fault in hydraulic system is becoming increasingly important. It is necessary and equally important to provide correct information regarding the systems health to the operators as quickly as possible in order to guide them toward fixing of fault related problems.

1.2 Literature Review

1.2.1 Different types of approaches for fault detection

This section reviews the developments in hydraulic system for fault diagnosis. It briefly categorizes the different approaches that can be used to detect faults in hydraulic systems and emphasizes on the methodology proposed in this dissertation. These approaches can be mainly categorized as follows:

1) Model based approach
2) Model free or knowledge based approach
3) Combination of model based and knowledge based approach

1) Model Based Approach

For fault detection in a hydraulic system, generation of residuals using a non-linear observer and their evaluation using a statistical method is a widely employed model based technique. In this approach, the observer estimates the next state of the system, given the initial state. The actual state y(k) of the system is known through the sensors while the expected state z(k) is predicted by the non-linear observer. This difference between the actual state and the estimated state is known as residual e(k).

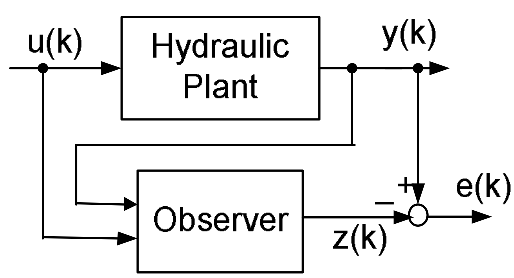


Figure 1.1 Nonlinear observer structure

When the system is in its normal operating condition, the residual is equal to (or nearly) zero. On the other hand when the residual is not zero it indicates that the system is diverging from the normal or expected state indicating a fault state.

$$|r(t)|\begin{cases}\approx 0 \Rightarrow NoFault \\ \gg 0 \Rightarrow Fault\end{cases}$$

Due to limitations in mathematical modeling, all the aspects of nonlinearity which are present in the system cannot be taken into account. Hence, the value of residual is never equal to zero even in case of no fault condition but there is always some small value of residual present either positive or negative. This gives rise to the concept of thresholds. Any residual between the upper and lower thresholds is 'not' considered as fault and anything outside the thresholds is considered as fault. The threshold is set depending on the systems tolerance limit which in turn depends on the area of application of the system.

A non-linear observer based method using the Lipchitz class of nonlinear equations can be seen in [1]. The non-linear observer estimates four different states of the system which are velocity, input pressure, output pressure and the spool displacement. Out of these four parameters, 'velocity' is used for performing residual evaluation. The residuals generated at each time interval are added over a period of time into so-called cumulative residuals. These cumulative residuals are evaluated by Wald's sequential test which uses the threshold concept to detect the occurrence of fault.

Similar fault detection techniques using the observers to generate residuals and evaluating them to make decisions for the occurrence of fault can be seen in [3], [4], [5], and [6].

2) Knowledge Based Approach

Many past controller applications involving diagnoses have been based on extracted information from data or an expert's knowledge or learning from experience in order to develop rules. This rule development process provides nonlinear mapping between the inputs and the outputs. The knowledge about the system's input and output or an extensive set of data are the basic requirements for knowledge based diagnosis. Different applications of expert systems include management decision making, mineral exploration, VLSI design, diagnosis of mechanical machinery, diagnostic medicine and many other areas.

Five stages are suggested in [7] to develop an expert system: 1) identification, 2) conceptualization, 3) formulization, 4) implementation, and 5) testing. The first stage determines all the input data available for the development process and the resources needed. In conceptualization, the available knowledge about the system and the system's data are organized with the help of an expert. The third stage, formulation, suggests the formal representation of relationships between the inputs and outputs of the system written in the form of rules. In the implementation stage, all the knowledge gathered until then is combined and the rules are implemented on the inputs. The results are then tested on different sets of input data during the final stage.

This approach is also known as artificial intelligence (AI) or expert system and mimics the decision making process of a human expert. The expert systems also consider the 'uncertainty' which is inherent in any kind of system. This human like behavior is advantageous in order to incorporate the operator's experience and knowledge in the methodology and guide the technicians toward fixing of the problems that cause the fault state.

Some of the limitations of the expert systems are that a great deal of knowledge and/or data needs to be available during model development. These systems involve significant trial and error which makes them very time consuming at times to develop [7]. Expert systems normally cannot reason when the subject domain is too broad. They are confined to using specific facts and heuristics acquired through the knowledge base. Expert system performance degrades rapidly when problems range beyond the specific task the system was designed to perform. [7] For these reasons, it is difficult to use expert system for particular problem solving alone. They are best used in combination with an already developed model which might be mathematical based.

As the design of an expert system depends on the system's available data and the expert's knowledge (developed over a period of time), it is almost impossible to develop a fault detection system before the actual system is developed and is in actual use for some period of time.

Neural networks (NN), genetic algorithms and fuzzy logic are expert systems that can be used separately or in combination. In [8], neural network models were used for residual estimation to detect fault in electro-hydraulic servo system (EHS). Residuals were directly generated from the difference between the measured signal and the signal calculated using NN. These residuals are then evaluated using the threshold approach to make the decision regarding fault state.

Another paper [9] proposes the application of the combination of neuro-fuzzy and genetic algorithms in mechatronic systems. It shows the benefits of knowledge based techniques like these, over mathematical modeling, in different fields including mechanical engineering, signal processing, actuation and control, etc. The results show a higher efficiency, good accuracy as well as a better processing of signals than competing techniques.

Feed forward NN's were investigated in [10] for fault diagnosis in the arena of chemical processing. Genetic Algorithms are used to obtain optimum and robust fault diagnosis rules in [11]. Several other applications of knowledge based methods for fault detection can be seen in [12], [13], [14], [15], [16], and [17].

3) Combination of Model Based and Knowledge Based Approach

Perhaps the best way to perform fault detection is the combined use of model based and knowledge based methods. This is because the controller can combine the advantages and eliminate the limitations of both the approaches resulting in a more efficient strategy. A combination of model based diagnostic strategies in conjunction with a set of logical rules representing human behavior provides an excellent diagnostic scheme as in [18].

For example using fuzzy logic rules with a previously developed model can be a good approach as it overcomes the complexity of the expert system development and, at the same time, provides a smooth transition between decisions. Tasks which require human intuition and experience can be well resolved using fuzzy logic system (FLS).

A rule based approach like FLS is usually the decision making component of such a system while the model based approach provides it with some processed results. FLS uses these processed results as inputs to make the final decision rather than using the original data from the system.

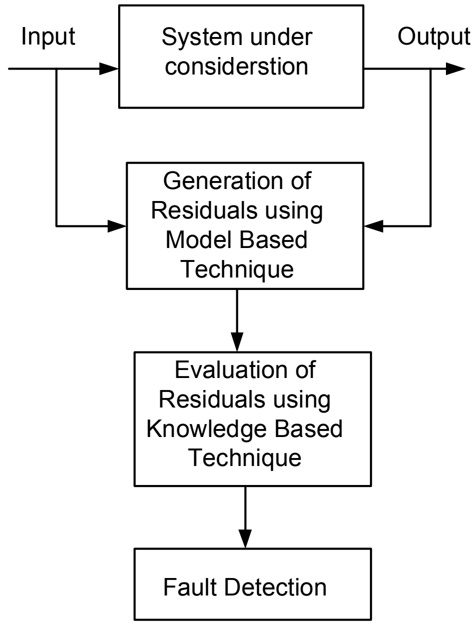


Figure 1.2 Combination of model based and knowledge based technique

1.2.2 Basics of Hydraulic System

1) Hydraulics principle

Hydraulic systems are governed by Pascal's Law. Pascal's law says that the pressure of a static hydraulic fluid in a closed system is everywhere the same. The pressure can be calculated using the formula

$$Pressure(psi) = \frac{Force(pounds)}{Area(inch^2)}$$

The basic idea behind any hydraulic system is very simple:

Force that is applied to one point is transmitted to another point using an incompressible fluid which is normally oil.

2) Overall Working

The hydraulic system under consideration here is a 'four way valve controlled linear actuator'. Its main components are: a four way proportional valve, a hydraulic actuator, the hydraulic fluid, a pump with pressure regulator, various sensors, and flexible hoses.

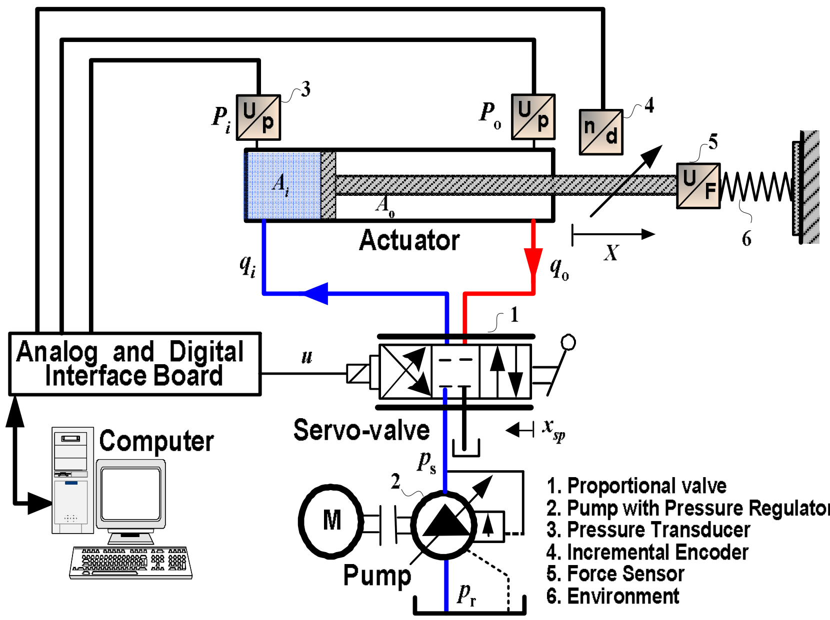


Figure 1.3 Hydraulic System under consideration

These parts work synchronously to produce linear motion at the output. The pump with the pressure regulator pumps the hydraulic fluid into the system at the required pressure. This fluid enters the solenoid valve at the supply port. The valve allows the flow of this fluid under pressure into the left hand side chamber of the cylinder. The pressure causes the shaft to move to the right pushing the fluid that was already present in the right hand side chamber out. This return fluid exits through the right side of cylinder and enters the solenoid valve. It then exits the valve and flows to the reservoir to be pumped into the system again. Events happen in the reverse order the shaft is to move in the opposite direction. In this way, all the components work together to accomplish the task of linear motion.

One can observe the flow due to internal leakage in Figure 1.4. A damaged seal allows the flow of fluid directly from one chamber of the actuator into the other eventually leading to a decrease in output velocity of the shaft. One can also observe the hydraulic fluid leaking out of the chamber mentioned as ‘external leakage’. It is the most recognizable type of leakage. Detection of leakage is the first step in controlling it.

Undetected leakage can cause a substantial drop in the performance efficiency of the machine. It also contributes to the environmental damage and financial loss to the company due to higher consumption of oil. External leakage such as a burst hose is usually obvious and therefore easy to find as opposed to internal leakage. An online fault detection system is proposed in this thesis which detects the internal leakage of the hydraulic system.

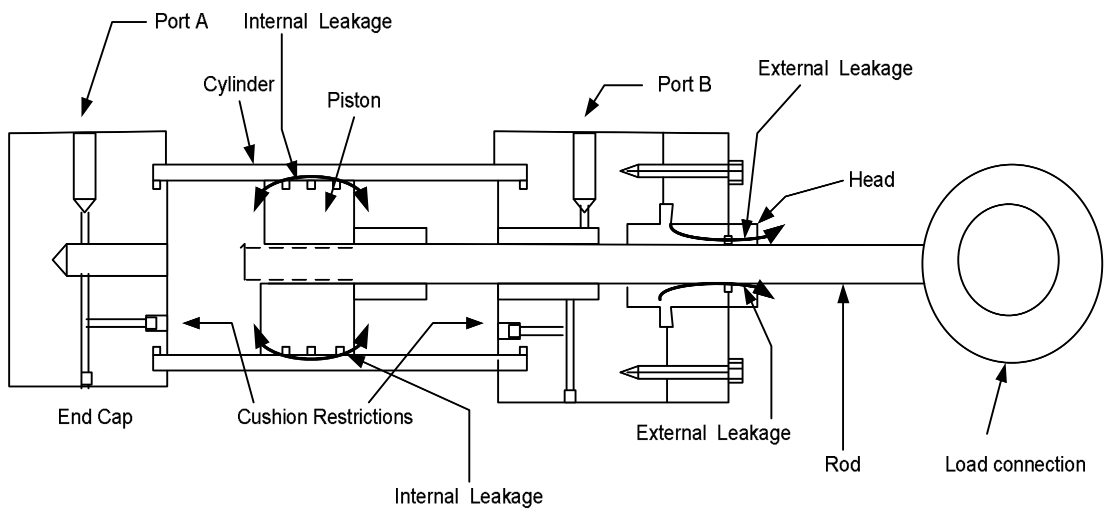


Figure 2.4 Internal leakage in actuator

2. FUZZY MODELING

In the "real world" engineers and control operators need to make decisions by analyzing the data or the information that is available to them. Information that is not available gives rise to uncertainty. As stated in [19]:

"Uncertainty is the condition in which the possibility of error exists, because we have less than total information about our environment"

The human mind tries to make decisions in the presence of uncertainty. Knowledge based systems try to mimic human mind and reasoning in this way. Uncertainty always gives rise to complexity. As the amount of uncertainty increases complexity increases too. For example, a child learning to ride a bike is 'uncertain' regarding how to ride a bike. He knows that one needs to sit on it and rotate the pedals but does not know how to balance (has less than total information/knowledge) and hence finds it to be a complex task. Comparatively an adult who is used to riding a bike has full knowledge and finds it to be a simple task.

With accuracy there is always a cost involved. It is up to the system user to decide the amount of precision required to accomplish a task. There is always a tradeoff between accuracy and cost. In some areas the precision required is very high, for example in a semiconductor company where chips are fabricated, while in others, like in case of an excavator, less precision should be acceptable.

In day-to-day situations the available information is most probably expressed in natural language. With the use of natural language comes a high level of 'vagueness'. For example when one says the water is warm, it might be comparatively cold during the winter season whereas it will be hotter during the summer season. Hence there needs to be a quantitative method that takes into account the vagueness in natural language when addressing complex problems.

Different soft computing methods including genetic algorithms, artificial neural networks and fuzzy logic each propose, in their own way, a solution for the real world problems mentioned above, that is system uncertainty, complexity, precision and vagueness.

As most of the real world concepts cannot be entirely described using 'Yes' or 'No', the concept of a matter of degree provides the required flexibility to deal with these real world situations. 'Fuzzy set theory' is a mathematical based framework which enables the study of imprecise and vague concepts. A fuzzy logic based soft computing approach is developed in this study to make a decision of whether the fault is present in the system looking at the available data and linguistic information.

2.1 Introduction to Fuzzy Set Theory

In his paper [20], professor Lofti Zadeh introduced the concept of fuzzy sets. Contradictory to crisp set theory, the boundaries of these sets are not precise. In case of crisp sets, an entity either belongs to a particular set or it does not. In other words it belongs to one and only one set. However, in case of fuzzy sets, a particular entity may belong to a particular set to some extent while, at the same time, it may also belong to another fuzzy set but with to a different degree of extent. This membership of an entity to a particular set is a matter of degree is called ‘grade of membership’ in terms of fuzzy set theory.

The example seen in the MATLAB fuzzy logic toolbox helps explain the concept of fuzzy sets very clearly. According to this example, when one describes a set of “Days of Weekend”, they can say, for sure, that Sunday and Saturday belong to the set. However, Friday is a day which cannot be explicitly said to belong to that set but it does often *kind of* feel like a day of the weekend. Therefore, in terms of fuzzy set theory, one can say that Friday does belong to the set of the “Days of Weekend” but, with a degree of membership equal to 0.7 rather than 1.0 as for Saturday or Sunday. Similarly, one can add that Thursday belongs to the set of “Days of Weekend” too, but with a degree of membership equal to 0.3.

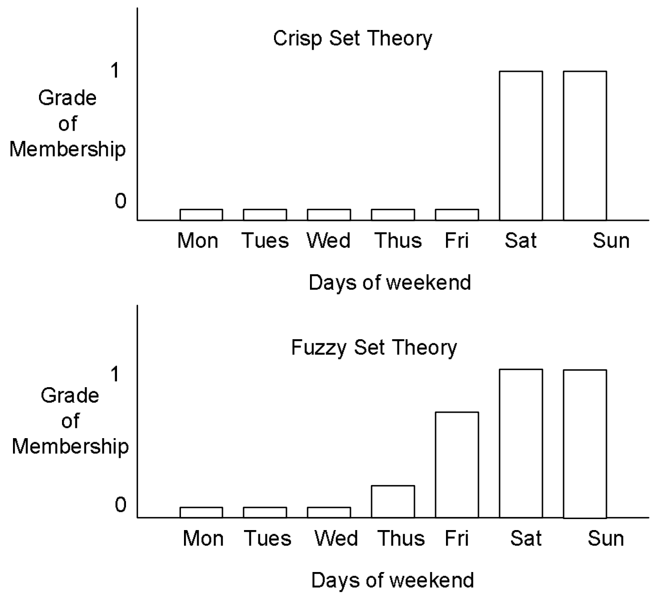


Figure 2.1 Fuzzy Representation

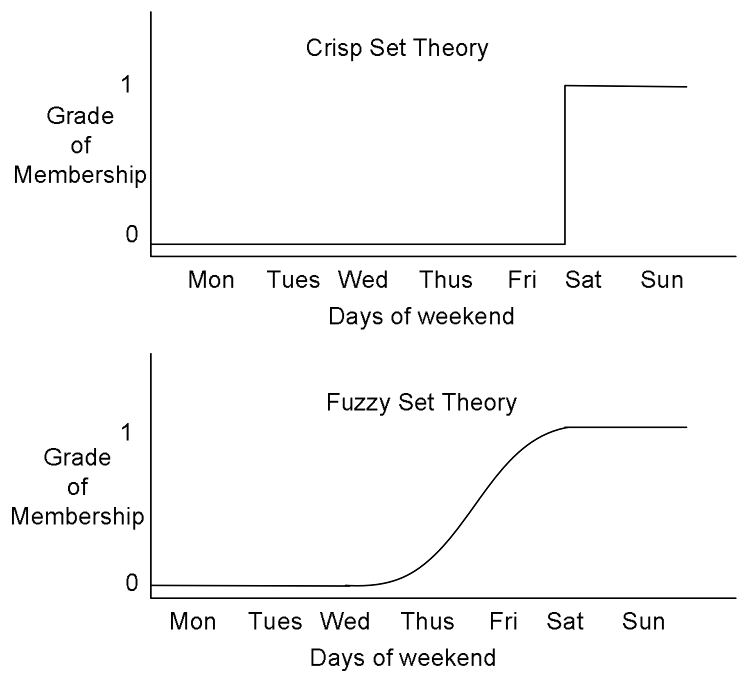


Figure 2.2 Smooth Transition from 'Belongs' to 'Does not belong'

So one can see a two valued logic in crisp set theory where the entity either belongs to the set or it does not. On the other hand, a smooth transition of items from ‘Does not belong’ to the set to ‘Belongs’ to the set can be seen by using fuzzy set theory.

2.2. Fuzzy Systems as Universal Approximators

Most of the systems found in engineering applications are nonlinear. Fuzzy Logic is an effective way of mapping nonlinear functions. Hence, a fuzzy logic system is known as a universal approximator. A fuzzy logic system can approximate a non-linear function to arbitrary accuracy. This property makes fuzzy logic applicable to a wide variety of applications.

A fuzzy system approximates a function by covering its graph with fuzzy patches. The average of patches is taken wherever they overlap. The accuracy of approximation increases as the fuzzy patches become smaller and increase in number [21]. Each patch represents a fuzzy rule in the input-output state space of the function. Figure 2.3 shows the fuzzy patches in the input-output state space covering the actual function *f*: X→Y.

It can be seen that the approximation improves as one adds smaller patches, which are greater in number. But, the cost and the computation time increases due to increase in complexity of the smaller segments of a higher count. Similarly, fewer patches decrease the complexity and computation time but they also decrease the accuracy of the approximation. There is always a tradeoff between actual cost, computation time and desired accuracy.

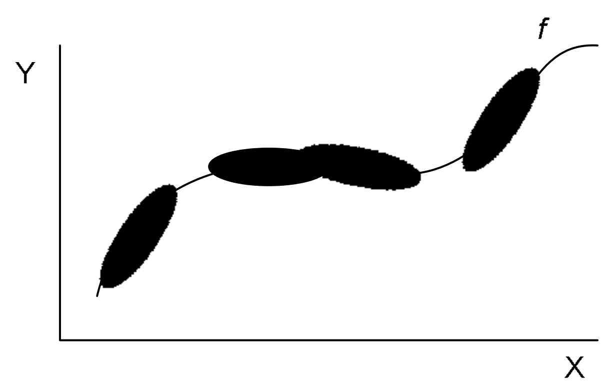


Figure 2.3 Four large fuzzy patches representing four different rules and covering part of the graph of unknown function *f:* X→Y

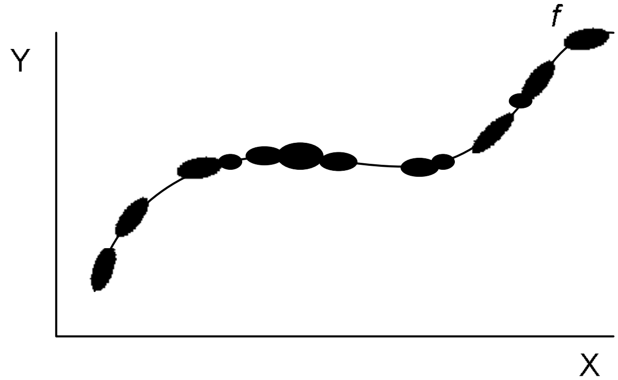


Figure 2.4 Fuzzy patches which are smaller and higher in number approximating the same function *f:* X→Y

2.3 Different methods to design Fuzzy Systems from Input-Output data

Let $(x^p{}_0; y^p{}_0)\ldots\ldots p = 1,2,3,\ldots\ldots,N$ be the input-output pairs where $y_0^p = [\alpha_y, \beta_y]$ for $y^p{}_0 \in V$ and $x_0^p = [\alpha_1, \beta_1] \times \ldots\ldots \times [\alpha_1, \beta_1]$ for $x^p{}_0 \in U$. The goal is to design a fuzzy system f(x) based on these N input-output pairs. Following are the different methods that can be used for the design [22].

2.3.1 Design of Fuzzy Systems using Recursive Least Squares Algorithm

This part of the chapter describes a recursive least squares algorithm to determine the parameters of the fuzzy system. This training algorithm minimizes the sum of matching errors for all the input output pairs up to p. So, the goal now is to design a fuzzy system f(x) such that the following function is minimized.

$$J_p = \sum_{j=1}^{p} [f(x_0^j) - y_0^j]^2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

Also if f_p is the fuzzy system designed to minimize J_p, then, f_p should be represented as a function of f_{p-1} The detailed steps for the design of the fuzzy logic system are as follows:

- **Step 1**: Suppose that $U = [\alpha_1, \beta_1] \times \ldots\ldots\ldots\ldots\ldots \times [\alpha_n, \beta_n]$. For each $[\alpha_i, \beta_i](i = 1,2,\ldots\ldots,n)$

Define N_i fuzzy sets $A_i^{l_i}(l_i = 1,2,3\ldots\ldots,N_i)$ which are complete in $[\alpha_i, \beta_i]$.

- **Step 2**: Construct a fuzzy system from the following $\prod_{i=1}^{n} N_i$ fuzzy IF-THEN rules:

IF x_1 is $A_1^{l_1}$ and x_n *is* $A_n^{j_n}$, THEN y is $B^{l_1 \ldots\ldots l_n}$ where $l_i = 1,2,3\ldots\ldots,N_i$, $i = 1,2,3\ldots\ldots,n$ and $B^{l_1 \ldots\ldots l_n}$ is any fuzzy set with the center at $y^{-l_1 \ldots\ldots l_n}$ which is free to change. For example, a fuzzy system with product inference engine, singleton fuzzifier and center average defuzzifier can be represented as shown below:

$$f(x) = \frac{\sum_{l_1=1}^{N_1} \ldots \sum_{l_n=1}^{N_n} y^{-l_1 \ldots\ldots l_n} [\Pi_{i=1}^{n} \mu_{A_i^{l_i}}(x_i)]}{\sum_{l_1=1}^{N_1} \ldots \sum_{l_n=1}^{N_n} [\Pi_{i=1}^{n} \mu_{A_i^{l_i}}(x_i)]} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

where $y^{-l_1 \ldots\ldots l_n}$ are free parameters to be designed and $A_i^{l_i}$ is already designed in step1.Collect the free parameters $y^{-l_1 \ldots\ldots l_n}$ into the $\prod_{i=1}^{n} N_i$ - dimensional vector.

$\theta = (y^{-1\ldots\ldots 1}, \ldots\ldots y^{-N_1 1\ldots\ldots 1}, y^{-121\ldots\ldots 1}, \ldots\ldots y^{-1N_2\ldots\ldots N_n}, \ldots\ldots y^{-N_1 N_2 \ldots\ldots N_n})$

Rewriting equation 1 as $f(x) = b^T(x)\theta$ where

$b(x) = (b^{1\ldots\ldots 1}(x), \ldots\ldots, b^{N_1 1\ldots\ldots 1}(x), b^{-121\ldots\ldots 1}(x), \ldots\ldots, b^{1N_2\ldots\ldots N_n}(x), \ldots\ldots b^{N_1 N_2 \ldots\ldots N_n}(x))^T$

- **Step 3: Chose the initial parameters $\theta(0)$ and the centers of THEN parts**

These $\theta(0)$ parameters can be chosen from linguistic rules from the human experts. Then choose $y^{-l_1 \ldots\ldots l_n}(0)$ which are the centers of the THEN parts.

- **Step 4:** For p=1,2,…. Compute the parameters θ using the recursive least squares algorithm [22]. The designed fuzzy system is of the form as in equation (2).

2.3.2 Design of Fuzzy Systems using Clustering

Choosing an appropriate number of rules in a fuzzy system is important because too many rules result in a too complex fuzzy system while too few rules will result in a less powerful fuzzy system. The basic idea here is to group the input/output pairs into clusters and use one rule for one cluster. First the designer constructs an optimal fuzzy system i.e. a fuzzy system which matches all the input output pairs with a certain degree of accuracy. Then determine the clusters of the input-output pairs using nearest a neighborhood clustering algorithm, before finally viewing the clusters as input-output pairs and then using the optimal fuzzy system to match them. The detailed steps are as follows:

- **Step 1:** Starting with the first input-output pair $(x_0^1; y_0^1)$ establish a cluster center x_c^1 at x_0^1 and set $A^1(1) = y_0^1, B^1(1) = 1$. Select an arbitrary radius r.

- **Step 2:** If the k[th] input-output pair $(x_0^k; y_0^k)$ is considered where k=2,3,….there are M clusters with centers at $x_c^1, x_c^2, x_c^3, ...x_c^M$ .Compute the distances of x_0^k to these M cluster centers, $\left|x_0^k - x_c^l\right|, l = 1, 2, ...M$ and let the smallest distances be $\left|x_0^k - x_c^{l_k}\right|$ that is, the nearest cluster to x_0^k is $x_c^{l_k}$ Then,

1) If $\left|x_0^k - x_c^{l_k}\right| > r$ establish x_0^k as a new cluster center $x_c^{M+1} = x_0^k$ set $A^{M+1}(k) = y_0^k, B^{M+1}(k) = 1$ and keep $A^l(k) = A^l(k-1), B^l(k) = B^l(k-1)$ for $l = 1, 2, ...M$

2) If $\left|x_0^k - x_c^{l_k}\right| \le r$ do the following

$A^{l_k}(k) = A^{l_k}(k-1) + y_0^k$

$B^{l_k}(k) = B^{l_k}(k-1) + 1$

and set $A^l(k) = A^l(k-1)$

$B^l(k) = B^l(k-1)$

for $l = 1, 2, ...M$ with $l \ne l_k$

- **Step 3:** If x_0^k does not establish a new cluster , then the designed fuzzy system based on the k input-output pairs $(x_0^j; y_0^j), j = 1, 2, 3....k$ is

$$f_k(x) = \frac{\sum_{l=1}^{M} A^l(k) \exp(-\frac{\left|x - x_c^{\,l}\right|^2}{\sigma}]}{\sum_{l=1}^{M} B^l(k) \exp(-\frac{\left|x - x_0^{\,l}\right|^2}{\sigma}]} \text{...(3)}$$

If x_0^k establishes a new cluster, then the designed fuzzy system is

$$f_k(x) = \frac{\sum_{l=1}^{M+1} A^l(k) \exp(-\frac{\left|x - x_c^{\,l}\right|^2}{\sigma}]}{\sum_{l=1}^{M+1} B^l(k) \exp(-\frac{\left|x - x_0^{\,l}\right|^2}{\sigma}]} \text{...(4)}$$

- **Step 4:** Repeat by returning to step 2 with *k=k+1*

The radius r determines the complexity of design. For a larger r, the number of clusters is less and the complexity is less too. For smaller r, the number of clusters increase and the result is a more sophisticated system. Practically a good r is obtained from trial and errors.

2.3.3 Design of Fuzzy Systems using a Table Look-Up Scheme

For any given input-output pair, $(x^p{}_0; y^p{}_0).....p = 1,2,3,....,N$ the goal here is to design a fuzzy system using a Table Look-Up scheme.

- **Step 1: Define fuzzy sets to cover the input/output spaces**

For each $[\alpha_i, \beta_i], i = 1,2,3...n$ define N_i fuzzy sets $A_i^j (j = 1,2,....,N_i)$ which are required to be complete in $[\alpha_i, \beta_i]$ that is for any $x_i \in [\alpha_i, \beta_i]$ there exists A_i^j such that $\mu_{A_i^j}(x_i) \neq 0$

- **Step 2: Generate one rule from one input-output pair**

For each input-output pair, $(x^p{}_{01}, ...x^p{}_{0n}; y^p{}_0)$ determine membership values of $x^p{}_{0i}(i = 1,2,3....n)$ in fuzzy sets $A^j{}_i (j = 1,2,...,N_i)$ and membership values of $y^p{}_0$ in fuzzy sets $B^l (l = 1,2,...,N_y)$

Then for each input variable $x_i (i = 1,2,3....n)$, determine the fuzzy set in which x_{0i}^p has the highest membership value. Finally obtain the fuzzy IF-THEN rules

IF x_1 is A_1^j and …. and x_n is A_n^j, THEN y is B^l

- **Step 3: Assign a degree to each rule generated in step 2**

Sometimes, the number of input-output pairs is too large and it is likely that there are conflicting rules. That is the rules with the same IF parts but different THEN parts. To resolve this conflict, the designer assigns a degree to each generated rule and keeps only one rule from the conflicting group that has the maximum degree. This technique also eliminates the rules which have very low importance thus saving a significant amount of computation time. In practice, the number of rules is determined from observations and an expert's advice.

- **Step 4: Create a fuzzy rule base**

A fuzzy rule base as a look up table in the two input case is illustrated in the Figure 2.5

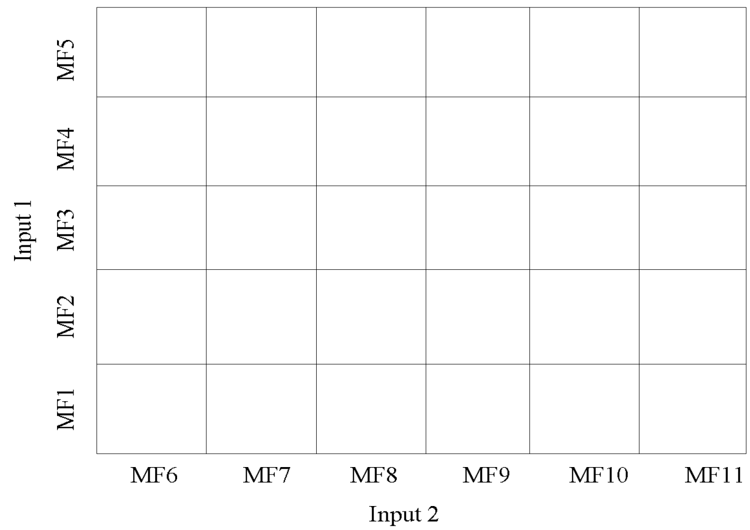


Figure 2.5 Look-up table illustration of the fuzzy rule base

With reference to Figure 2.5, MF1 to MF5 are the membership functions of the first input 'Input 1' and MF6 to MF11 are the membership functions of the second input 'Input2'. Here each box represents a combination of input fuzzy sets and output fuzzy sets and hence a possible rule.

- **Step 5:** The designer can chose any inference engine (product or minimum), any fuzzifier (Singleton, Gaussian, triangular) and any defuzzifier (Center of area, Mean of maximums) module based on the rule base created during Step4.

This approach is well suited for a situation where the number of inputs and their membership functions is not too large, i.e. the number of rules is not too large. It is also a simple and direct way to apply the available knowledge. Among all the approaches discussed above, this approach 'designing of fuzzy systems using a Table Look-Up scheme' has been used for solving the internal leakage detection problem in this thesis.

3. EVALUATION OF SYSTEM'S PERFORMANCE

References [1], [3], [4], [5] and [6] propose a model based approach to detect fault in a hydraulic system. It is a two step approach where residuals are generated using the non-linear observer in the first step and their evaluation is done in the second step. In [1] they are evaluated using the threshold approach. Both these stages involve mathematical modeling.

3.1 Background

This study directly uses the first part from [1] i.e. estimation of the state of the system using non-linear observers and generating residuals, (which is a model based approach). However, the second part which is residual evaluation is upgraded to calculating the severity of output fault using a fuzzy logic approach,(which is a knowledge based approach). Thus, this technique for fault detection is a combination of model based and knowledge based approach.

The schematic of the system under consideration is shown in Chapter 1 (figure 1.3). The mathematical modeling from [1] which is used in this study, is explored below. The nonlinear equations for the fluid flow distribution in the valve can be represented as follows,

$$q_i = \begin{cases} k_d w x_{sp} \sqrt{P_s - P_i} & x_{sp} \geq 0 (extension) \\ k_d w x_{sp} \sqrt{P_i - P_r} & x_{sp} < 0 (retraction) \end{cases} \qquad \text{..................(1)}$$

$$q_o = \begin{cases} k_d w x_{sp} \sqrt{P_o - P_r} & x_{sp} \geq 0 (extension) \\ k_d w x_{sp} \sqrt{P_s - P_o} & x_{sp} < 0 (retraction) \end{cases} \qquad \text{..................(2)}$$

Here q_i and q_o represent fluid flows into and out of the valve, respectively, w is the orifice area gradient that relates the spool displacement x_{sp} to the orifice area and k_d is the metering coefficient. P_i and P_o are the input and output line pressures, respectively, while P_s and P_r are the pump and return pressures respectively. Continuity equations for oil flow through the cylinder, neglecting the leakage flow across the actuator's piston are:

$$\left.\begin{aligned} q_i &= \frac{V_i(x)}{\beta}\dot{P}_i + A_i v \\ q_o &= -\frac{V_o(x)}{\beta}\dot{P}_o + A_o v \end{aligned}\right\} \qquad (3)$$

where $v = \dot{x}$ is the actuator's linear velocity, A_i and A_o are the piston effective areas and $V_i(x)$ and $V_o(x)$ are the volumes of the fluid trapped at the sides of the actuator, including hoses and β is the effective bulk modulus of the hydraulic fluid.

A first-order model showing the relationship between the spool displacement, x_{sp} and the input voltage u is shown in the equation (4)

$$u = \frac{\tau}{k_{sp}}\dot{x}_{sp} + \frac{1}{k_{sp}}x_{sp} \qquad (4)$$

where k_{sp} and τ are gains characterizing the dynamics of the valve. The equation defining the piston dynamics is as follows:

$$F = (P_i A_i - P_o A_o) = m\dot{v} + f_d v \qquad (5)$$

where f_d is the equivalent viscous friction and m is the mass of the actuator.

The hydraulic system being nonlinear, can be expressed in terms of following equations:

$$\begin{cases} \dot{x} = g(x(t), u(t)) \\ x(0) = x_0 \end{cases} \qquad (6)$$

From these equations the state vector is defined as follows:

$$x = [x_1 \quad x_2 \quad x_3 \quad x_4]^T = \begin{bmatrix} v & P_i & P_o & x_{sp} \end{bmatrix}^T$$

Equations (1) to (5) are rearranged as follows:

$$
\begin{cases}
\dot{x}_1 = v = \frac{1}{m}[-f_d x_1 + A_i x_2 - A_o x_3] \\
\begin{cases}
\dot{x}_2 = \dot{P}_i = \frac{\beta}{V_i}[-A_i x_1 + k_d \omega x_4 \sqrt{p_s - x_2}]; x_{sp} \geq 0 \\
\dot{x}_2 = \dot{P}_i = \frac{\beta}{V_i}[-A_i x_1 + k_d \omega x_4 \sqrt{x_2 - p_s}]; x_{sp} < 0
\end{cases} \\
\begin{cases}
\dot{x}_3 = \dot{P}_o = \frac{\beta}{V_o}[-A_o x_1 - k_d \omega x_4 \sqrt{x_3 - p_s}]; x_{sp} \geq 0 \\
\dot{x}_3 = \dot{P}_o = \frac{\beta}{V_o}[-A_o x_1 - k_d \omega x_4 \sqrt{p_s - x_3}]; x_{sp} < 0
\end{cases} \\
\dot{x}_4 = \dot{x}_{sp} = \frac{1}{\tau}[-k_{sp} - x_4]
\end{cases}
\quad \text{.............................(7)}
$$

Combining equations (3) and (5), and transforming equation (6) into a suitable form the author obtains,

$$
\left.
\begin{array}{l}
\dot{x} = g(x(t), u(t)) = Ax(t) + f(x(t), u(t)) \\
y(t) = Cx(t)
\end{array}
\right\}
\text{...(8)}
$$

where matrices *A* and *C* and the nonlinear terms *f*(x, u) of the state model are as follows (ref equation (9) and equation (10)):

$$
A = \begin{bmatrix}
-\frac{f_d}{m} & \frac{A_i}{m} & -\frac{A_o}{m} & 0 \\
-\frac{\beta A_i}{V_i} & 0 & 0 & 0 \\
\frac{\beta A_o}{V_o} & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{1}{\tau}
\end{bmatrix}
\text{..(9)}
$$

$$\begin{cases} f(x_1,u)=0 \\ \begin{cases} f(x_2,u)=\dfrac{\beta}{V_i}k_d\omega\sqrt{p_s-x_2}\,;x_{sp}\geq 0 \\ f(x_2,u)=\dfrac{\beta}{V_i}k_d\omega\sqrt{x_2-p_s}\,;x_{sp}<0 \end{cases} \\ \begin{cases} f(x_3,u)=-\dfrac{\beta}{V_o}k_d\omega\sqrt{x_3-p_r}\,;x_{sp}\geq 0 \\ f(x_3,u)=-\dfrac{\beta}{V_o}k_d\omega\sqrt{p_s-x_3}\,];x_{sp}<0 \end{cases} \\ f(x_4,u)=\dfrac{k_{sp}}{\tau} \end{cases} \quad \text{......................................(10)}$$

$$y=[1 \quad 0 \quad 0 \quad 0][x_1 \quad x_2 \quad x_3 \quad x_4]^T \text{..(11)}$$

As, seen from equation (7), the observer predicts four parameters of the system: velocity, input pressure, output pressure, and spool displacement. All of these can be used to calculate the value of the error residual. However, the author chose to make use of only one measurement, which is velocity, for residual calculation and further evaluation.

The velocity and the control signal *u* are used as observer inputs, whose dynamics are of the form:

$$\dot{\hat{x}}=\phi(\hat{x}(t),y(t),u(t)) \text{..(12)}$$

The block diagram of the overall system can be seen in the Figure 3.1 below. The state of the system predicted by the nonlinear observer is denoted by z(k), the actual state of the system is denoted by y(k) and the residual, which is the difference between the two, is denoted by e(k).

$$e(k)=M_p * y(k)-M_p * z(k)$$

where M_p is an identity matrix of size $m \times n$, $M_p = I_{4\times 1}$

The elements of the state vector $z \approx \begin{bmatrix} v & P_i & P_o & x_{sp} \end{bmatrix}^T$ are velocity v, input pressure P_i, output pressure P_o and x_{sp} spool displacement. As only the *velocity residual* is taken into account, the identity matrix M_p becomes [1 0 0 0].

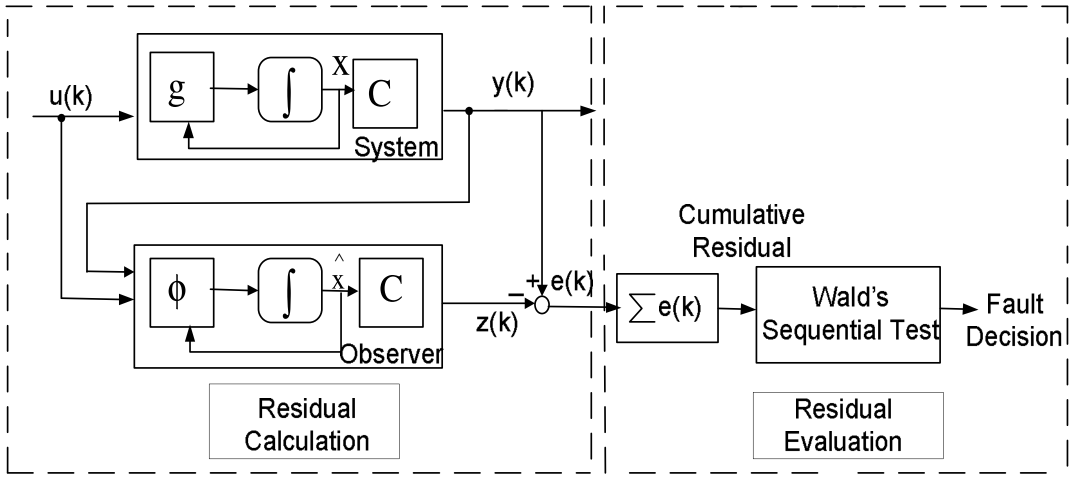


Figure 3.1 Block diagram of the original system

Theoretically, these residuals should be zero under an unfaulted condition. However, in practice, because of system noise, inexact mathematical modeling and system nonlinearity, this residual is never exactly zero even under an unfaulted condition, but will display a small value either positive or negative.

In the later part of [1] the residuals are evaluated using the Wald's Sequential test. In this statistical method, a cumulative residual below the threshold is not considered a fault. However, as it crosses the threshold, the system is considered to be faulty. (A cumulative residual is the residual added over a period of time) There are several drawbacks in this approach.

Firstly, the output is a binary decision concerning fault: present or not. Secondly, a value of cumulative residual just above the threshold is marked a fault while another residual just below the threshold is not. This might lead to missing alarms and false triggers. This information could be potentially misleading to the operators working on the hydraulic system.

This method does not consider a smooth transition between the faulty and the unfaulted condition. It does not provide any information about the state of potential fault if the cumulative residual value lies in between the thresholds. In order to reduce the effect of these issues this author will work to replace this binary logic by multi-valued one using fuzzy logic. This study will overcome the drawbacks mentioned above by replacing the output from a yes/no decision to one generating a smooth transition, in percentage of fault, using fuzzy logic. Thus, this author will replace the statistical modeling approach for residual evaluation found in [1] with a rule based inferencing method..

3.2 Role of Fuzzy Logic

The block diagram for the overall system is as shown below where u(k) is the control signal input to the system. As already seen, the difference between the expected state z(k) and the actual state of the system y(k) gives the residual e(k). The value of residual is added over a period of time which gives the cumulative residual ∑e(k). This value is subtracted from the predefined threshold and is called 'cumulative residual difference'. The 'residual' and the 'cumulative residual difference' are the two inputs to the fuzzy logic system.

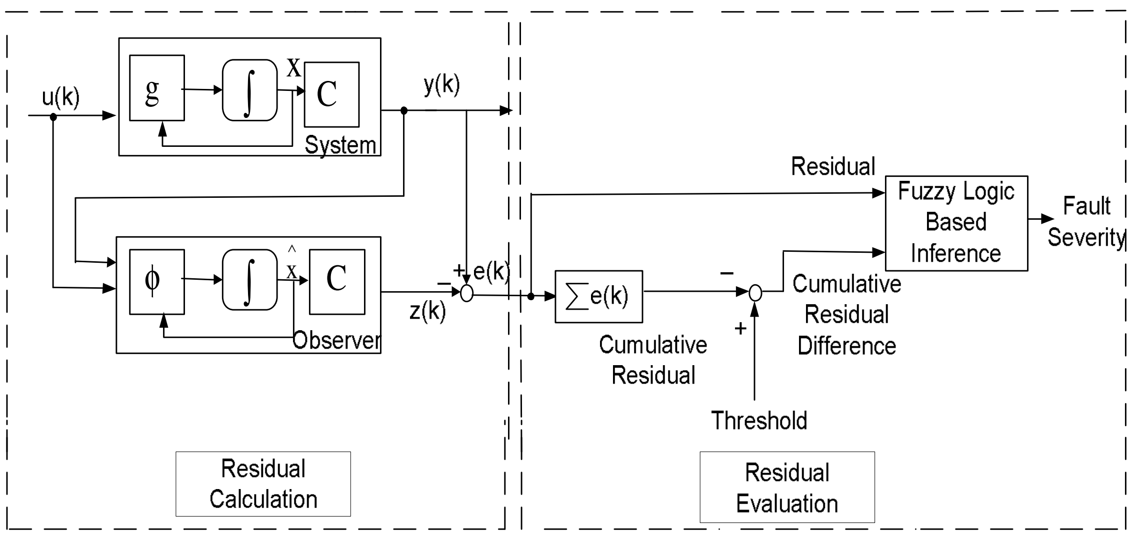


Figure 3.2: Block Diagram of the proposed system

The farther the value of cumulative residual from the upper threshold, the lower is the severity of fault. That means for lowest fault severity, the cumulative residual difference should be a positive value. Once it starts approaching the threshold, the cumulative residual difference value starts approaching zero and the severity of fault increases. And, when it crosses the threshold, its value goes into negative and the severity of fault is the highest at this time.

Similarly, the 'residual' which is the second input to our fuzzy logic system is bounded between the upper and the lower threshold. When the residual is close to zero, it will have the lowest fault severity. On the other hand, as the value of residual starts increasing by a positive or negative value and it approaches the threshold, the fault severity starts increasing too. Note that the thresholds are determined by observations. They will vary depending upon the fault tolerance of the system which in turn depends on where the system is used. Having the knowledge of these conditions, the designer can decide on the final fault severity by comparing both these inputs.

Figure 3.3 illustrates the actual and calculated velocities. The difference is due to the error introduced in the actual system by adding random noise to the *velocity* during simulation.

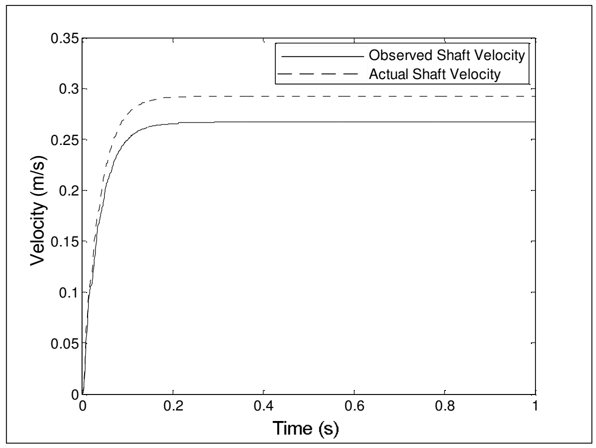


Figure 3.3 Actual velocity and observed velocity vs time

The plot of residual, cumulative residual, cumulative residual difference along with the thresholds can be seen in Figure 3.4 and Figure 3.5. As described earlier, the ‘residual’ and the ‘cumulative residual difference’ are the two inputs to the fuzzy logic controller.

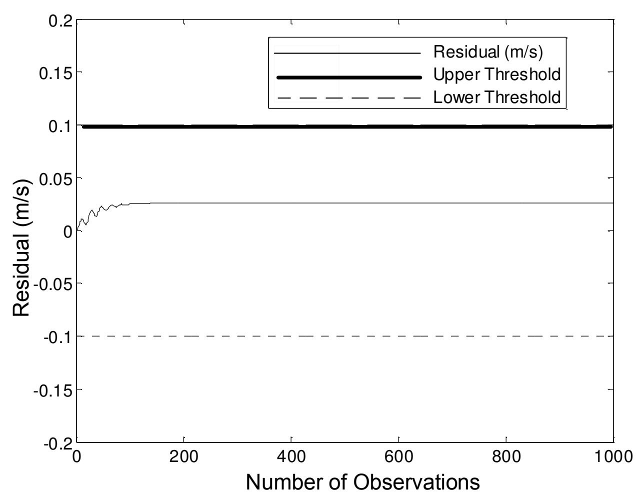


Figure 3.4 ‘Residual’ along with the upper and lower thresholds vs ‘Number of Observations’

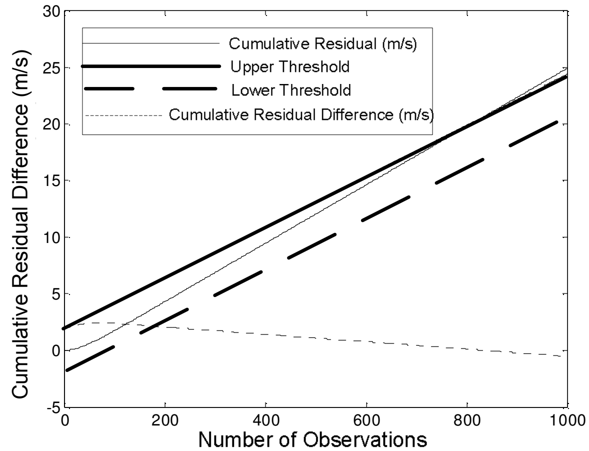


Figure 3.5 Cumulative residual and the cumulative residual difference along with the upper and lower thresholds vs 'Number of Observations'

3.3 Methodology

3.3.1 Designing Membership Functions

The first input is the computed residual has a universal space defined from -0.1 to 0.1 from observations. If the value of this residual goes beyond this range, it is considered to be in 100% fault. This input is divided into 7 membership functions namely, Big Negative (BN), Negative(N), Small Negative(SN), Zero(Z), Small Positive(SP), Positive(P) and Big Positive(BP) as shown in Figure 3.6.

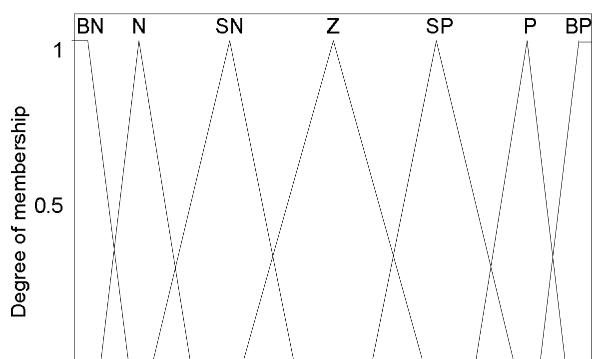


Figure 3.6 Membership functions for the first input 'Residual'

Similarly, 5 membership functions are developed for the second input which is ‘cumulative residual difference’. They are Large Negative (LNeg), Medium Negative (MNeg), Small Negative (SNeg), Zero (Zero) and Positive (POS) as seen in figure 3.7. Here too the universal space is defined from -10 to 1 from observations and a value that goes beyond this range is considered to be a 100% fault.

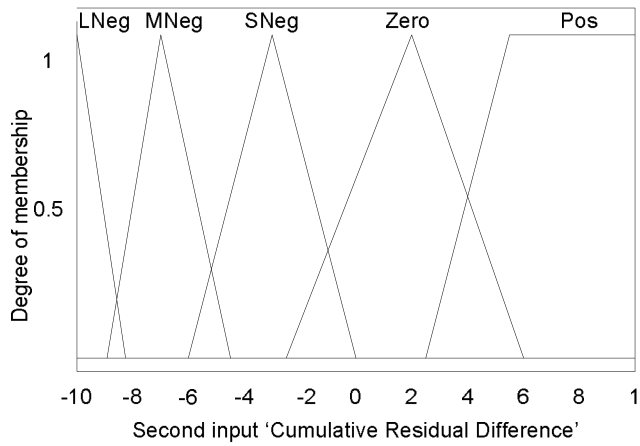


Figure 3.7 Membership functions for the second input 'Cumulative Residual Difference’

The membership functions for the output i.e. fault severity are F0, F1, F2, F3, F4, F5 and F6 where F0 represents the lowest fault severity and F6 represents the highest fault severity. Here the universal space is from zero (0) to 100 where ‘zero (0)’ represents 0% fault and 100 represents a 100% fault. The selection of the type of the membership functions and their parameters is done based on the simple guidelines suggested in [19]. The membership functions are shown in figure 3.8.

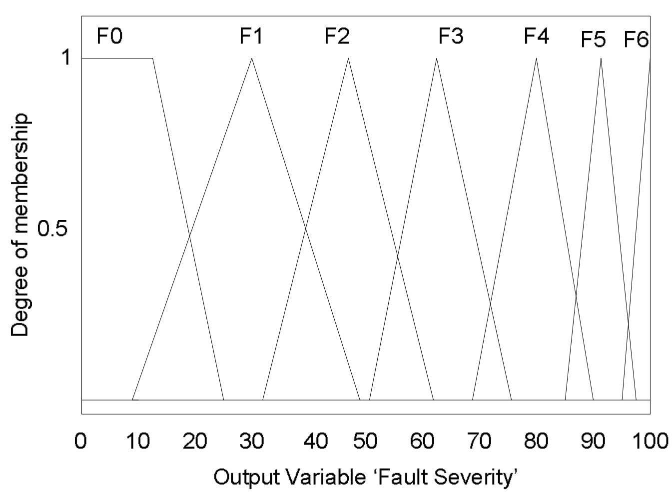


Figure 3.8 Membership functions for the output 'Fault Severity'

3.3.2 Rule Based Inferencing

Inference rules which relate the two inputs to the output were developed. They are summarized in the Table I. As seen in table I, there are in all 35 rules. For example, if the residual is Big Positive (BP) and the cumulative residual difference is Large Negative (LNeg) then the output fault severity is the highest (F6). Similarly, if the residual is Zero (Z) and the cumulative residual difference is Positive (Pos) then the output fault severity is the lowest (F0).

Table I: Rule Based Inference

Cumulative Residual Difference →	Residual →							
		BN	**N**	**SN**	**Z**	**SP**	**P**	**BP**
	Pos	F3	F2	F1	F0	F1	F2	F3
	Zero	F4	F3	F2	F1	F2	F3	F4
	SNeg	F5	F4	F3	F2	F3	F4	F5
	MNeg	F6	F5	F4	F3	F4	F5	F6
	LNeg	F6	F6	F5	F4	F5	F6	F6

3.3.3 Defuzzification

After converting all the crisp information into fuzzy the last step is to reverse that effort. Converting the fuzzy information back to crisp is known as defuzzification. The center of area/centroid method was used by this author to defuzzify these sets. This approach can be represented mathematically as follows:

$$\textit{Defuzzified value} = \frac{\Sigma f_i \bullet \mu(f_i)}{\Sigma \mu(f_i)}$$

where *fi* is the fault severity at the output and μ(*fi*) is the output membership function.

3.3.4 Simulation

This simulation was carried out in MATLAB SIMULINK using a fuzzy logic controller from the fuzzy logic toolbox. As shown in Figure 3.9, the upper subsystem represents the actual system (actual state of the hydraulic system) and the lower subsystem is the nonlinear observer (which predicts the state of the system). The SIMULINK diagram is the implementation of the block diagram shown in Figure 3.2. The simulation is carried out using a unit step input. Fault is introduced in the actual system by adding noise to the *velocity* in the actual system and different fault severities are tested at the output.

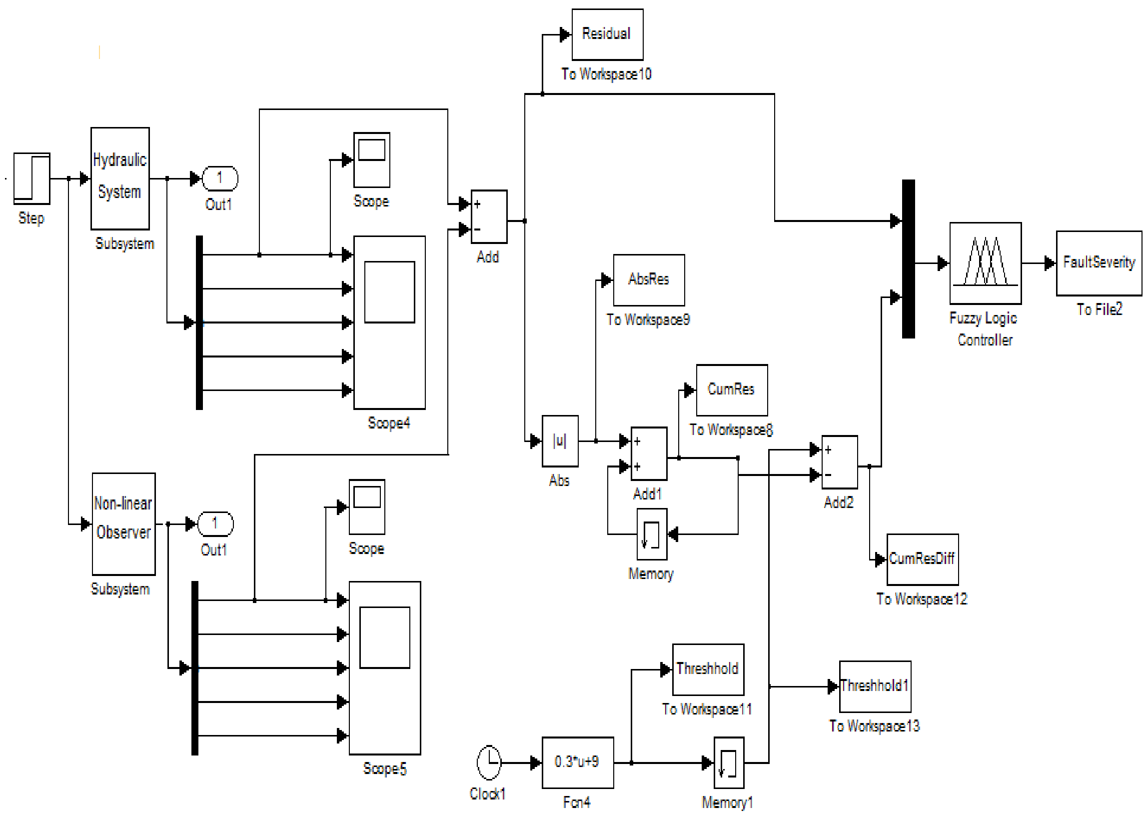


Figure 3.9 MATLAB/SIMULINK mode

3.4 Results: Rule Viewer

The rules can also be seen from the rule viewer using the fuzzy logic toolbox in MATLAB software. When the residual is 0.01, it is far away from both the upper and lower thresholds (almost at the center) and hence, has lower fault severity. Also, the cumulative residual difference is 9 which means the difference between the actual value of cumulative residual and threshold is high i.e. cumulative residual is far away from the threshold. Hence, the fault severity should be low. A combination of these values of residual and cumulative residual gives fault severity percentage of 9.96% which is low. Similarly, when the residual is 0.089 it indicates that it is very close to the threshold. A cumulative residual difference of -9 indicates that the threshold has been already crossed by the cumulative residual (hence it is negative). Both of these conditions lead to a very high fault severity of 98.4%. This can be seen with the help of the rule viewer facility in the fuzzy logic toolbox. These examples are shown in Figure 3.10 and Figure 3.11 respectively with the help of rule viewer.

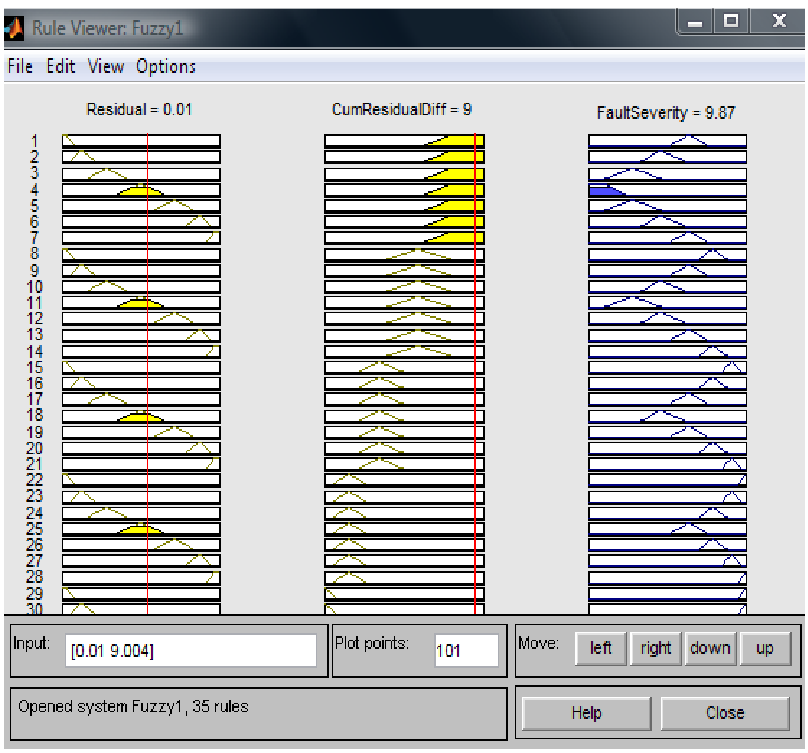


Figure 3.10 Test Results for low fault severity

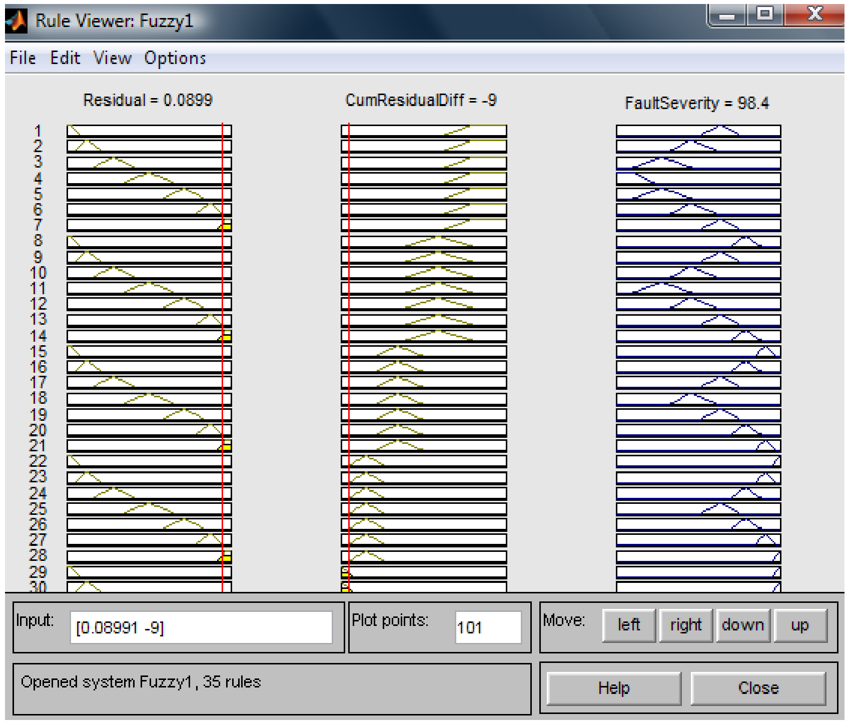


Figure 3.11 Test Results for high fault severity

3.5 Summary

The main goal here was to provide system operators continuous online information about the system's health which, in turn, would guide them to make decisions. This information needs to be given as early as possible in order to avoid any further serious damage to the system.

Using fuzzy logic over conventional method like Wald's sequential test [1] has several advantages. It provides the important information about system's health as the values move between the thresholds as well as when they are exceeded. It provides information that transitions smoothly from an unfaulted to faulty condition. This also helps to eliminate false triggers and missing alarms. Fuzzy logic is a good option because there is no general mathematical model available which describes the output fault severity based on the available inputs. . The available knowledge which is 'The amount of fault increases as the value of residuals and cumulative residuals approach the thresholds' can be directly incorporated in the design of fuzzy logic for fault detection process. This eliminates the need of detailed mathematical modeling (which would typically need development of a mathematical formula considering the residual and the cumulative residual values as variables). The method used in this study is comparatively a simple and can provide a systematic line of reasoning to draw conclusions regarding fault.

4. INTERNAL LEAKAGE FAULT DETECTION

4.1 Overview

One of the main concerns in any the hydraulic system is the leakage fault. As seen previously, leakage can be of two types depending, on their location. In internal leakage the hydraulic fluid leaks from one chamber of the cylinder in a hydraulic actuator to another. This happens due to a damaged piston seal which allows an undesired flow of fluid from one chamber directly to the other to another. The seal can be damaged for a number of reasons including wear and tear, contamination from external elements like dirt, improper installation, improper lubrication, chemical breakdown of the seal material etc. The second type of leakage is external leakage, here hydraulic fluid leaks out of the cylinder and is also caused due to damaged seals. Both types of leakage have a negative impact on the systems performance. External leakage can be directly observed but it is not possible to detect internal leakage easily until the component or full system performance starts degrading.

A new technique to determine faults by extracting feature patterns from spectral signals has recently emerged. Wavelet transforms are found to be more effective than the spectrum analysis approach in detecting faults in pumps. A wavelet transform is a signal processing method that is capable of providing the time and frequency domain information simultaneously. The time-domain signal is passed through various high pass and low pass filters, which filters out both high frequency and low frequency portions of the signal. This procedure is repeated, every time some portion of the signal corresponding to selected frequencies is removed from the signal.

In this study the project team concentrated on the detection of malfunction in the operation of the hydraulic system due to internal leakage. In [2], a method for offline diagnosis of internal leakage faults in hydraulic actuators, based on the difference in patterns observed in their wavelet transform was developed. This method uses the signal processing method of wavelet transform that breaks the original signal into shifted and scaled versions of the original signal. The pressure signal at one end of the chamber is measured. It is then decomposed into approximate and detailed wavelet coefficients using the multiresolution signal decomposition technique as described in the sentences above. It was found in the work done in [2] that a signal decomposed in such a way up to the second level, carries feature patterns whose use can distinguish between the presence or absence of internal leakage, i.e. between ‘faulty’ and ‘unfaulted’ conditions. It is found that the changes in the level two coefficients during absence of leakage are larger both in amplitude as well as energy, compared to those during leakage. Thus, the spikes that were visually noticed in the second level wavelet transform are larger, in magnitude and frequency, before leakage is introduced compared to those after the leakage is introduced. [See Figure 4.1]

The fuzzy logic system (FLS) developed here makes use of this information and the available system's data to develop an online fault detection system which is able to detect the occurrence of leakage.

4.2 Methodology

The change in the pattern in the level two wavelet transform can be seen in the Figure 4.1 after leakage was introduced manually at the 300th second.

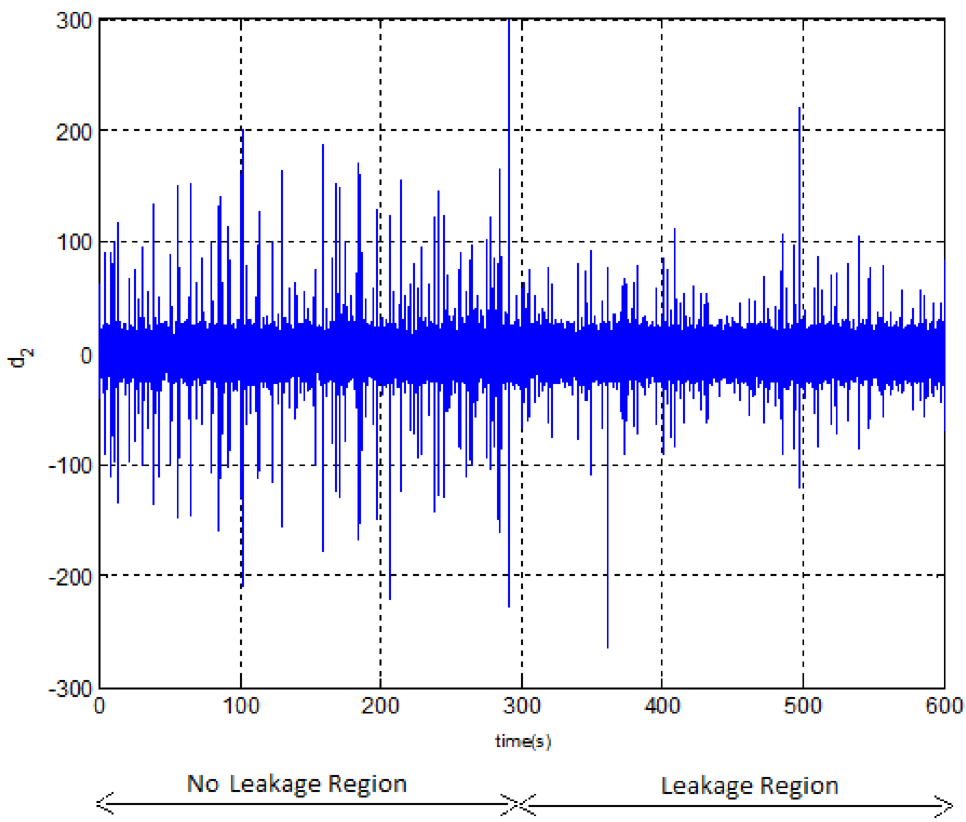


Figure 4.1: Change in pattern in the second level wavelet transform when leakage was introduced at 300th second

One can observe a decrease in the amplitude and the frequency of spikes in second level wavelet transform (d2) after the leakage was introduced after 300^{th} second. This knowledge forms a basis for the fuzzy logic detection system developed in this research.

As a first step in developing the fuzzy logic system for detection of leakage fault the designer tried to classify the spikes by their magnitude. Observing Figure 4.1 all the spikes with absolute magnitude *around* 0 to 40 are defined as small spikes, all the spikes with absolute magnitude *around* 40 to 100 as medium spikes and all the spikes with absolute magnitude greater than *about* 100 as large spikes. It was observed that there is no change seen in the number of small spikes when comparing the 'no leakage' region and the 'leakage' region. Thus, all the 'small spikes' in our decision making process were ignored. However, a significant change in the amplitude and the frequency of the medium and large spikes was observed. Hence, these two inputs were used for the fault detection algorithm. Further more, two different methods are proposed. The first one uses the 'number of medium spikes' and the 'number of large spikes' as inputs. While in the second approach the 'ratio of medium spikes' and the 'ratio of large spikes' are used as inputs. In both the methods, the fault decision is made after every 20 seconds (looking at 20 second data interval at one time). Both methods are repeated for a 60 second data interval. Finally, the results are compared at the end for different data sets.

4.3 Method I

In this approach, fuzzy logic is applied at three levels as seen in Figure 4.2. At the first level, the classification of a spike as 'Medium' or 'Large' is done using fuzzy sets 'Medium' and 'Large' instead of a crisp classification. For example, if this were a crisp definition, one could have said that a spike having absolute magnitude exactly between 40 and 100 is a 'Medium' spike and the one having absolute magnitude greater than 100 is a 'Large' spike. So, a spike of magnitude 94 will belong to the set 'Medium' completely and will not belong to the set 'Large' at all. But, due to uncertainty in both measurement and time, the designer would have to say that a spike having absolute magnitude *around* 40 to 100 is a 'Medium' spike and a spike having absolute magnitude greater than *around* 100 is a 'Large' spike. In this case, a spike of magnitude 94 will belong to the fuzzy set 'Medium' with a grade of membership 0.74 and to the fuzzy set 'Large' with a grade of membership 0.26.

Considering the fuzzy sets 'Medium' and 'Large' as inputs, the 'number of medium spikes' and the 'number of large spikes' are counted for an interval of 0-20 seconds. These two sets ‘number of medium spikes’ and ‘number of large spikes’ are the inputs for the fuzzy decision making block at the second level. Then a decision regarding fault severity from 0-20 seconds is made. Similarly, a decision regarding fault severity from 20-40sec and 40-60 seconds is made. These three decisions act as an input for the third and the final level of fuzzy decision making block where a final decision regarding the fault severity is made. The detailed design of these three levels is explained in the following paragraphs.

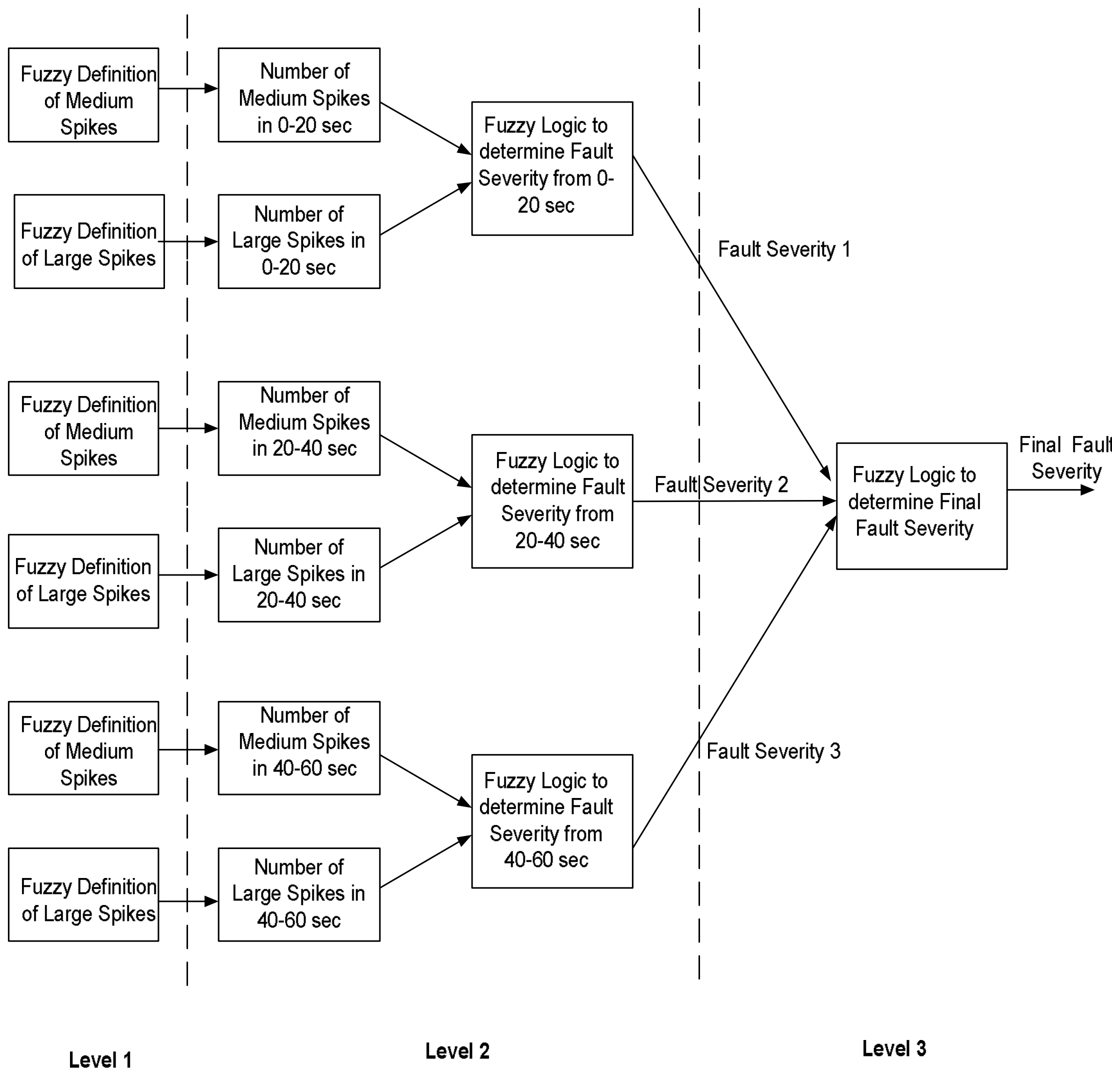


Figure 4.2: Fuzzy logic applied at three levels for decision making process (Method I)

1) Level 1

At the first level there is a fuzzy definition of a medium and large spike. Looking at the data if the medium and large spikes are defined crisply, then a spike of magnitude between 40 and 100 could be considered as a medium spike while a spike having magnitude greater than 100 could be considered as a large spike. However, these definitions involve some amount of uncertainty. So, in order to account for that, a medium spike is linguistically defined as a spike having magnitude *around* 40 to 100. So, a spike of magnitude 35 belongs to the fuzzy set 'Medium' with grade of membership 0, a spike of magnitude 40 belongs to the fuzzy set 'Medium' with grade of membership 0.5, and a spike of magnitude 45 belongs to the fuzzy set 'Medium' with grade of membership 1. Similarly, the upper boundary for the medium spike which is *around* 100 is fuzzified from 90 to 110. This can be visualized from the figure 4.3 below:

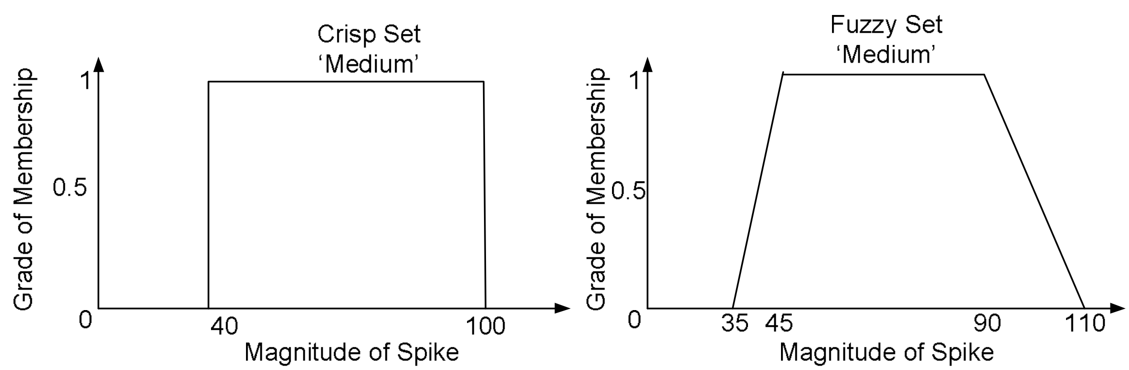


Figure 4.3 Crisp and fuzzy definitions of a medium spike

Similarly, the crisp definition of 'Large' will be a spike having magnitude greater than 100. Now, to incorporate the amount of uncertainty in this assumption, a large spike is linguistically defined as a spike with magnitude greater than *about* 100. Similar to the medium spikes, the fuzzy definition of 'Large Spike' can be seen in the figure 4.4 where a spike of magnitude 90 belongs to the fuzzy set 'Large' with a grade of membership 0, a spike of magnitude 100 belongs to the fuzzy set Large with a grade of membership 0.5 and a spike of magnitude 110 belongs to the fuzzy set Large with a grade of membership 1.

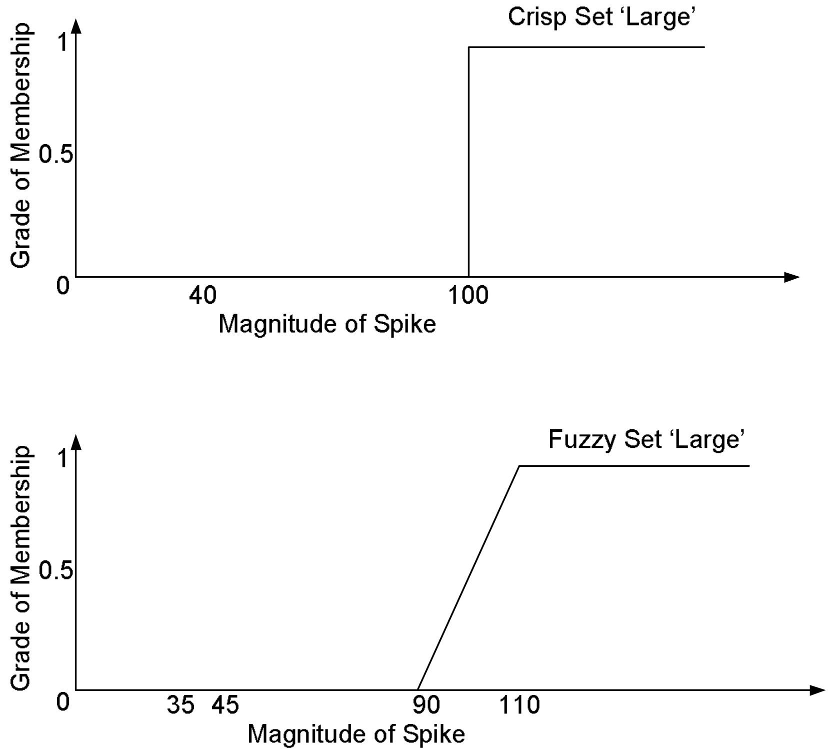


Figure 4.4 Crisp and fuzzy definition of a large spike

The universe of discourse is restricted from 0-300. That means the absolute magnitude of any spike under consideration should range from 0-300. So, any spike of absolute magnitude 'greater than 300' is considered to be of absolute magnitude of '300' and will belong to the fuzzy set 'Large' with a degree of membership 1.

To summarize the first fuzzy level, two fuzzy sets 'Medium' and 'Large' are defined and the belonging of any spike to any of these sets is not 0 or 1 but is a matter of degree. Going forward, these definitions will be helpful in counting the 'Number of Medium Spikes' and 'Number of Large Spikes'.

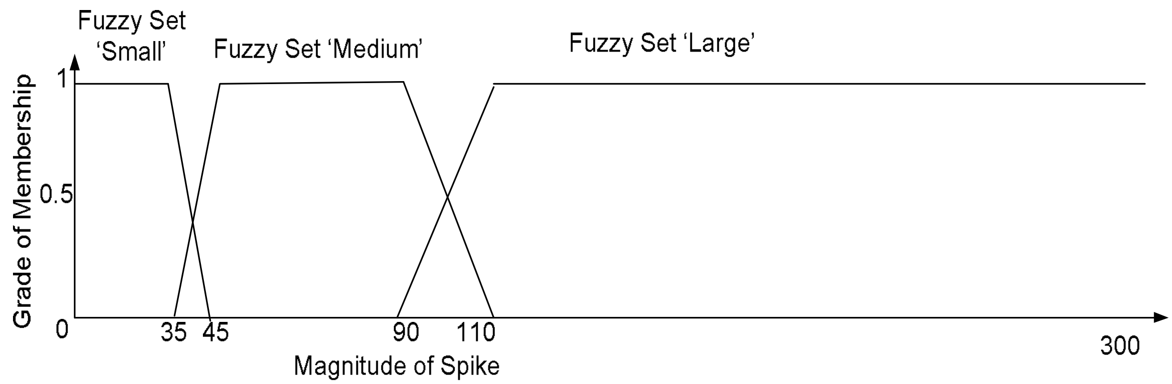


Figure 4.5: Defining fuzzy sets 'Medium' and 'Large'

2) Level 2

After defining a spike as medium or large or belonging to both to a certain degree, the total 'number of medium spikes' and the total 'number of large spikes' is counted for a time period of 0-20 seconds. After counting the resultants which are total number of medium spikes 'NumMedSpikes' and the total number of large spikes 'NumLargeSpikes', serve as inputs to the second level FLS. For example, a spike having magnitude 94 in the 0-20 second interval, will add 0.74 to the set 'NumMedSpikes' and 0.26 to the set 'NumLargeSpikes'. (Note: If this was a crisp definition, it would have been right to add 1 to the set 'number of medium spikes' and 0 to the set 'number of large spikes'). Similarly, each spike in the interval is inspected for its magnitude and belongingness to the fuzzy sets 'NumMedSpikes' and/or 'NumLargeSpikes'.

The universal space for the first input 'NumMedSpikes' is considered to be from 0-300. If that number exceeds 300, then it is considered to be 300. Four membership functions are defined in this universal space for the input 'NumMedSpikes' namely,

- Zero (Zero), Trapezoidal, parameters: [0 0 7 16]
- VeryVeryFew (VVFew), Triangular, parameters: [4 30 58]
- VeryFew(VFew), Triangular, parameters: [31 60 90]
- Few(Few), Trapezoidal,parameters: [68 100 300 300]

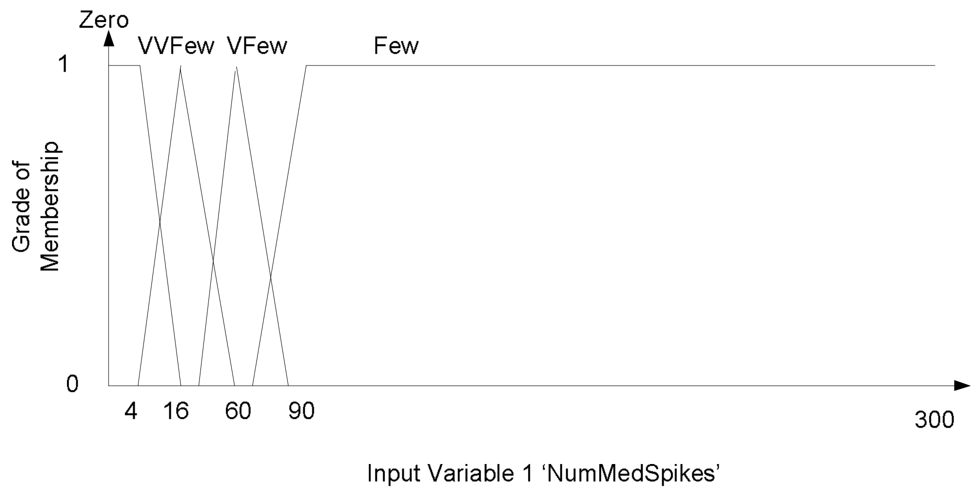


Figure 4.6 Membership function plot for first input variable 'NumMedSpikes'

Similarly, the universe of discourse for the number of large spikes 'NumLargeSpikes' is from 0-100. Again if the number exceeds 100, then it is considered to be 100 for the purpose of defining the universal space. The membership functions defined for the second input 'NumLargeSpikes' are

- Zero (Zero), Triangular, parameters: [0 0 1 3]
- VeryVeryFew (VVFew), Triangular, parameters: [1.5 4 8]
- VeryFew(VFew), Triangular, parameters: [4 8 15]
- Few(Few), Trapezoidal, parameters: [10 15 100 100]

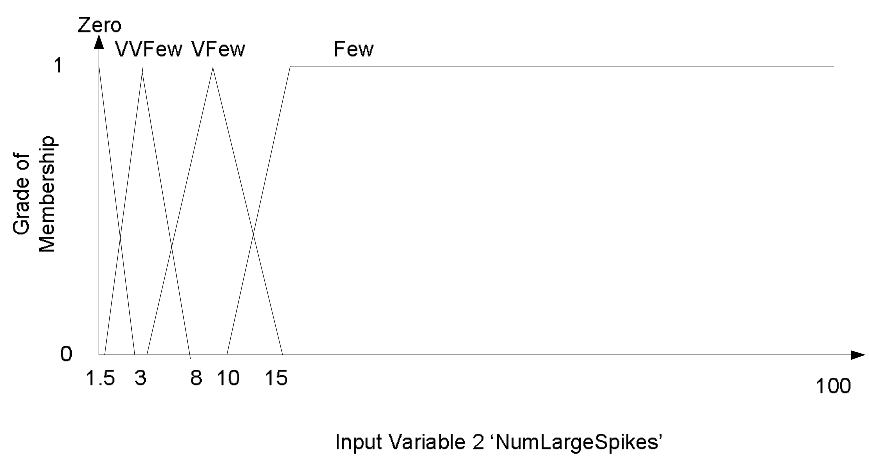


Figure 4.7 Membership function plot for second input variable 'NumLargeSpikes'

These values of the universal space and the parameters for the membership functions are decided based on the available data. Looking at the data, one can distinguish between the number of medium spikes in the area before the leakage was introduced and compare it to that after the leakage was introduced. Similarly, looking at the number of large spikes in those two areas from the available data, the universe of discourse and the parameters of membership functions are decided. A substantial amount of trial and error is needed as the available data contains an arbitrary amount of inaccuracy.

Based on these two inputs: number of medium spikes 'NumMedSpikes' and number of large spikes 'NumLargeSpikes', the decision regarding the fault severity is made at the output for 0-20 seconds. The output is a fault severity on a scale 0-1 where 0 represents 0% fault and 1 represents 100% fault. Hence, the universe of discourse for the output fault severity ranges from 0-1. The names of membership functions, their shape and the parameters are given below:

- Low, Trapezoidal, parameters: [0 0 0.14 0.5]
- Medium, Trapezoidal, parameters: [0.14 0.4 0.6 0.8]
- High, Trapezoidal, parameters: [0.5 0.8 10]

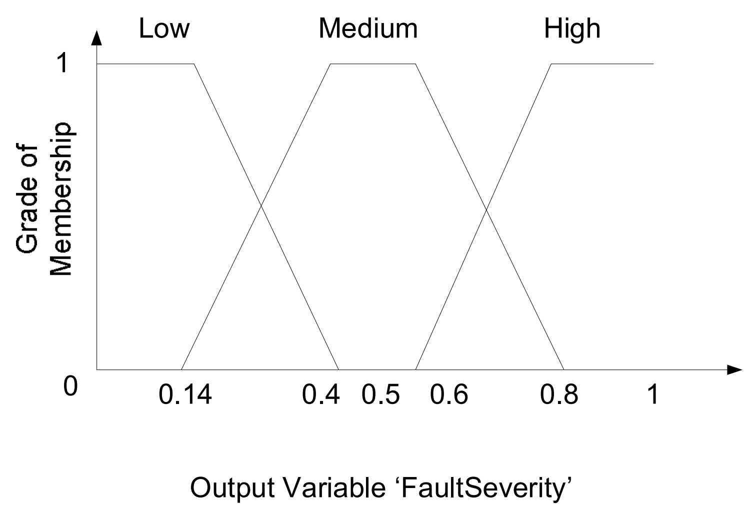


Figure 4.8 Membership function plot for output variable 'Fault Severity'

The rules that relate the first input variable 'NumMedSpikes' and the second input variable 'NumLargeSpikes' to the output 'Fault Severity' are shown in table II.

Table II. Fuzzy Rule Base relating the two inputs ‘NumLargeSpikes’ and ‘NumMedSpikes’ to the output ‘Fault Severity’

	Second input variable ‘NumLargeSpikes’->				
First input variable ‘NumMedSpikes’ ->		**Zero**	**VVFew**	**VFew**	**Few**
	Zero	Yes	Yes	Yes	Maybe
	VVFew	Yes	Yes	Maybe	No
	VFew	Yes	Maybe	No	No
	Few	Maybe	No	No	No

As the number of medium spikes or large spikes decreases from Few to Zero, the output fault severity increases from no to yes. This is the fuzzy logic representation of the basic assumption made by looking at the second level wavelet transform which is:

The number of medium and large spikes decreases in amplitude and frequency after the leakage is introduced.

Thus, the available information from knowledge and data is incorporated into the fuzzy logic fault detector to make the decision regarding fault. The fuzzy inference engine uses these rules and the membership functions to make inferences.

3) Level 3

After defuzzification, the result obtained is a crisp value denoting the actual fault severity of the fault at the output. This output fault severity called ‘FaultSeverity1’ is calculated looking at the number of medium spikes and large spikes in a data interval of 0-20. Same method is used to calculate the output fault severity for the 20-40 second interval called ‘FaultSeverity2’ and then 40-60 second interval called ‘FaultSeverity3’. At the end of 60 seconds, a final third level FLS decides a final output fault severity based on these three inputs which are the outputs from the previous level, level2.

The membership functions for each of these three outputs i.e. ‘FaultSeverity1’, ‘FaultSeverity2’ and ‘FaultSeverity3’ are assigned as Low, Medium and High and they act as inputs for final fault decision. As the inputs and output both are fault severities the universal space for them is from 0-1 where 0 represents 0% fault and 1 represents 100% fault. Hence, the universe of discourse for all three inputs and the final output is from 0-1. The names of membership functions, their shape and the parameters are same for all three inputs shown below:

- Low(L), Trapezoidal, parameters: [0 0 0.2 0.4]
- Medium(M), Trapezoidal, parameters: [0.2 0.4 0.6 0.8]
- High(H), Trapezoidal, parameters: [0.6 0.8 1 1]

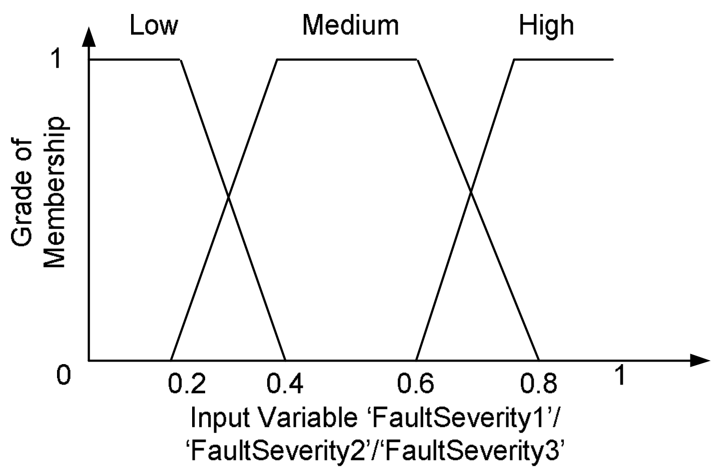


Figure 4.9 Membership function plot for input variables ‘Fault Severity1’/‘Fault Severity2’ /‘Fault Severity3’

Finally for output which is ‘FinalFaultSeverity’, the membership functions, their shape and the parameters are

- No(N), Trapezoidal, parameters: [0 0 0.2 0.4]
- Maybe(M), Trapezoidal, parameters: [0.2 0.4 0.6 0.8]
- Yes(Y), Trapezoidal, parameters: [0.6 0.8 1 1]

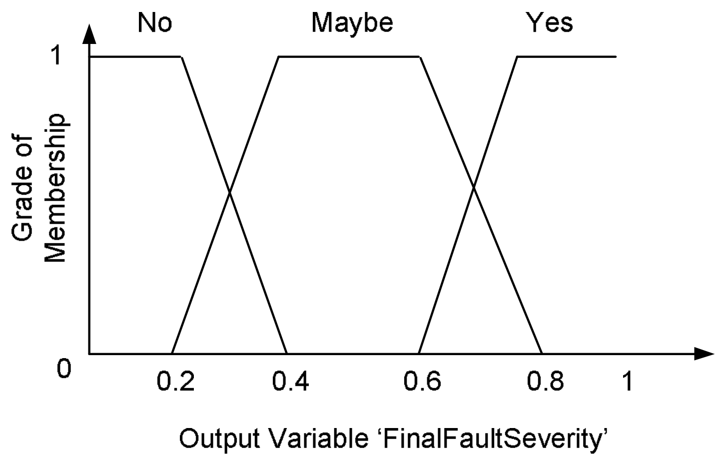


Figure 4.10 Membership function plot for output variable ‘FinalFaultSeverity’

The rules that relate the three inputs ‘Fault Severity1’, ‘Fault Severity2’ and ‘Fault Severity3’ to the output ‘FinalFaultSeverity’ are summarized in the table III:

Table III Rules that relate the three inputs ‘Fault Severity1’, ‘Fault Severity2’ and ‘Fault Severity3’ to the output ‘FinalFaultSeverity’

Rule No.	Input 1-2-3	Output		Rule No.	Input 1-2-3	Output		Rule No.	Input 1-2-3	Output
1	L-L-L	N		10	M-L-L	N		19	H-L-L	N
2	L-L-M	N		11	M-L-M	N		20	H-L-M	M
3	L-L-H	Y		12	M-L-H	Y		21	H-L-H	Y
4	L-M-L	N		13	M-M-L	M		22	H-M-L	N
5	L-M-M	N		14	M-M-M	M		23	H-M-M	M
6	L-M-H	Y		15	M-M-H	Y		24	H-M-H	Y
7	L-H-L	M		16	M-H-L	M		25	H-H-L	M
8	L-H-M	M		17	M-H-M	Y		26	H-H-M	Y
9	L-H-H	Y		18	M-H-H	Y		27	H-H-H	Y

These rules are developed from the general observations looking at the data in 3 consecutive intervals of 20 seconds. For example, if the fault severity in all the 3 intervals is low, it indicates that there is least probability of that leakage fault is present. So, in this case, the final output severity is ‘No’. This can be seen in rule no. 1.

Similarly, if the fault severity is high especially for the 3rd time slot, then there is high probability that the leakage fault is present. So, in all those cases which have a high fault severity in the third a slot, the final fault severity is high. So, the final output severity is “Yes” in that case. This can be seen in rule no. 3, 6, 9, 12, 15, 18, 21, 24 and 27.

However, if the fault severity is high in the first time slot and low in both the next, then there is very less probability that the leakage fault is present. So, the final output severity is “No” in that case. This can be seen in rule no. 19.

After defuzzification, a crisp value denoting the final severity of the fault at the output is obtained. As, the first fault severity is looking at the 0-20 second data interval, 2nd looking at 20-40 second data interval and 3rd looking at the 40-60 second data interval, the 1st final decision is made at the end of 60th second.

This process is repeated for next three time slots of 20-40 seconds,40-60seconds and 60-80 seconds and to obtain final fault severity at the end of 80th second. The next decision is made looking at the three time slots of 40-60seconds, 60-80 seconds and 80-100 seconds and to obtain final fault severity at the end of 100th second. This process continues and a final decision regarding fault severity is obtained at the end of each 20 second data interval. This is represented in figure 4.11.

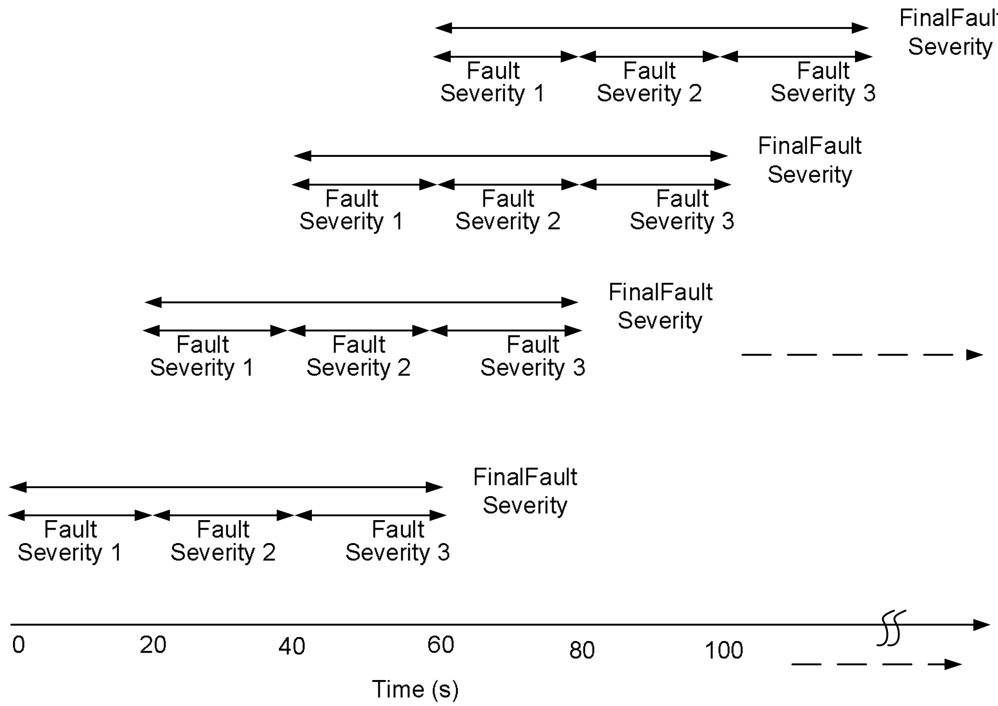


Figure 4.11 Final decision regarding fault severity made at the end of each 20 second data interval

The same process is repeated for a time interval of 60 seconds instead of 20. The membership function parameters differ in that case because the number of medium and large spikes found in an interval of 20 seconds is not same as that in a 60 second interval. However, the shape of the parameters and everything else remains the same. The results of this algorithm on different data sets for 20 second and 60 second data interval are compared in the consecutive chapter.

Also, note that in this case an output is obtained after every 60 seconds i.e. first output will be seen at 60th second, 2nd at 120th second, 3rd at 180th second and so on.

4.4 Method II:

Method II has the same overall concept as in Method I except for a couple of differences. The first difference could be found in level two where instead of using the ‘number of medium spikes’ and ‘number of large spikes’ as inputs for determining the fault severity the ‘ratio of medium spikes’ and the ‘ratio of large spikes’ are used as inputs.

$$\text{Ratio of medium spikes} = \frac{\text{Number of medium spikes}}{\text{Total Number of spikes (small + medium + large)}}$$

$$\text{Ratio of large spikes} = \frac{\text{Number of large spikes}}{\text{Total Number of spikes (small + medium + large)}}$$

The second change found at level three is that instead of using 0-20 seconds, 20-40 seconds and 40-60 seconds data interval, for final decision, a data interval of 0-20 seconds, 0-40 seconds and 0-60 seconds data interval is used for final decision. This continues for the next cycle which is from 20-40, 20-60 and 20-80 seconds and so on. Because they are ratios, the data is more normalized as compared to the earlier method. These changes are highlighted in Figure 4.12

1) Level 1

Level one is everything same compared to the first method where fuzzy sets ‘Medium’ and ‘Large’ are obtained which act as two inputs for the next level.

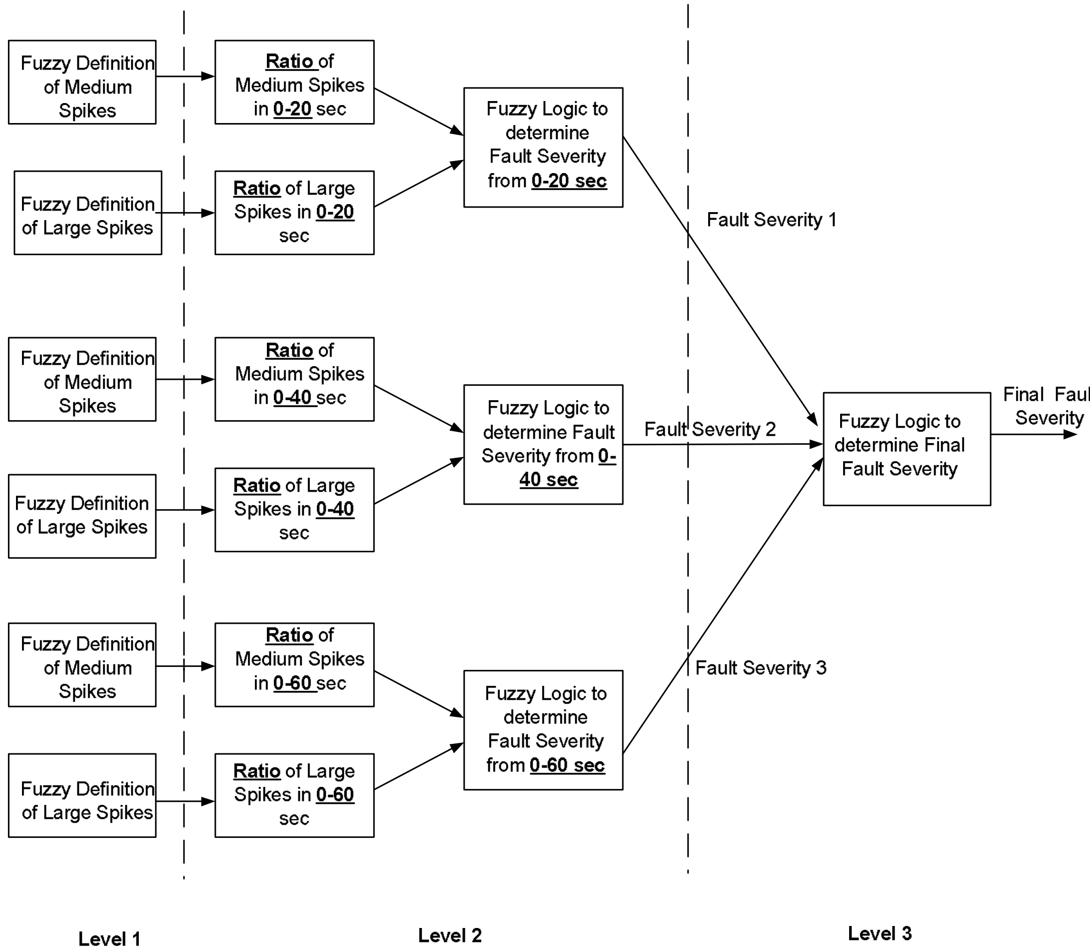


Figure 4.12: Fuzzy logic applied at three levels for decision making process (Method II)

2) Level 2

After defining a spike as medium or large or belonging to both to a certain degree, the ‘ratio of medium spikes’ and the ‘ratio of large spikes’ is calculated for a time period of 20 seconds, 40 seconds and 60 seconds respectively. These ratios are the ratios of number of medium or large spikes to the total number of spikes present in the 20 second, 40 seconds and 60 seconds data interval. These ratios are denoted by ‘RatioMedSpikes’ and ‘RatioLargeSpikes’ respectively which are inputs to the second level FLS.

The universal space for the first input ‘RatioMedSpikes’ is considered to be from 0-100. If the number exceeds 100, then it is considered to be 100. Four membership functions are defined in this universal space for the input ‘NumMedSpikes’ namely,

- Zero , Trapezoidal, parameters: [0 0 4 10]
- Small , Triangular, parameters: [4 12 20]
- Medium, Triangular, parameters: [14 20 30]
- Large, Trapezoidal, parameters: [22 30 100 100]

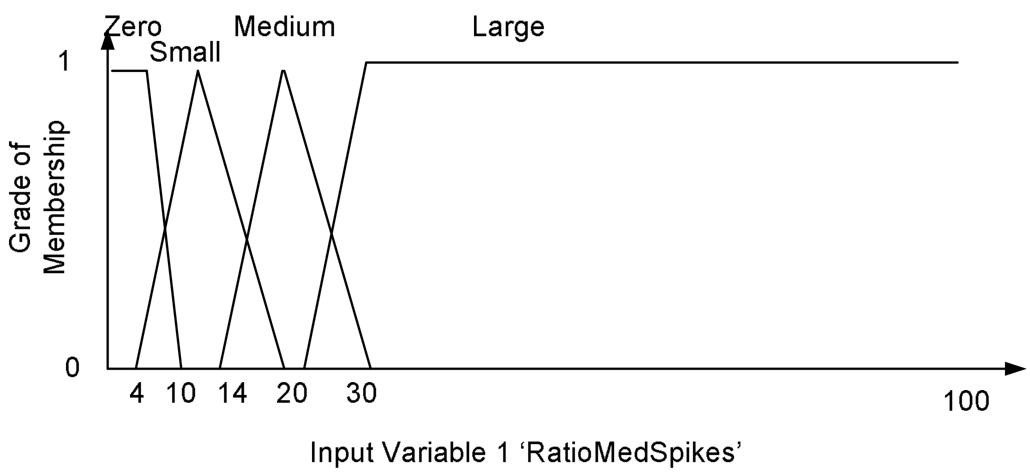


Figure 4.13 Membership function plot for first input variable ‘RatioMedSpikes’

Similarly, the universe of discourse for the ratio of large spikes ‘RatioLargeSpikes’ is from 0-10. Again if the number exceeds 10, then it is considered to be 10 for the purpose of defining the universal space. The membership functions defined for the second input ‘RatioLargeSpikes’ are

- Zero , Trapezoidal, parameters: [0 0 0.5 1]
- Small , Triangular, parameters: [0.5 1.2 2]
- Medium, Triangular, parameters: [1.4 2 3]
- Large, Trapezoidal, parameters; [2 3 10 18]

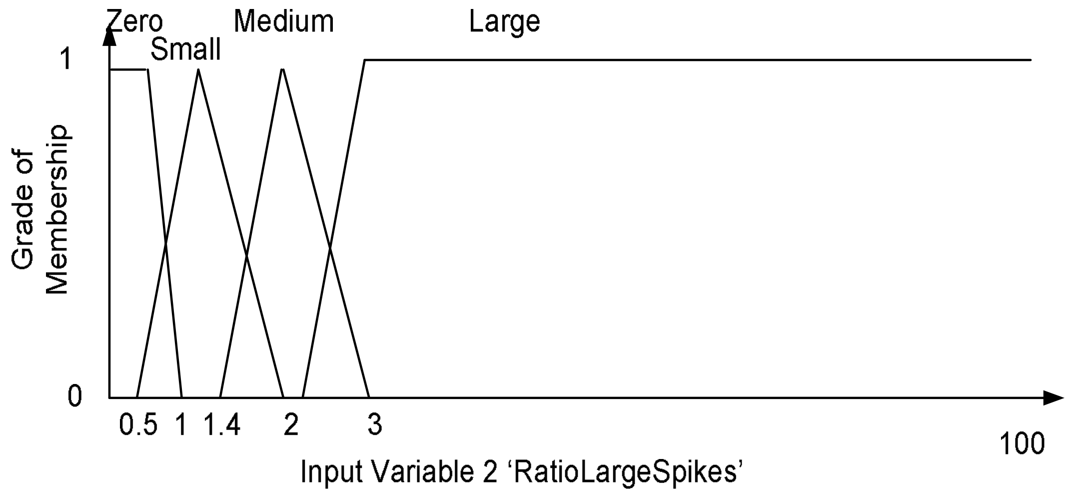


Figure 4.14 Membership function plot for second input variable 'ratioLargeSpikes'

Based on these two inputs: ratio of medium spikes 'RatioMedSpikes' and ratio of large spikes 'RatioLargeSpikes', the decision regarding the fault severity is made at the output for 0-20 seconds first, then 0-40 seconds and then 0-60 seconds in the 1st iteration. The output is a fault severity on a scale 0-1 where 0 represents 0% fault and 1 represents 100% fault. Hence, the universe of discourse for the output fault severity is from 0-1. The rules and the inferencing methods are same as those discussed in method I.

The names of membership functions, their shape and the parameters are as follows:

- Low, Trapezoidal, parameters: [0 0 0.14 0.5]
- Medium, Trapezoidal, parameters: [0.14   0.4   0.6   0.8]
- High, Trapezoidal, parameters: [0.5 0.8 10]

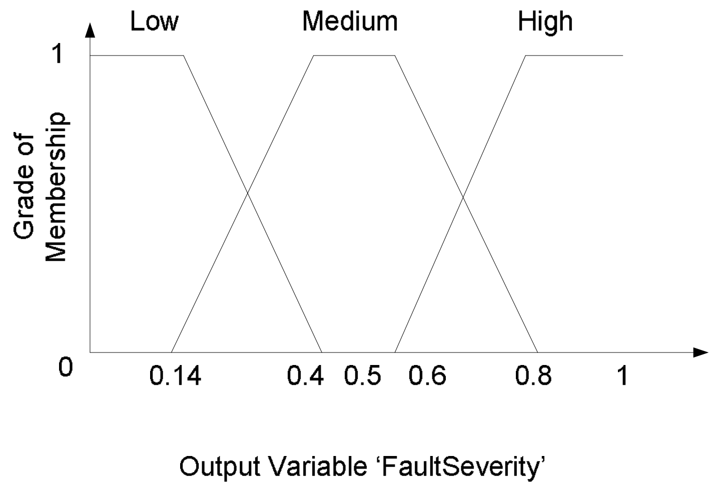
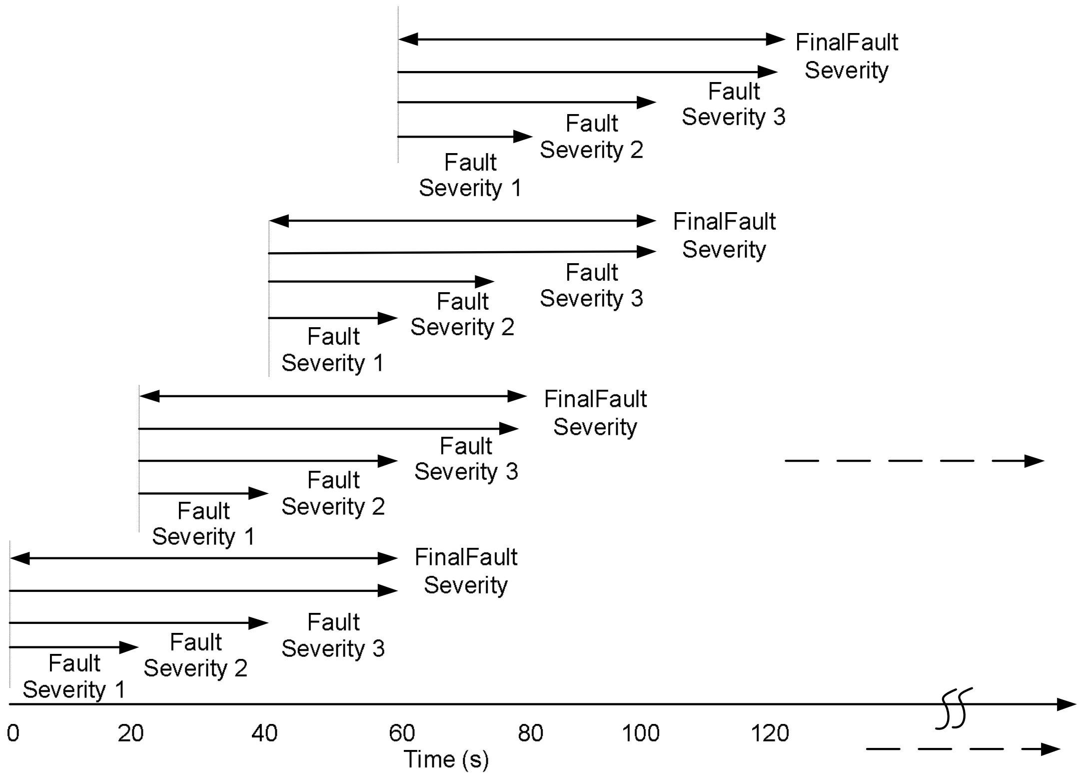


Figure 4.15 Membership function plot for output variable 'Fault Severity'

3) Level 3

The second change from method I is the data interval. Instead of taking intervals from 0-20,20-40 and 40-60, time intervals from 0-20, 0-40 and 0-60 are taken into consideration. This repeats for detection of each 'FinalFaultSeverity' as shown in the figure 4.16. This second method is again repeated considering a time interval of 60 seconds instead of 20.



F

igure 4.16 Final decision regarding fault severity made at the end of 20 second data interval, 40 second data interval and 60 second data interval during each iteration

5. RESULTS AND CONCLUSION

5.1 Results

Data samples were collected for a time frame of 0 to 600 seconds where the leakage was manually introduced. The algorithm developed was tested on these data samples and the results obtained are discussed in this chapter.

The system designed here is capable of detecting an average leakage of 0.2 lit/min or more. This is because as the leakage decreases the changes in the feature patterns in the wavelet transform that enable the algorithm to detect the leakage decrease too.

This capability is mainly determined from 16 samples of available data The step-by-step application of algorithm to one of the data sample (Sample no. 12) is explained first. The variety of results obtained (good, acceptable or not acceptable) are demonstrated using three data samples (Sample no. 1, Sample no. 4 and Sample no. 10). The chapter concludes with the summary of results obtained for all data samples.

Figure 5.1 shows the average leakage of about 0.22 lit/min introduced manually at 300th second during the 600 second data interval. The x axis represents the time in seconds and the y axis represents the leakage that was introduced in lit/min.

Figure 5.2 shows the second level wavelet transform of the pressure signal measured at one end of the hydraulic chamber. As noticed, the number of spikes and their length is greater in the time period of 0-300 sec (where there was no leakage present). On the other hand the number of spikes and their length have decreased in the latter half time period of 300-600 seconds (where the leakage was introduced). This knowledge is the basis of all the algorithms developed for the leakage detection.

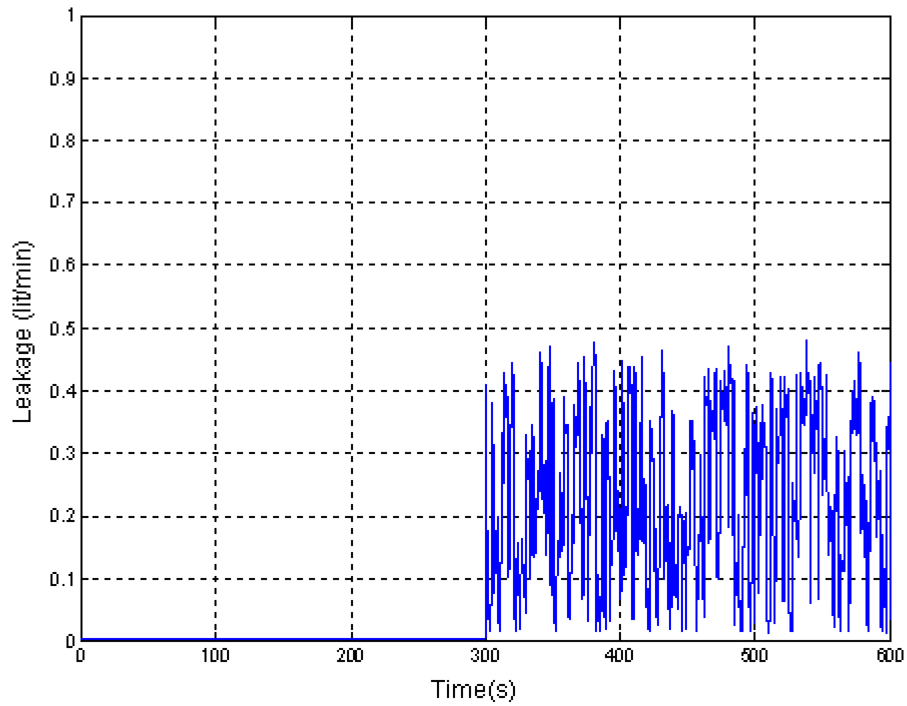


Figure 5.1 Plot of actual leakage introduced at 300th second versus time

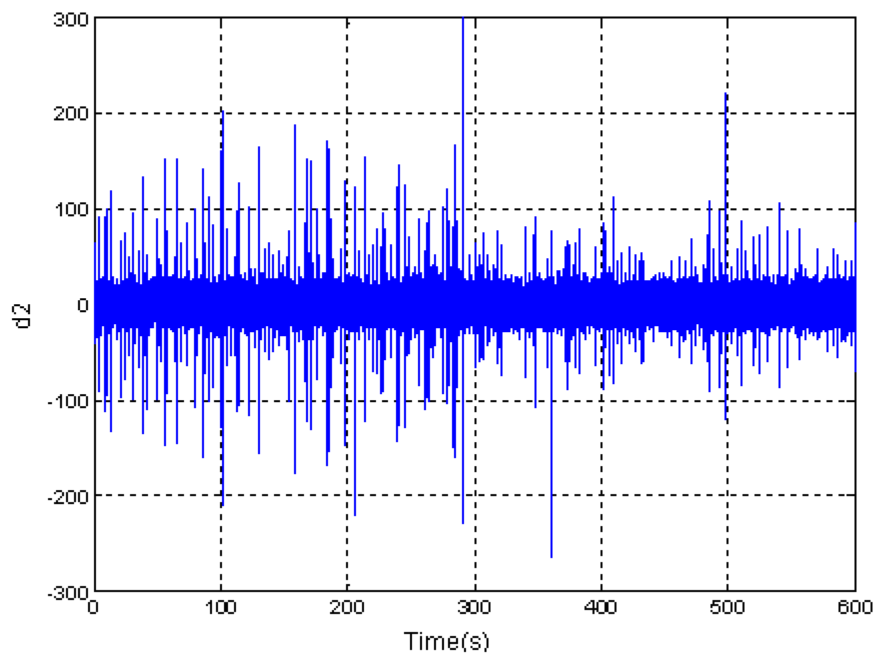


Figure 5.2 Second level wavelet transform of the pressure signal taken at one end of the chamber

Following flowcharts represent the algorithm i.e. the methodology explained in the previous chapter. Figures 5.4 and 5.7 represent the algorithm for Method I and Method II respectively. Figure 5.3 is the subroutine which is an equivalent representation of the Level I mentioned in chapter 4 which is the same for both the methods.

In figure 5.3, is a subroutine called 'Inner Loop'(Equivalent of Level I in the previous chapter). In this flowchart. As a first step, the universal space for magnitude of the spike is defined A spike of magnitude 300 or more is considered as 300 for analysis. In the next step, a 'small' spike is discarded. A grade of membership to the fuzzy sets 'medium' and 'large' is calculated for each spike which is added over a time interval decided by outer loop.

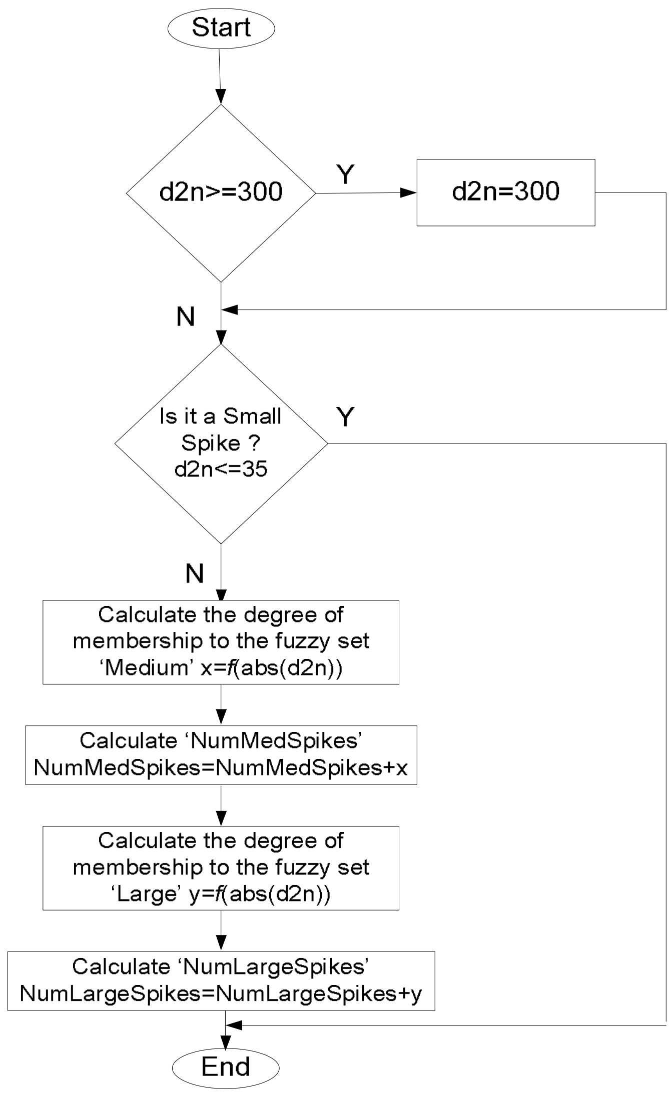


Figure 5.3 Subroutine 'Inner Loop' used by the main algorithm

Figure 5.4 represents the algorithm for Method I. As seen from the flowchart, the input to the algorithm is the second level wavelet transform (n, t_n, d_{2n}). The Fault Severities are calculated at the end of each iteration. The data interval can be chosen to be y=20 or y=60 seconds.

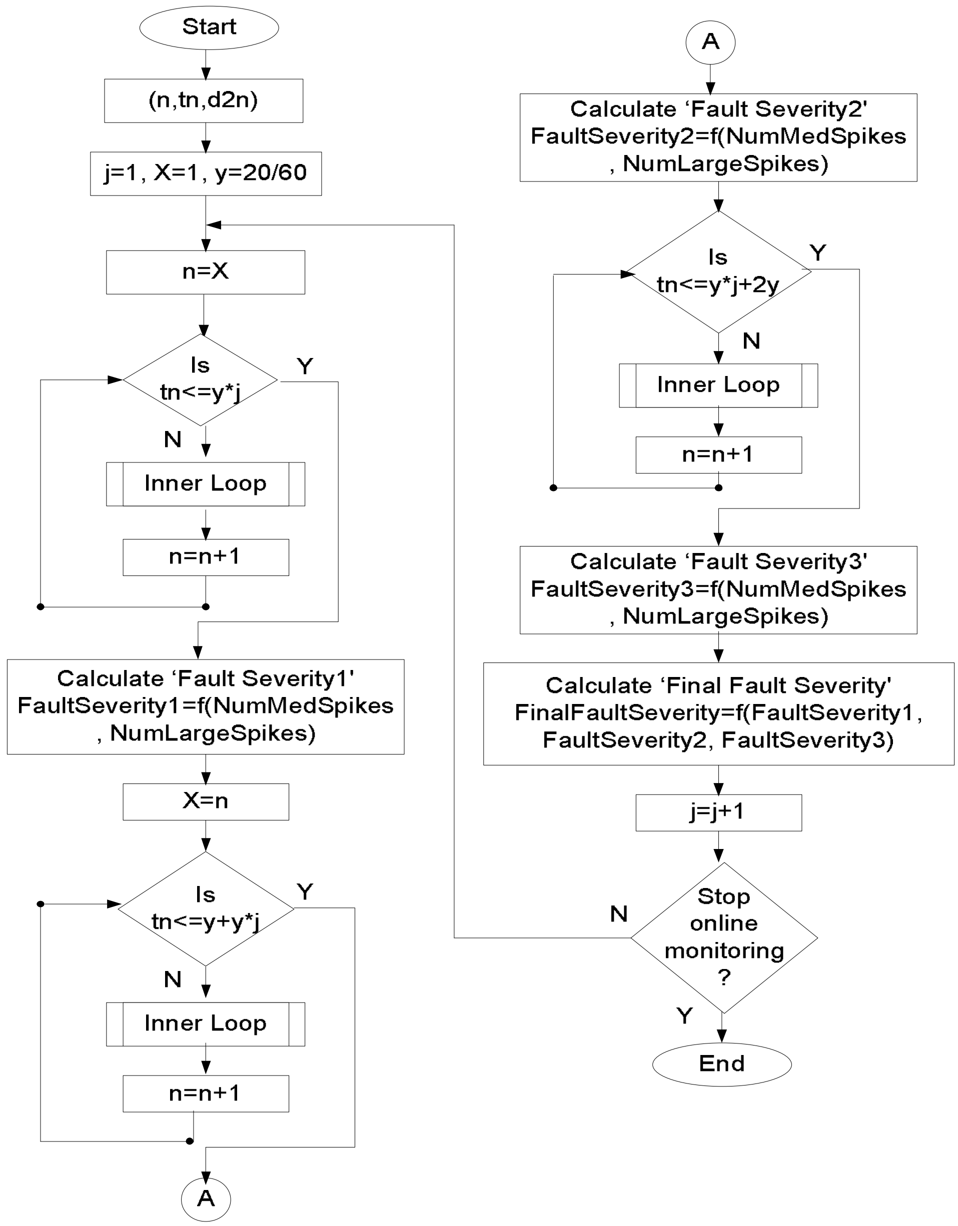


Figure 5.4 Method I Algorithm Flowchart for 20/60 second data interval

The result of application of the algorithm described in the flowchart above on data sample no. 12 can be seen in the following figures. Figure 5.5 shows the result after applying Method I on data sample no. 12 which uses number of spikes in 20 second data interval as input. The 1st plot represents the output i.e. the leakage decision (in percentage) after level 2 described in the previous chapter. The 2nd plot represents the final output i.e. final leakage decision (in percentage) after level 3 again described in the previous chapter. The 3rd plot is conversion of the final decision into a 3 level digital decision. (A value that falls in between 0 to 0.33 in the 2nd plot is represented by 0, similarly a value in between 0.33 to 0.66 is represented by 0.5 and a value in between 0.66 to 1 is represented by 1).

Recall that in case of Method I at level 2, the decision regarding leakage/fault is made using 'Number of medium spikes' and 'Number of large spikes' as inputs for fuzzy logic system for each 20 second data interval. Similarly, the fault decisions made by three consecutive 20 second data intervals act as inputs to make a final decision regarding fault. The sudden drop in output at 500th second (in 1st and 2nd graph) is due to presence of a 'non-leakage' pattern in leakage area in that particular time interval.

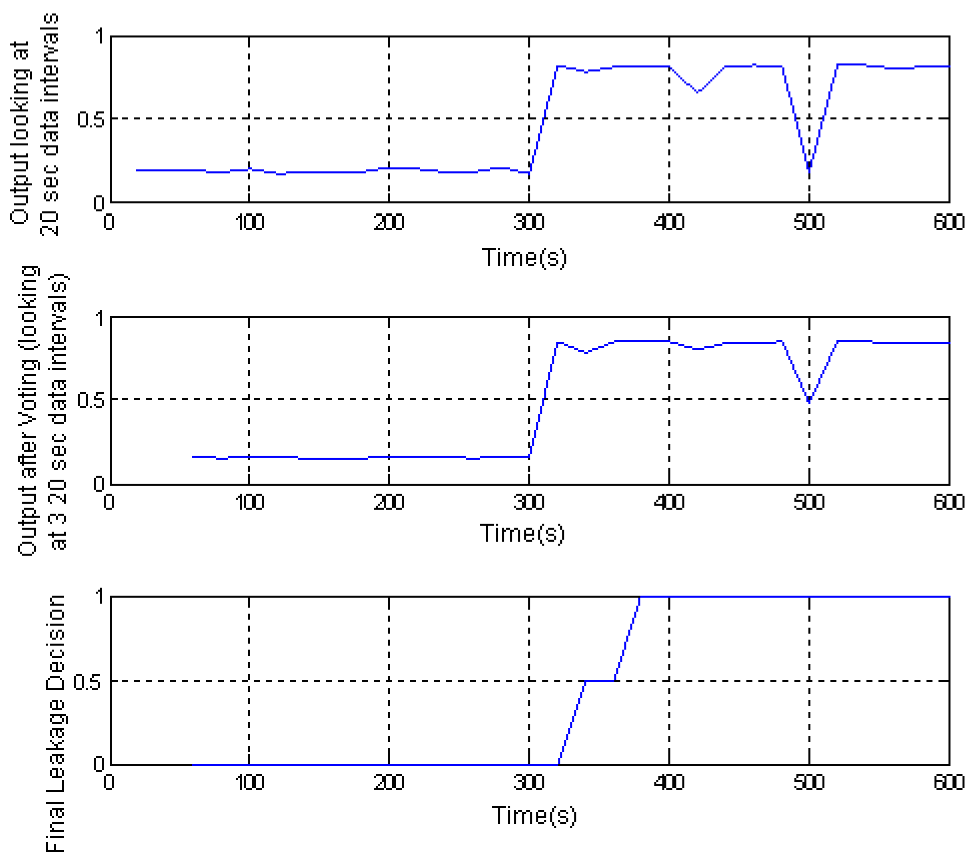


Figure 5.5 Result of applying Method I using 20 second time interval on data sample no. 12

When this result is compared to the original leakage which is Figure 5.1 it can be concluded that this algorithm successfully detects the leakage in the hydraulic cylinder.

The Figure 5.6 again shows Method I applied on the same data sample but using a 60 second data interval. Here too the 1st plot represents the output i.e. the leakage decision after level 2 described in the previous chapter. The 2nd plot represents the output i.e. leakage decision after level 3 again described in the previous chapter. The 3rd plot is conversion of the final decision into a 3 level digital decision. Recall that in this case the algorithm for Method I remains the same only the data interval changes from 20 seconds to 60 seconds.

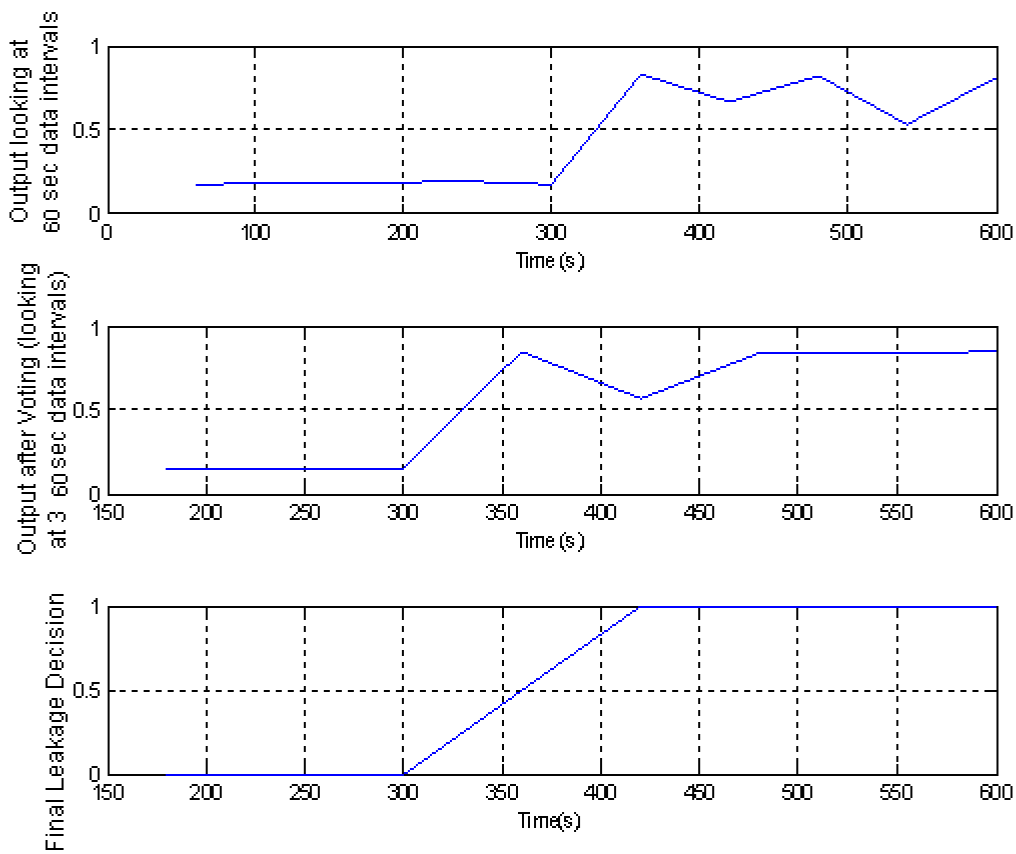


Figure 5.6 Result of applying Method I using 60 second data interval on data sample no. 12

The flowchart for Method II algorithm (Figure 5.7) is quite same as Method I with a few exceptions such as 'Ratio of Spikes' have been used for analysis instead of 'Number of Spikes'. Here again the input to the algorithm is the second level wavelet transform (n, t_n, d_{2n}). The 'Number of spikes' are calculated at the end of each iteration by the subroutine 'Inner Loop'.

These are then converted to ratio in the main algorithm. Here too the data interval can be chosen to be y=20 or y=60 seconds.

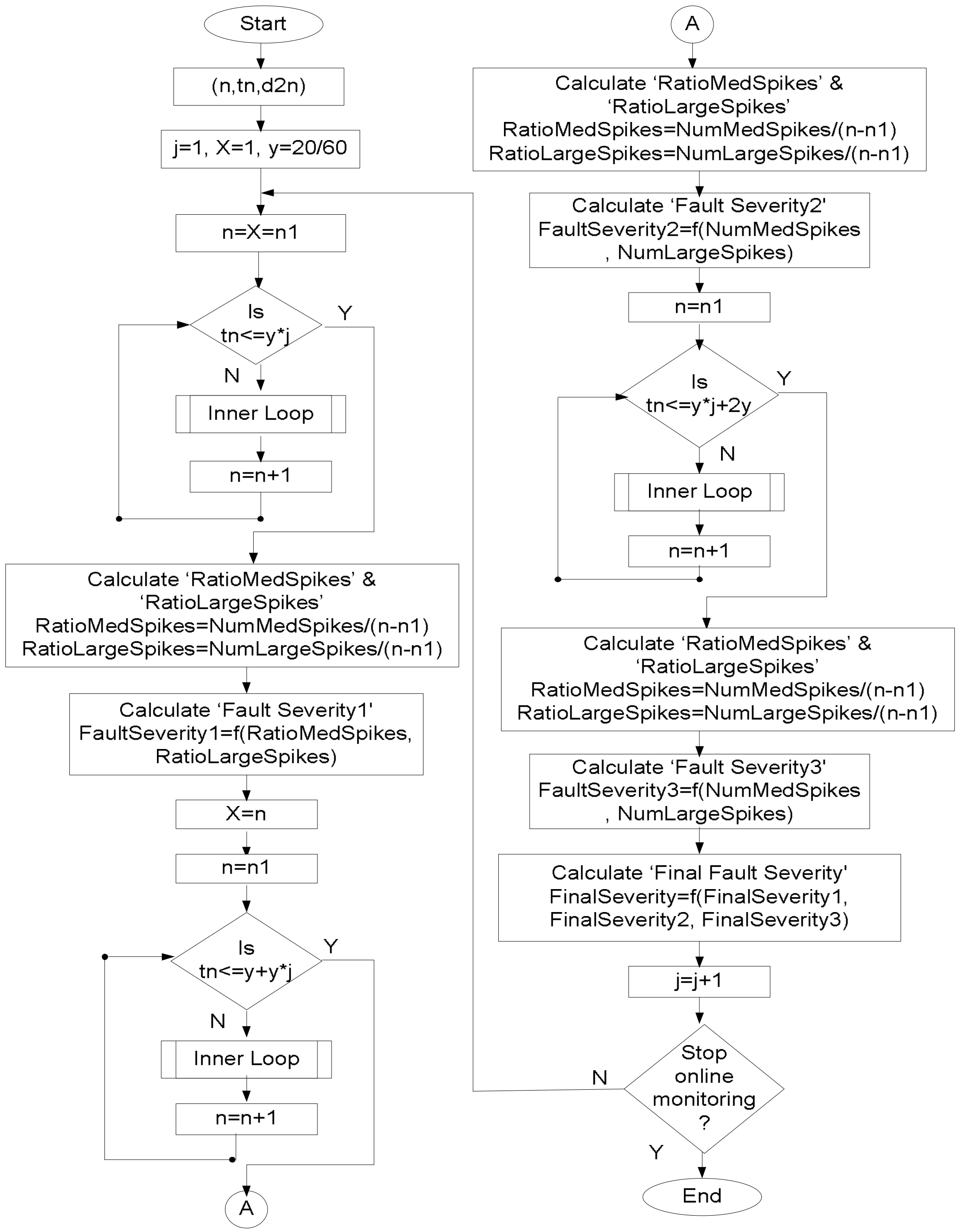


Figure 5.7 Method II Algorithm Flowchart for 20/60 second data interval

Figure 5.8 shows the result after applying Method II on the same data sample (data sample no. 12) which uses “ratio of spikes” in 20 second data interval as input.

In case of Method II, the plot for leakage decision after level 2 does not give a smooth curve, but various zigzag lines. This can be explained with an example. Let us consider the decision regarding leakage made at 40th second. During 1st iteration in Method II, the ratio of spikes in interval 0-20 sec, 0-40 sec and 0-60 sec is considered to make a final decision. Now, during 2nd iteration, the ratio of spikes in the interval from 20-40 sec and 20-60 sec and 20-80 sec is considered. As the data intervals differ (0-40 in 1st iteration and 20-40 in 2nd iteration) so, does the ratio of spikes and ultimately the decision made at 40th second during both the iterations. This leads to the zigzag output rather than a smooth curve after level 2 (shown in plot 1 in figure 5.8).

Note that in case of Method I, the 1st iteration would consider a data interval from 0-20 sec, 20-40 sec and 40-60 sec and 2nd iteration would consider a data interval from 20-40 sec, 40-60 sec and 60-80 sec. That means during both the iterations, the decision made at 40th second is by looking at 20-40 data interval. As the interval is same, the number of spikes is also the same and so is the leakage decision. Hence, a smooth curve is obtained for Method I as opposed to Method II.

The level 3 decision is made using the fault decisions made by three data intervals as inputs to make a final decision regarding fault (shown in plot 2 in figure 5.8). The final decision in plot2 is converted into a 3 level digital decision (shown in plot 3 in figure 5.8).

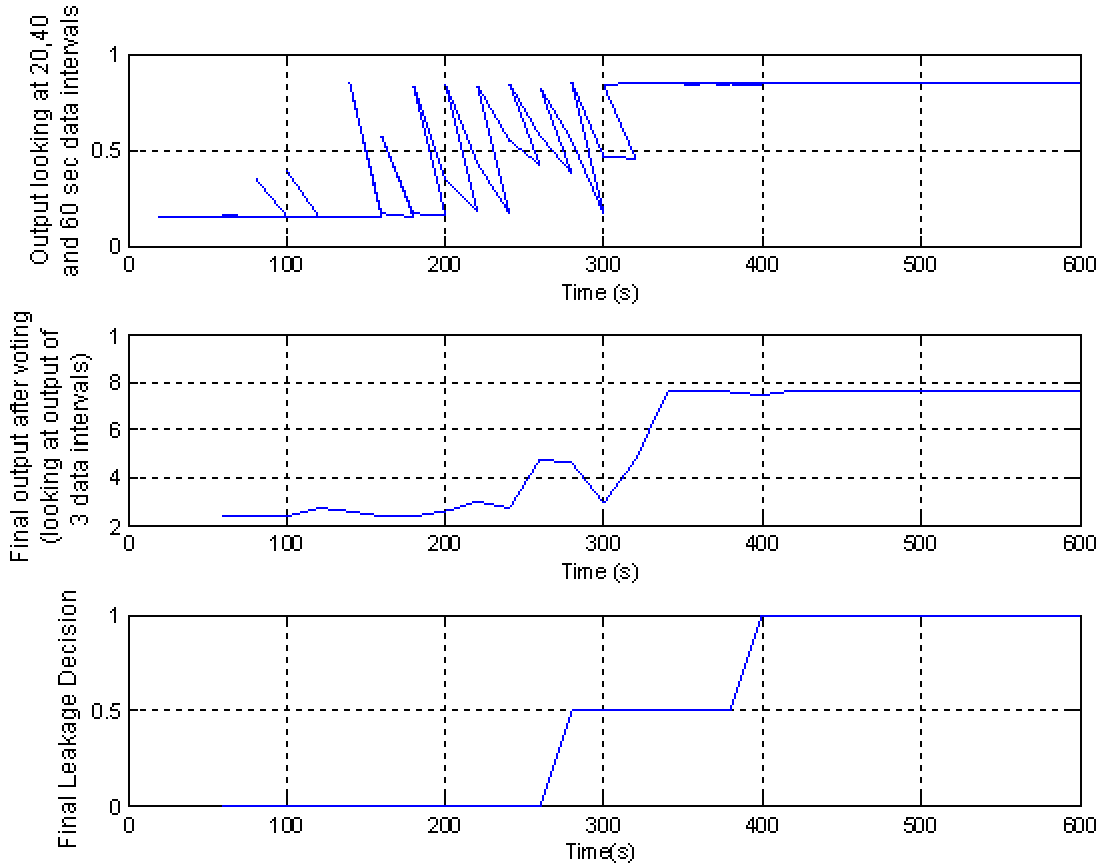


Figure 5.8 Result of applying method II using 20 second data interval on data sample no. 12

Figure 5.9 again shows. Method II applied for the same data sample (data sample no. 12) but for a 60 second data interval. Here too the 1st plot and the 2nd plot represents the output i.e. the leakage decision after level 2 and level 3 respectively described in the previous chapter. The 3rd plot is conversion of the final decision into a 3 level digital decision.

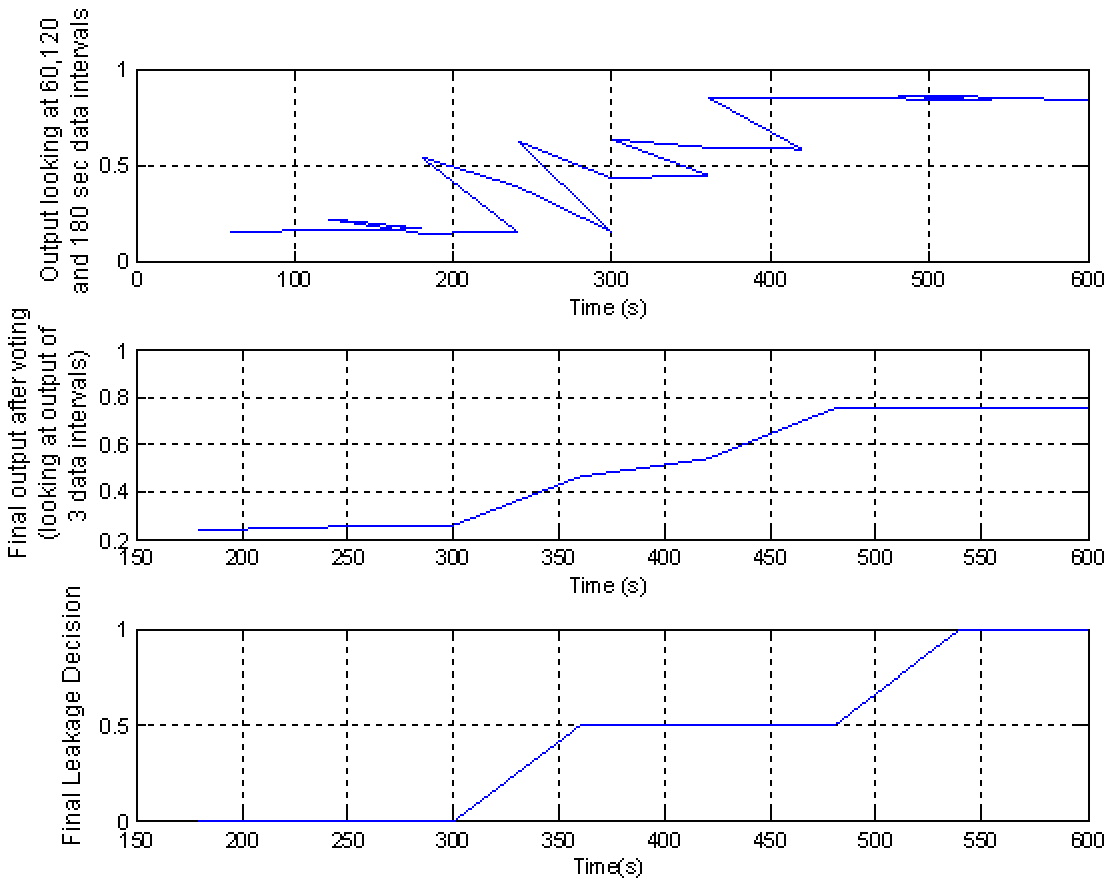
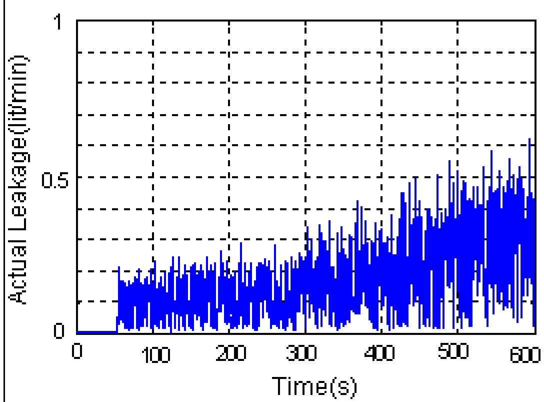
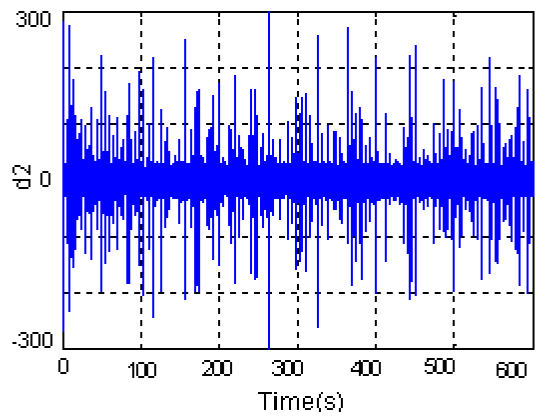
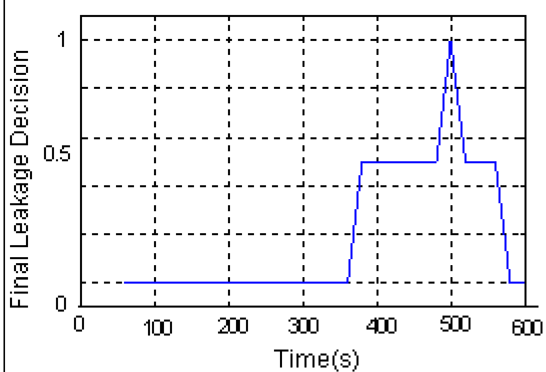
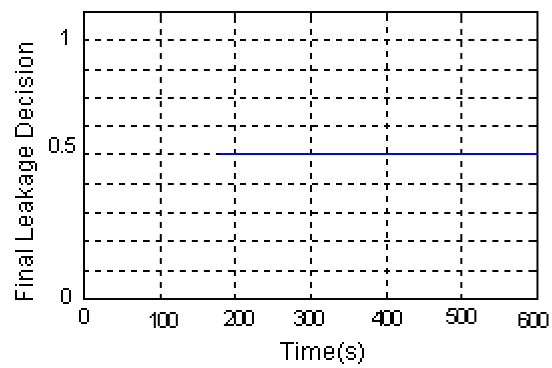
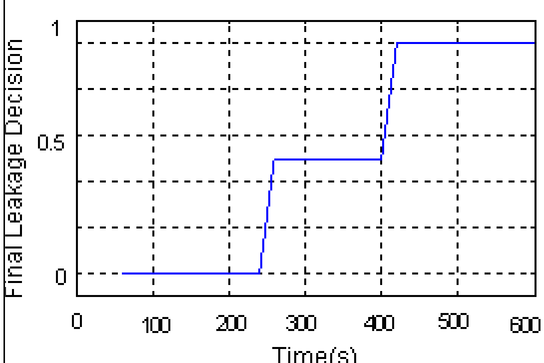
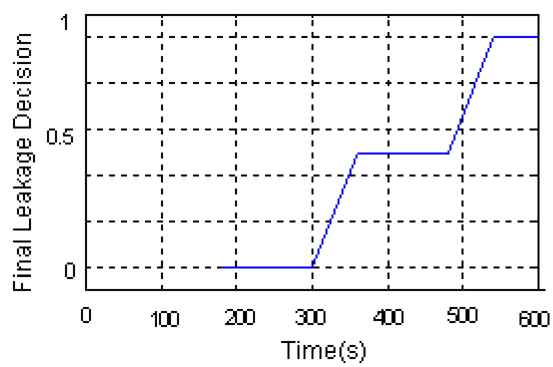


Figure 5.9 Result of applying method II using 60 second data interval on data sample no. 12

In the following data sample (sample no. 1, Table IV) it can be noticed that leakage introduced is very low. It starts from 0.2lit/min (about 0.1 lit/min on average) and continues to increase linearly till it reaches around 0.5lit/min (around 0.22 lit/min on average) at 600th second. Note that there is not a significant change seen in the number of spikes and their magnitude in the second level wavelet transform as there is gradual increase in the leakage. The results after applying both the methods for 20 seconds and 60 seconds time interval respectively can also be seen.

Table IV Results Data Sample I

Original Leakage 	2^{nd} Level Wavelet Transform 
Method I: 20 seconds 	Method I: 60 seconds 
Method II: 20 seconds 	Method II: 60 seconds 

From the above table it can be concluded that Method I is not capable of detecting leakages that are very low (typically less than 0.2 lit/min on average) whereas Method II is able to successfully detect the leakage. This is because Method I uses ‘Number of spikes’ as input for decision in contrast to ‘Ratio of spikes’ which is used in Method II. By considering ratios the approach becomes more normalized and hence more robust. Also, instead of looking at three 20 second data intervals (Method I), if looked at 20, 40 and 60 second data intervals (Method II). That means more data is taken into consideration which leads to a better result.

In the following data sample (sample no. 10, Table V) it can be seen that the leakage introduced is around 1lit/min (0.5 lit/min on average) starting from 300th second. The second level wavelet transform shows a significant change in the number of spikes and their magnitude after leakage was introduced i.e. approximately after 300th second. The results after applying both the methods for 20 seconds and 60 seconds time interval respectively can also be seen

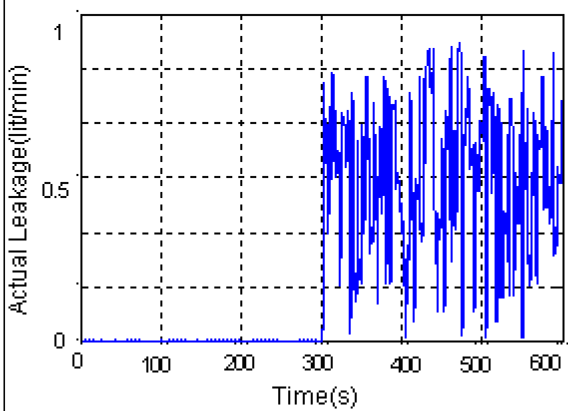
It can be concluded from table V that both the methods are able to successfully detect an average leakage of 0.5 lit/min (which is well above 0.2 lit/min). It is also noticed, that the leakage detected using 20 second data interval is earlier compared to 60 second data interval. This is because a leakage decision is obtained every 20 seconds using 20 second data interval and every 60 seconds using 60 second data interval. The earlier the leakage is detected the better which means that leakage detection using 20 second data interval gives better results.

In the data sample (Sample no. 4, Table VI) it can be seen that the leakage introduced was around 1lit/min (about 0.5 lit/min on average) starting from 70th second and it increases linearly to reach 1.8 lit/min(about 0.9 lit/min average) at 600th second. The second level wavelet transform shows a significant change in the number of spikes and their magnitude after leakage was introduced i.e. approximately after 70th second. The results after applying both the methods for 20 seconds and 60 seconds time interval respectively can also be seen.

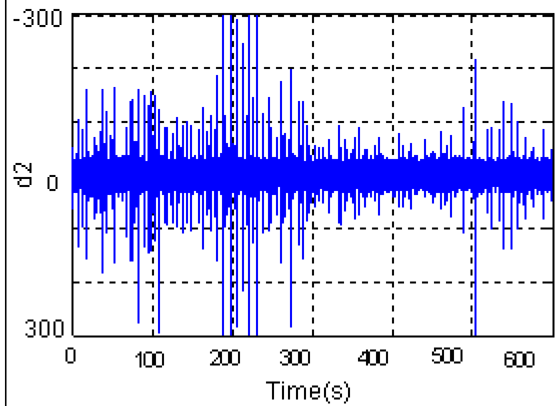
It can be concluded from table VI that both the methods are able to successfully detect the leakage which is considerably high. It is also noticed, that using 20 second data interval the leakage is detected earlier compared to 60 second data interval.

Table V Results Data Sample 10

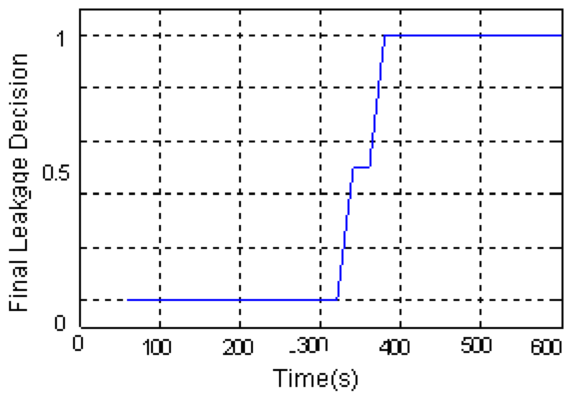
Data Sample 10
Original Leakage



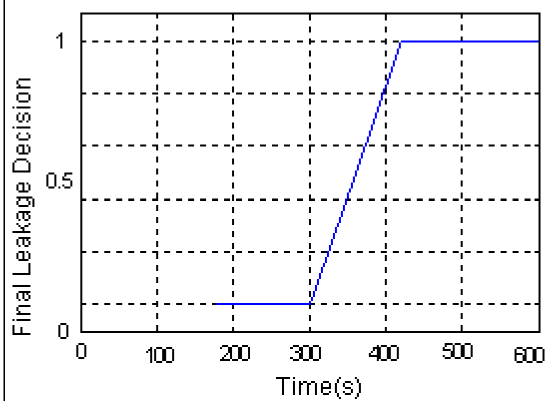
2nd Level Wavelet Transform



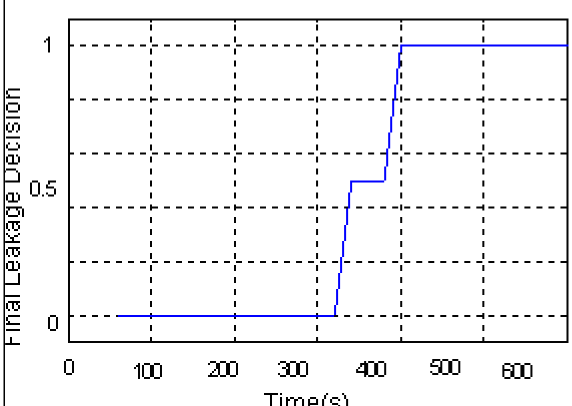
Method I: 20 seconds



Method I: 60 seconds



Method II: 20 seconds



Method II: 60 seconds

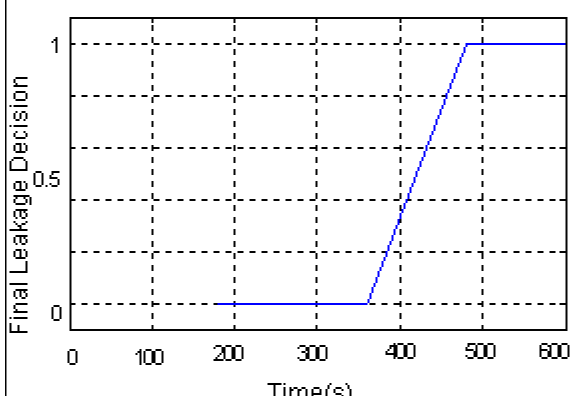
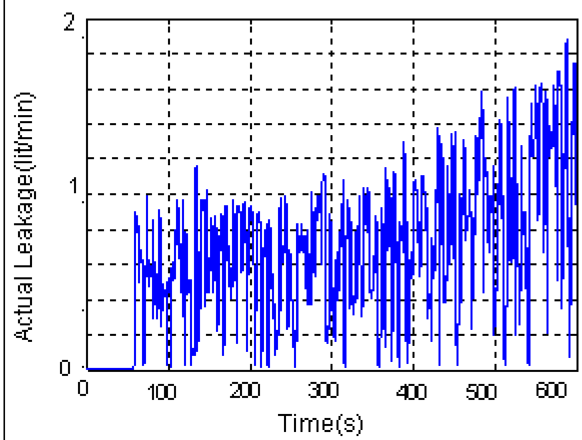
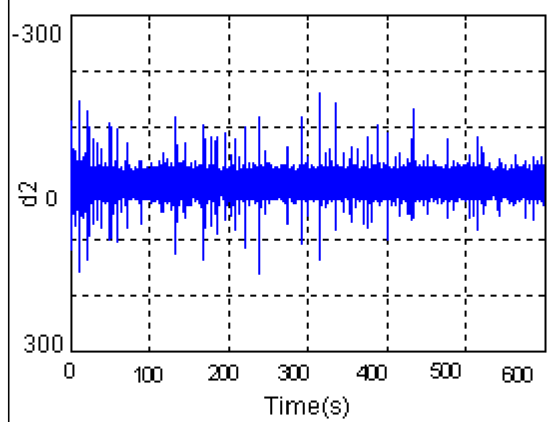


Table VI Results Data Sample No. 4

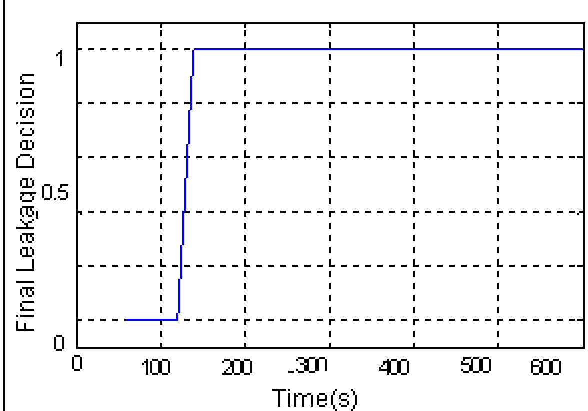
Data Sample 4
Original Leakage



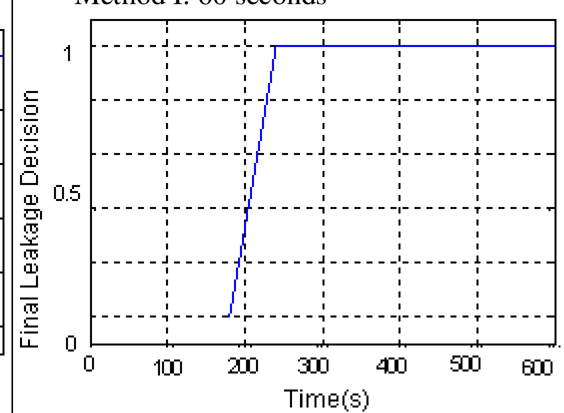
2nd Level Wavelet Transform



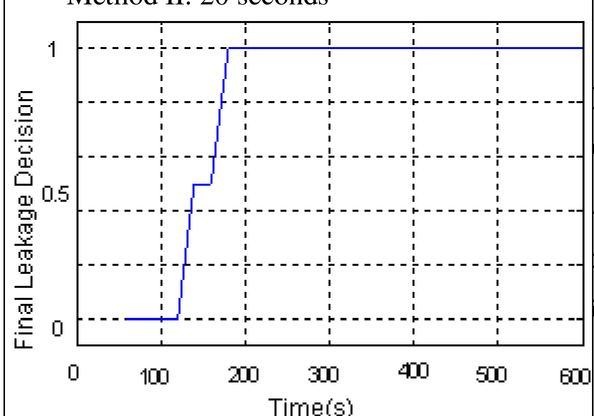
Method I: 20 seconds



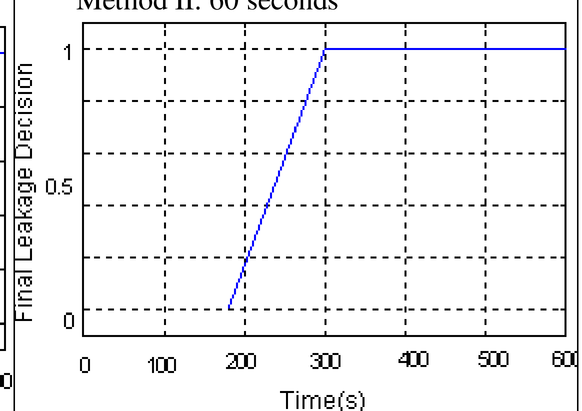
Method I: 60 seconds



Method II: 20 seconds



Method II: 60 seconds



Total 16 data samples were taken into consideration and the following table shows that

Table VII Summary of Results

Sample	Method I (20 sec)	Method I (60 sec)	Method II (20 sec)	Method II (60 sec)
1	Not Acceptable(0)	Not Acceptable(0)	Good(1)	Good(1)
2	Not Acceptable(0)	Not Acceptable(0)	Good(1)	Good(1)
3	Not Acceptable(0)	Not Acceptable(0)	Good(1)	Good(1)
4	Good(1)	Good(1)	Good(1)	Good(1)
5	Not Acceptable(0)	Not Acceptable(0)	Acceptable(0.7)	Acceptable(0.7)
6	Good(1)	Acceptable(0.7)	Good(1)	Acceptable(0.7)
7	Acceptable(0.7)	Good(1)	Acceptable(0.7)	Good(1)
8	Acceptable(0.7)	Good(1)	Acceptable(0.7)	Acceptable(0.7)
9	Not Acceptable(0)	Good(1)	Acceptable(0.7)	Good(1)
10	Good(1)	Good(1)	Good(1)	Good(1)
11	Acceptable(0.7)	Good(1)	Acceptable(0.7)	Acceptable(0.7)
12	Good(1)	Good(1)	Good(1)	Good(1)
13	Good(1)	Good(1)	Good(1)	Good(1)
14	Good(1)	Good(1)	Good(1)	Good(1)
15	Good(1)	Good(1)	Acceptable(0.7)	Acceptable(0.7)
16	Acceptable(0.7)	Good(1)	Acceptable(0.7)	Good(1)
Ratings	9.8	11.7	13.9	14.5
Percentage	61.25%	73.125%	86.88%	90.63%

All four methods are compared using the rating technique. Considering a rating of 0 for a ‘Not Acceptable’ results, 0.7 for ‘Acceptable’ results and 1 for all ‘Good’ results, the total of ratings for all four methods is calculated in the table above. It is found that Method II which takes into account data interval of 60 seconds has the highest ranking with 14.5 (or a success percentage of a little over 90%). Next to that the 2nd highest ranking of 13.9 (or a success percentage of 87% which is quite good) goes again to Method II which is the ratio method for 20 second data interval. Method I with 60 seconds has the rating of 11.7 (or a success percentage of 73% which is quite low) and Method I with 20 seconds has the lowest rating of 9.8 (or a lowest success percentage of 61%).

5.2 Conclusion

From the above results it can be concluded that using method II (ratio approach) and taking 60 second data interval as input renders highest success. The reason for this is mainly because by taking ratio, the data is more normalized. Also, a 60 second data interval yields more data for analysis. With more data for analysis the accuracy of diagnosis always increases. With a highest rate of success of 90% this approach is the most desirable.

Method II with 20 second data interval also renders satisfactory success rate of 87% as contrast to method I. The choice between method I and method II depends on whether accuracy is more important for the application or time. The method which uses 60 second data as time interval renders a decision after every 60 seconds. On the other hand, the method using 20 second data as time interval renders a decision after every 20 seconds and hence faster. At the same time there is a tradeoff that the accuracy of results might be lesser as compared to 60 second data time interval.

Method I which uses Number of Spikes as input leaves us with very low success rate. Hence, the use of this method is not recommended at all unless there is a particular reason to do so. Here too the same limitations and advantages apply for choosing different time intervals.

The major results contributed by this study is listed below

1) Addition of knowledge based methodology which is fuzzy logic fault detector to already developed model based approach.
2) Development of algorithm which detects internal leakage based on changes observed in the second level wavelet transform.
3) Applying same algorithm in four ways which differ a little from each other.
4) Testing of all four methods on 16 sets of data
5) Making inferences as regards to which method is the best
6) Finally demonstration of novel application of fuzzy logic as a decision making tool in the area of fault detection.

Overall, fuzzy logic is a good tool for solving problems similar to one described here. It renders good results for vague data and knowledge base available. Whenever the domain area of problem is specific (not to broad), with some amount of trial and error fuzzy logic is capable of acceptable fault diagnosis. With simplified & reduced development cycle and ease of implementation, fuzzy logic is an efficient way to solve real world problems.

The concept of the excluded middle, where every logical preposition has to either be completely true or false, does not seem to fulfill expectations of today's technical and logic dependent world. Although behavior of many applications in today's world still relies on the True/False conditions a need is always felt to adjust classical logic to accept the concept of something being neither completely true nor false but a value in-between

Generally the fuzzy logic is recommended for the implementation of a process, where a simple mathematical model cannot be obtained. It is not recommended to employ fuzzy logic into systems where a simple and adequate mathematical model already exists or where the conventional control theory yields a satisfying result. Fuzzy logic seems to be a general case for the classical logic and as such it does not present any better solutions for problems that might be easily solved using the "crisp" sets.

5.3 Future Work:

The first part of this research is limited to 'detecting' faults. If further knowledge and data becomes available this study could be further extended to 'isolation' of fault. The fault could be isolated based on the 'component' of hydraulic system which is causing fault or the 'reason' behind fault (which might be change in one/many of the system parameter/s like bulk modulus or viscosity).

The algorithm developed in the second part of this research is tested on 16 datasets. This algorithm could be vigorously tested on many more datasets to verify the result. It may also be tested on different hydraulic systems as a future study.

Along with Fuzzy Logic other soft computing technologies like Artificial Neural Network and Genetic Algorithms can be incorporated separately or in combination with existing methodology to achieve further optimization.

REFERENCES

[1] Khan H., Seraphin C. Abou and Sepehri N., “Nonlinear Observer-Based Fault detection Technique for Electro hydraulic Servo positioning Systems”, Elsevier Science, vol. 15, pp. 1037-1059, 2005.

[2] Goharrizi, A.Y. and Sepehri, N. , “A Wavelet-Based Approach to Internal Seal Damage Diagnosis in Hydraulic Actuators”, Industrial Electronics, IEEE Transactions, vol. 57, issue 5 , pp. 1755 – 1763, May 2010

[3] Dings Yu, D.N.Shields, J.L.Mahtani, “A non-linear fault detection method for a hydraulic system”, Control 1994. International Conference, vol. 2, pp. 1318- 1322, 1994.

[4] G. J. Preston, D. N. Shields, and S. Daley, “Application of a robust nonlinear fault detection observer to a hydraulic system”, Proceedings of UKACC International Conference Control, U.K., pp. 1484–1489, 1996.

[5] Dingli Yu, D. N. Shields and S. Daley, “Fault detection for bilinear systems with application to a hydraulic system”, Control Applications, 1994., Proceedings of the Third IEEE Conference , vol. 2, pp. 1379- 1384, 1994.

[6] Phanindra Garimella and Bin Yao, “Model Based Fault Detection of an Electro-Hydraulic Cylinder”, American Control Conference June 8-10, 2005. Portland, OR, USA, pp. 484-489, 2005

[7] Galbraith, L. Al-Najjar, M. Babu, A.J.G., “Expert systems in engineering”, Aerospace and Electronic Systems Magazine, IEEE, vol. 3, issue 2, pp. 12-14, 1988.

[8] Davor LinariC, Vladimir Koroman, “Fault diagnosis of a hydraulic actuator using neural network”, Industrial Technology, 2003 IEEE International Conference, vol. 1, pp. 108-111, 2003.

[9] Florin Ionescu, “Intelligent Solutions for Mechatronics”, Knowledge-Based Intelligent Information Engineering Systems, 1999. Third International Conference, pp. 153-157, 1999.

[10] Yifeng Zhou, Juergen Hahn and M. Sam Mannan, “Fault detection and classification in chemical processes based on neural networks with feature extraction”, *ISA Transaction*, vol. 42, pp. 651--664, 2003.

[11] Chang Feng, Tingdi Zhao, Nuo Zhao and Shuyue Yin, “Research on Obtaining Dynamic, Robust Fault Diagnosis Rules”, Annual Reliability and Maintainability Symposium, pp. 64-68, 2009.

[12] Raghunathan Rengaswamy and Venkat Venkatasubramanian, “A fast training neural network and its updation for incipient fault detection and diagnosis”, Computers & Chemical Engineering, vol 24, issues 2-7, pp. 431-437, 2000.

[13] Burak Özyurtand Abraham Kandel, “A hybrid hierarchical neural network-fuzzy expert system approach to chemical process fault diagnosis”, Fuzzy Sets and Systems vol. 83, issue 1, pp. 11-25, 1996.

[14] R. P. Leger, Wm. J. Garland and W. F. S. Poehlman, “Fault detection and diagnosis using statistical control charts and artificial neural networks”, Artificial Intelligence in Engineering, vol. 12, issues 1-2, pp. 35-47, 1998.

[15] Enrique E. Tarifa, Demetrio Humana, Samuel Franco, Sergio L. Martínez, Alvaro F. Núñez and Nicolás J. Scenna, “Fault diagnosis for a MSF using neural networks”, Desalination, vol. 152, issues 1-3, pp. 215-222, 2003

[16] K.P. Ferentinos, L.D. Albright a, B. Selman, “Neural network-based detection of mechanical, sensor and biological faults in deep-trough hydroponics”, Computers and Electronics in Agriculture vol. 40, issues 1-3, pp. 65-85, 2003.

[17] Q. Chen, Y.W. Chan b and K. Worden, “Structural fault diagnosis and isolation using neural networks based on response-only data”, Computers & Structures, vol. 81, issues 22-23, pp. 2165-2172, 2003.

[18] Juan Anzurez-Marin, Bernardino Castillo Toledo, Elisa Espinosa-Juárez, “Fault Detection and Isolation Problem: Sliding Mode Fuzzy Observers and Linear Integer Programming”, Electronics Robotics and Automotive Mechanics Conference, pp. 319-324, 2008.

[19] George J. Klir, Ute St. Clair and Bo Yuan, “Fuzzy Set Theory: Foundations and Applications”, New York, Prentice Hall; Facsimile edition, April 27, 1997

[20] Zadeh L.,”Fuzzy Sets” , Information and Control, vol: 8, pp:338-353

[21] Bart Kosko, “Fuzzy Systems as Universal approximators”, IEEE Transactions on Computers, vol 43, issue 11, pp: 1329 - 1333 , 1994

[22] Li-Xin Wang, “A course in fuzzy systems”, New Jersey, Prentice Hall; Facsimile edition, August 30, 1996.

[23] Jon Yen and Reza Langari, “Fuzzy Logic: Intelligence, Control, and Information”, Prentice Hall, United States edition, December 3, 1998.