A HYBRID SYSTEM FOR FAULT DETECTION AND SENSOR FUSION BASED ON FUZZY CLUSTERING AND ARTIFICIAL IMMUNE SYSTEMS

A Dissertation

by

MOHAMMAD ABDEL KAREEM RASHEED JARADAT

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2005

Major Subject: Mechanical Engineering

A HYBRID SYSTEM FOR FAULT DETECTION AND SENSOR FUSION BASED ON FUZZY CLUSTERING AND ARTIFICIAL IMMUNE SYSTEMS

A Dissertation

by

MOHAMMAD ABDEL KAREEM RASHEED JARADAT

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Reza Langari
Committee Members,	Alan Palazzolo
	Won-jong Kim
	Hamid Toliyat
Head of Department,	Dennis O'Neal

December 2005

Major Subject: Mechanical Engineering

ABSTRACT

A Hybrid System for Fault Detection and Sensor Fusion Based on Fuzzy Clustering and Artificial Immune Systems. (December 2005)

Mohammad Abdel Kareem Rasheed Jaradat, B.S., Jordan University of Science and Technology; M.S., Texas A&M University

Chair of Advisory Committee: Dr. Reza Langari

In this study, an efficient new hybrid approach for multiple sensors data fusion and fault detection is presented, addressing the problem with possible multiple faults, which is based on conventional fuzzy soft clustering and artificial immune system (AIS).

The proposed hybrid system approach consists of three main phases. In the first phase signal separation is performed using the Fuzzy C-Means (FCM) algorithm. Subsequently a single (fused) signal based on the information provided from the sensor signals is generated by the fusion engine. The information provided from the previous two phases is used for fault detection in the third phase based on the Artificial Immune System (AIS) negative selection mechanism.

The simulations and experiments for multiple sensor systems have confirmed the strength of the new approach for online fusing and fault detection. The hybrid system gives a fault tolerance by handling different problems such as noisy sensor signals and multiple faulty sensors. This makes the new hybrid approach attractive for solving such fusion problems and fault detection during real time operations.

This hybrid system is extended for early fault detection in complex mechanical systems based on a set of extracted features; these features characterize the collected sensors data. The hybrid system is able to detect the onset of fault conditions which can lead to critical damage or failure. This early detection of failure signs can provide more effective information for any maintenance actions or corrective procedure decisions.

To my dearest lovely parents, wife, son, and family ……

Mohammad

ACKNOWLEDGMENTS

I would like sincerely to thank my advisor, Dr. Reza Langari, who provided my always with his support, guidance and encouragement during my graduate study. I gained a lot from his thorough knowledge and experience throughout my studying period.

I would like to thank my committee members, Dr. Alan Palazzolo, Dr. Won-jong Kim and Dr. Hamid Toliyat, for serving on my advisory committee, and for their helpful ideas, suggestions, and support.

I would like to extend my sincere thanks to my dearest parents, lovely wife and son, my brother and his family, my sisters and their families, and my parents in law and their family for their endless support, patience, encouragement, deep understanding and being there me whenever I needed them with love and support.

Finally I would like to thank all my friends and colleagues for their help and support during my study period at college station.

Thanks for you all, for the encouragement, support, and love which made me able to accomplish my graduate studies.

TABLE OF CONTENTS

CHAPTER Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

I. INTRODUCTION

In the last two decades, a notable amount of research has addressed the problem of using multiple sensors to achieve better performance in diagnostics as well as feedback control systems.

The process of combining the provided information from multiple sensors is called sensor fusion, and can overcome a number of problems ranging from noise to incipient sensor failure. Even in the absence of these issues, one can increase the system's accuracy and the reliability using sensor fusion [1].

Where the main interest in multiple sensor systems springs from the common realization of fundamental limitations for the information provided by using a single sensor in the system. Whyte and others illustrated these limitations in the following points [2, 3]:

- Using a single sensor can provide only partial information about the operating conditions and environment.
- Using the single sensor systems, cause the resulted systems observations uncertain and occasionally it could be incorrect.

This dissertation follows the style and format of *IEEE Transactions on Systems, Man, and Cybernetics*.

- Using multiple sensors allows the incorporation of various type of information, where multiple tasks can be achieved by the same system.
- Using single sensor systems, the system will completely fail with possible sensor operational faults, which can lead to critical situations or consequences on that system. For instance in feedback control systems, the failure of the sensor which provides the output measured values will lead to a system failure.

Due to these reasons, several conventional approaches have been developed for multiple sensors fusion as well as fault detection. The most important used approaches can be categorized as follows:

- Kalman filtering, the weighted average, Bayesian estimators and nonlinear fusion based approaches [4-15].
- Fuzzy logic, neural network and soft computing based approaches [1,16-23].
- Hybrid systems approaches formed by a combination of soft computing and Kalman filtering, or Bayesian estimators from the first category [24,25].

For instance the Kalman filter based approaches from the first category, are developed for multiple sensor fusion and fault detection based on the statistical characteristics of

the measurement model to provide the fusible estimate for the given inputs and residual outputs. Taking into consideration the ability to provide a linear model for the addressed system and the measurement of the noise, these facts arise from the nature of the Kalman filter as an optimal linear estimator, which works in recursive nature to provide the future state based on the system transition matrix [4].

The Kalman filter approach can be extended to the Extended Kalman Filtering approach, when some of the assumptions are used to provide an approximate linear model for the system present potential limitations or numerical instability [4].

From the second category several neural network based approaches were developed in different layer structures to achieve the addressed objectives. These layers consist of simple processing neurons which are fully interconnected with each other providing the network output, as shown in figure 1.1.

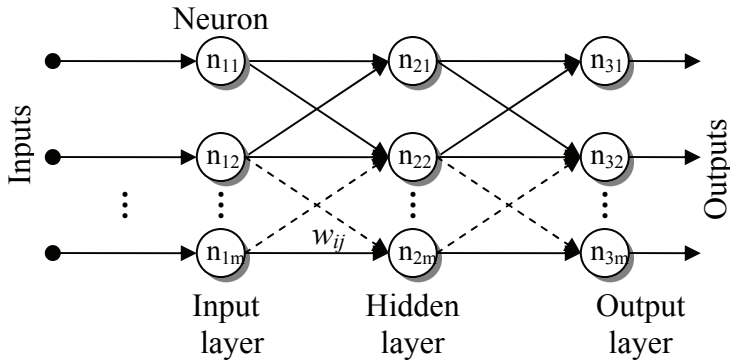


Fig. 1.1. Structure of three layer neural network.

The neuron output is defined as the weighted sum of the entire inputs applied to the neuron activation function. Training or learning phase is required to select the appropriate weights for the network, which can be achieved through different learning approaches based on a provided training set of known input/output data for the system. After that the network will be ready for fault detection and fusion purposes [26, 27].

Hybrid approaches are developed by considering different combinations between the first and the second approaches categories. The main idea is to maximize the benefits from combining the different approach advantages for sensors fusion and fault detection. The hybrid approach presented by Ambrosio and Mort [24] for example, is making use of the fuzzy logic inference system capabilities to develop a Fuzzy-adaptive Kalman filter, which shows a good performance based on the fuzzy logic ability to deal with imprecise information based on the fuzzy membership representation of the variables.

As we discussed, the majority of these methods use linear estimation models which require a previous knowledge of signal statistics, while others require knowledge of the behavior of the system to generate the governing rules of the fusion approach. Other approaches could have problems when it is necessary to add new sensors to the system. The presence of more than one faulty signal is an essential limitation of some of these approaches performances.

This study focuses to provide new approach for both multiple sensor fusion and fault detection. A preliminary introduction is presented in this section, in the following section, the problem statement and methodology outline is covered, while the addressed multiple sensors problem architecture is presented in section three.

II. PROBLEM STATEMENT AND METHODOLOGY OUTLINE

Our objective in this study is to presents a new approach for multiple sensor data fusion and fault detection. In this approach a new hybrid system is developed, which consists of the following three main phases:

- In the first phase, a signal separation is performed using the Fuzzy C-Means (FCM) algorithm.
- Subsequently, a single (fused) signal based on the information provided from the sensor signals is generated by the fusion engine.
- The information provided from pervious two phases are used for fault detection in the third phase based on Artificial Immune System (AIS*)* negative selection mechanism.

The above approach is detailed in the following chapters with reference to the multiple sensors problem architecture which will be discussed shortly. In the next chapter the approach development with required backgrounds is discussed. The hybrid system performance evaluations are subsequently demonstrated through the simulated results as well as the experimental results. In this chapter the hybrid system performance is compared to and evaluated to other approaches. In the next chapter an early fault detection system is developed based on the hybrid system approach, the experimental

results from applying the hybrid system is discussed too. Finally, the last chapter covers the concluded remarks from this study and the future work.

III. SENSOR FUSION PROBLEM ARCHITECTURE

The general multiple-sensor scheme discussed in this study is shown in Figure 1.2. As shown in the figure, the system consists of *m* input signals (from *m* sensors) and the objective is to achieve one fused output S_f based on these inputs.

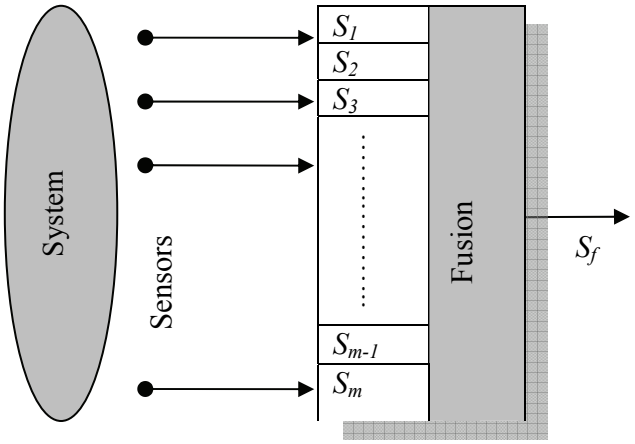


Fig. 1.2. General fusion block diagram.

If the input sensors try to measure the same type of signal, i.e. light, sound, temperature, position or velocity, the value measured by all the sensors at any time instant *t*, should be ideally the same as shown in figure 1.3. Note that in general, a tolerance value for the measurements will be used to account for inherent inaccuracy in the sensor measurements, leading to an acceptable region for the measured values, as shown in figure 1.4.

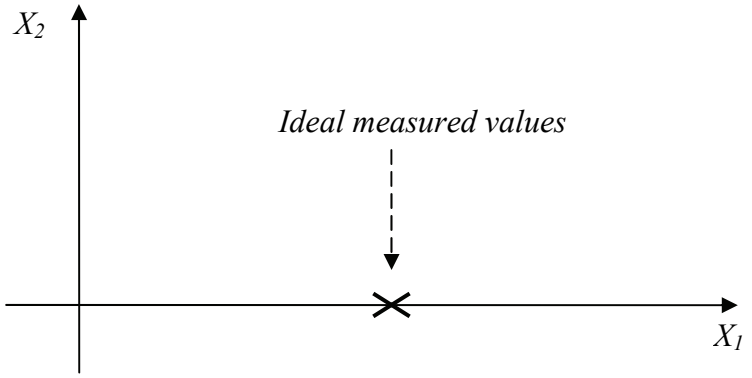


Fig. 1.3. Ideal measured values by the sensors.

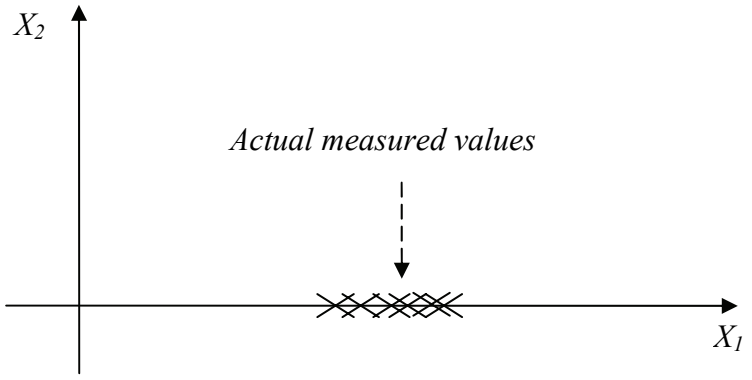


Fig. 1.4. Actual measured values by the sensors.

Some of these measured values could drift away from this acceptable region during real time operation, due to the fault influence during the sensors operation such as parameter changes or changes in the sensors operational characteristics [28], the drifted values will be distributed in some manner relative to the acceptable measurements region, as shown in Figure 1.5, where the drifted measurements can be noticed in the figure. One possibility is that the measured value drifted to the right side of the acceptable region, while the other possibility is that the measured values drifted to the left side of the acceptable region; it is also possible for the two cases to occur simultaneously with some readings drifting to the left and others to the right of the acceptable region.

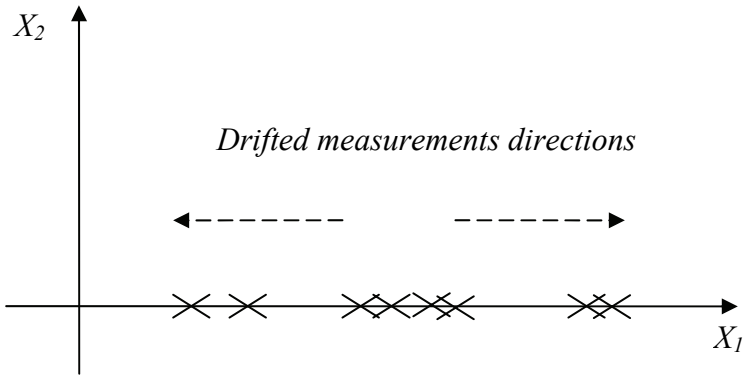


Fig. 1.5. The drifted measurement directions.

To perform any useful function using the measured readings (diagnostic or control) a single fused signal, S_f, output is required to represent the correct measured values, as in

equation 1.1 [29], and to eliminate the effects of the operational problems of the sensor as described above.

$$S_f = Fusion(S) \tag{1.1}$$

Where S_f is the fused output we are looking for through the implemented fusion approach, for a given set of the measured values by the multiple sensors denoted as $S = \{S_1, \ldots, S_m\} \in \mathbf{R}^{1x\ m}$, this fused output will be considered for fault detection process subsequently.

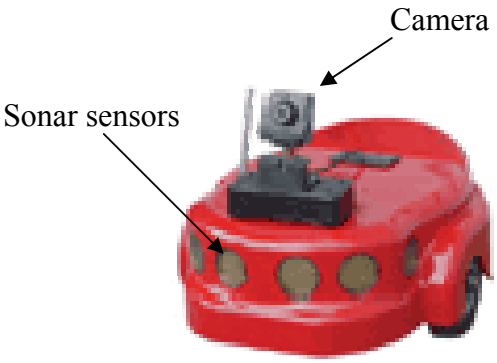
The used sensors in the multiple sensor system scheme shown in figure 1.2, could be of the same type or different types providing the same measured information. For instance, in some mixing manufacturing processes that are using the same sensors type, require adding various additive materials at specific temperatures to finalize the product. In such systems several thermocouples are positioned at different locations around the mixing tank to provide accurate temperature measurements for the mixed materials, where some of these additive processes are sensitive to the changes in the measured temperature [30].

The mobile robot platforms are another example for multiple sensor systems using several sensors mounted at several locations onboard the robot body to provide different type of information for obstacle avoidance, navigation, map building, and localization [31].

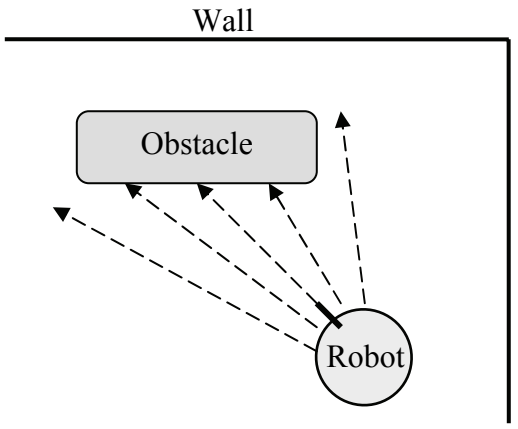
These sensors could be from different types, such as laser scanners, cameras, IR sensors, and sonar sensors, as shown in figure 1.6 part *a* [32]. In spite of these sensors

are heterogeneous, they provide the same information about the detected obstacles for example or the robot surrounding environment as shown in figure 1.6 *b*.

This general multiple-sensor scheme will be addressed in the following chapters based on the methodology outline as mentioned before in section II, for both multiple sensor fusion and fault detection.



(*a*)



(*b*)

Fig. 1.6. (*a*) Mobile robot platform with multiple sensors onboard, (*b*) the obstacle detected by the robot sensors.

CHAPTER II

HYBRID SYSTEM IMPLEMENTATION

I. INTRODUCTION

The major objective of this chapter is to introduce main components of the hybrid system and algorithm implementation based on the fuzzy clustering and the artificial immune system. Additionally, an introductory material and information required for the discussion progress through out the chapter sections is provided.

Firstly, an introductory discussion about the fuzzy set theory is presented in section two, the fuzzy clustering as a fusion mechanism is covered in section three, the fusion engine description will follow in section four, in the sequel the cluster selection and the algorithm summery is presented in section five and six. Finally, section seven will cover the hybrid fault detection system as well as a basic introductory about the artificial immune systems.

II. FUZZY SET AND MEMBERSHIP FUNCTION

The idea of fuzzy set theory was proposed by Lutfi A. Zadieh forty years ago [33] as the foundation for computing with words [34]. A fuzzy set theory is a generalization for the classical crisp set theory, this set is defined as a set of smooth boundary allowing the

variable in a set to have a partially membership degree to belongs to this set. This membership degree is expressed by a value between *0* and *1*[35,36].

Thus a fuzzy set is defined by a function that maps each element in the universe of discourse to its membership value in the set between *0* and *1*. This function is called the membership function, denoted by $\mu(x)$ [35, 36]. Figure 2.1 illustrates the fuzzy set membership function definition, while the fuzzy set is defined as follows for a fuzzy set *A*:

$$A = \{x \mid \mu_A(x) \longrightarrow [0,1], x \in X\} \tag{2.1}$$

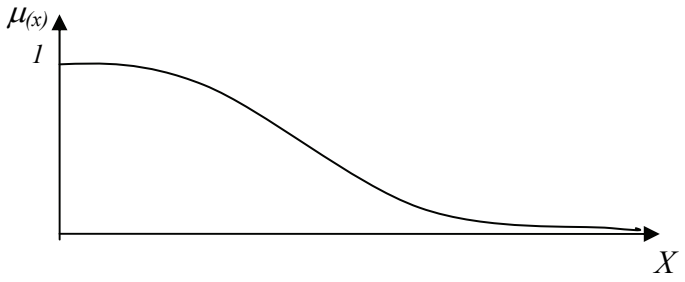


Fig. 2.1. Fuzzy set membership function.

III. FUZZY CLUSTERING AS A FUSION MECHANISM

A fuzzy (soft) partition for any given data set $X \in \mathbf{R}^{1x\ m}$, where $x_i \in X$, can be defined such as $P = \{C_1, C_2, \ldots, C_l\}$ of *X*, if and only if it satisfies the following conditions [35]:

1- $\forall x_i \in X \quad \forall C_j \in P, \quad 0 \le \mu_{C_j}(x_i) \le 1.$

2- $\forall x_i \in X \quad \exists C_j \in P \quad such \quad that \ , \mu_{C_j}(x_i) > 0.$

3- $\sum_{J} \mu_{C_j}(x_i) = 1, \quad \forall x_i \in X.$

Where $\mu_{ci}(x_i)$ shows the degree to which x_i belongs to a given partition C_j as follows :

$$\mu_{C_j}(x_i) \quad : X \longrightarrow [0,1] \text{ , where } 1 \le j \le l, \quad 1 \le i \le m.$$

The first condition in the definition above assures that any data point $x_i \in X$ can partially belong to multiple partitions, while the second condition assures the coverage of all the data points in the data set X by the resulted partitions P, finally the third condition assures the partition is consistent with the intuitive notion that a given x_i does not altogether belong to more than one whole set.

Now, let us assume that the available m sensor signals are noisy and subject to operation faults such as parameter changes or changes in the sensors operational characteristics [28]. A given input set of measurement $S = \{S_1 , ..., S_m\} \in \mathbf{R}^{1x\ m}$ will not contain the same measured values for any measurement at any time instant t, as follows:

$$S_i(t) \neq S_{i+1}(t) \ , for \quad \forall i, \quad i = 1,...,m. \tag{2.2}$$

The input vector S, could be separated into different partitions by using the fuzzy c-means (*FCM*) clustering algorithm, which is one of the most active and widely used algorithms in practical data analysis [35,37], as shown in figure 2.2. One of these partitions characterizes the right input values, while the other partitions characterize the

faulty readings or measurements. Given the measurements of vector *S*, the data shown in Figure 1.5 could be identified by using the fuzzy c-means clustering algorithm which tries to minimize the following objective function for the given vector *S* [35,38,39]:

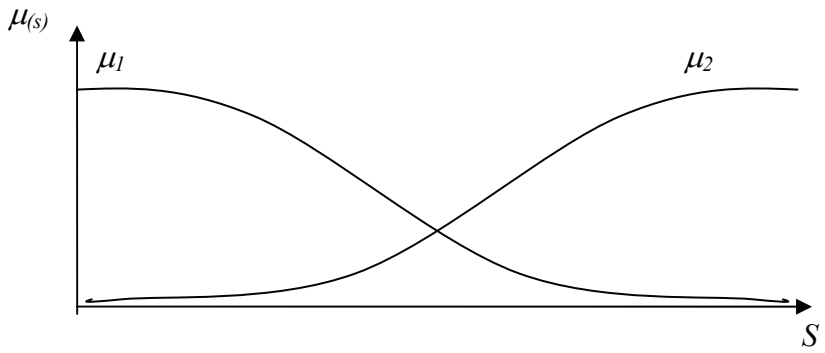


Fig. 2.2. Two fuzzy partitions.

$$J(\mu, v) = \sum_{i=1}^{k} \sum_{x_k \in X} (\mu_{C_i}(x_k))^p \left\| x_k - v_i \right\|^2 \tag{2.3}$$

Where p is a weighting value which represents the partial membership effects on the clustering result. V is the cluster center vector, the vector $X = [S^T\ Z^T] \in \mathbf{R}^{m \times 2}$ is the input vector where $Z \in \mathbf{R}^{1 \times m}$ is the zero vector, and C_i is the generated fuzzy partition for the input data, when $i = 1, 2, 3$.

The objective cost function J can be minimized be means of the partial derivative with respect to μ and v as follows [40-42]:

$$\frac{\partial J}{\partial \mu} = 0 \tag{2.4}$$

and

$$\frac{\partial J}{\partial v} = 0 \tag{2.5}$$

Where the constraints for the fuzzy partitioning are considered more by the following Lagrange function, where λ is the Lagrange multiplier [40-42]:

$$J(\mu, v) = \sum_{i=1}^{k} \sum_{x_k \in X} (\mu_{C_i}(x_k))^p \left\| x_k - v_i \right\|^2 - \lambda \left(\sum_{x_k \in X} \mu_{C_i}(x_k) - 1 \right) \tag{2.6}$$

By taking the partial derivatives for J with respect to μ and λ which equal to zero, and solving for μ we get the following membership function update equation:

$$\frac{\partial J}{\partial \lambda} = \left(\sum_{x_k \in X} \mu_{C_i}(x_k) - 1 \right) = 0 \tag{2.7}$$

$$\frac{\partial J}{\partial \mu} = p \cdot (\mu_{C_i}(x_k))^{p-1} \left\| x_k - v_i \right\|^2 - \lambda = 0 \tag{2.8}$$

$$\mu_{c_i}(x) = \frac{1}{\sum_{j=1}^{k} \left(\frac{\left\| x - v_i \right\|^2}{\left\| x - v_j \right\|^2} \right)^{\frac{1}{p-1}}} \quad , 1 \le i \le k, x \in X \tag{2.9}$$

Considering the partial derivative for the objective cost function J, with respect to v, which must equal to zero too, and solving for v, we get the following cluster center update equation [40-42]:

$$\frac{\partial J}{\partial v_i} = \sum_{x_k \in X} (\mu_{C_i}(x_k))^p (x_k - v_i) = 0 \tag{2.10}$$

$$v_i = \frac{\sum_{x \in X} (\mu_{c_i}(x))^p \times x}{\sum_{x \in X}^{m} (\mu_{c_i}(x))^p} \quad , \quad 1 \le i \le k \tag{2.11}$$

The convergence in the fuzzy c-means is guaranteed using the above membership functions $\mu_{ci}(x_i)$, and cluster center V_i, shown in equations 2.9 and 2.11, which are updated iteratively [35,38,39]. The iteration process is terminated by performing a convergence test, when the following criterion is met: If $max\ \{|V^t - V^{t-1}|\} < \varepsilon$ stop, Else update the cluster center and membership functions, where ε is a threshold for convergence criteria that has been predefined by the user initially.

The final fused output depends upon the final cluster centers and their membership functions as it will be discussed shortly in the next section. The expected output from the *FCM* clustering for $C = 3$ is shown in Figure 2.3. As it is noticed from the figure it models the drifted measured values shown in figure 1.5.

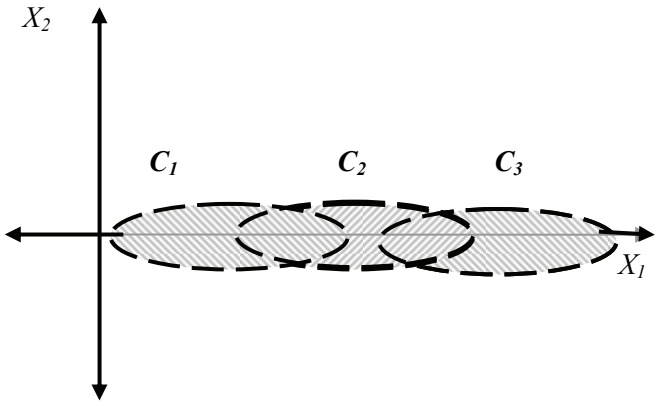


Fig. 2.3. *FCM* expected output.

IV. FUSION ENGINE

Each cluster membership function is represented as a binary code $BC_i \in \mathbf{2}^{\mathrm{m}}$. The creation of this code depends upon the membership functions of the clusters and a variable threshold level α such that:

$$BC_i(S) = \begin{cases} 1 & if \quad \mu(s) < \alpha \\ 0 & if \quad \mu(s) \geq \alpha \end{cases} \tag{2.12}$$

Where α is given as follows:

$$\alpha = 1 - \frac{1}{m} \sum_{j=1}^{C} \sum_{i=1}^{m} \mu_j(s_i) \log \mu_j(s_i) \tag{2.13}$$

The first term in the equation above is used for mapping the α values, while the second term is an entropy measure for the resulted clustering process from previous section.

The fused output will be the cluster center that achieves the following minimization argument for the generated binary codes:

$$\begin{aligned} S_f &= v(\mu_{i^*}) \\ i^* &= \arg\min_i (BC_1, ..., BC_i) \end{aligned} \tag{2.14}$$

The example shown in figure 2.4, demonstrates the code generation process for seven measured values by seven sensors. As it can be noticed from the figure, the measured values are softly partitioned among three clusters which resulted from the previous step,

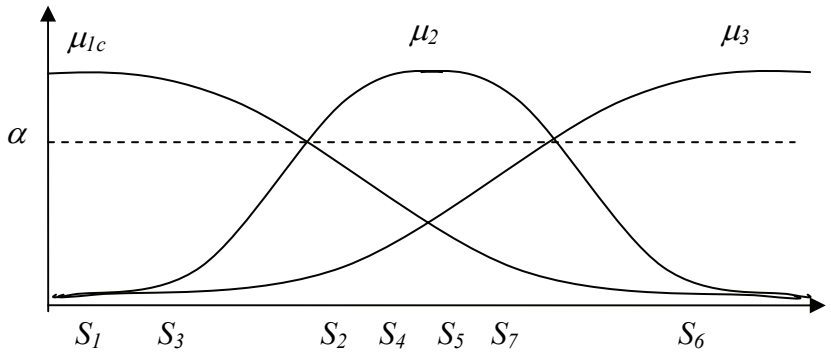
where the resulted membership functions are denoted as μ_1, μ_2 and μ_3. The three binary codes which represent the three membership functions are:

BC_1 0 0 1 1 1 1 1

BC_2 0 0 0 0 1 1 1

BC_3 0 1 1 1 1 1 1

BC_2 is the binary code which minimize the argument in (2.14) above, accordingly i^* is equal to *2*, as a result the single fused output will be the cluster center V_2, for the given seven sensor readings. The proposed approach guarantees the right fused output with the presence of (m-1)/2 faults.



BC_1 0 0 1 1 1 1 1 = $(11111)_2$ = $(31)_{10}$
BC_2 0 0 0 0 1 1 1 = $(111)_2$ = $(7)_{10}$
BC_3 0 1 1 1 1 1 1 = $(111111)_2$ = $(63)_{10}$

Fig. 2.4. Binary code generation process.

V. SELECTION OF THE BEST CLUSTER NUMBER

During the previous development of the fusion approach, the assumption that the measured values drifted in two directions around the acceptable region of measurements was adopted, which is the worst case scenario for possible occurring faults. As shown in figure 2.5, the drifted values are disrupted in two directions in such manner around the acceptable measurement region. In this case, three clusters (C=*3*) are required to model these measurements, where the resulted clustering was shown on figure 2.3.

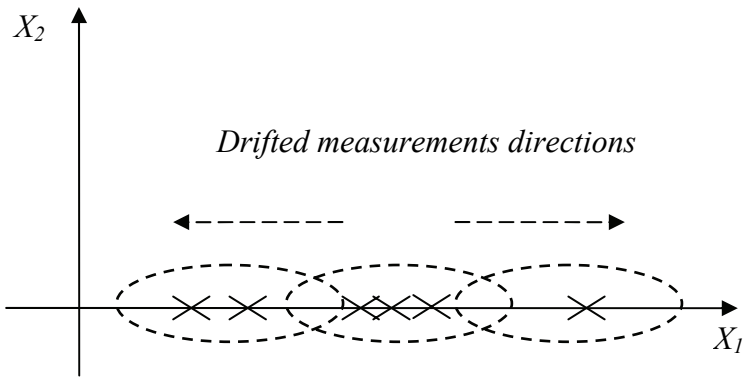


Fig. 2.5. Partitioning for the measurements with two directions for the drifted measurements.

However, there may be only one direction for the drifted measured values to the left or to the right of the acceptable measured data, in such cases two clusters are only needed in the fusion process (C=*2*) to model this case. These two drifted direction are

demonstrated in figure 2.6, where the deviated measurements are drifted to the right in part *a* of the figure, while in part *b* they are drifted to the left.

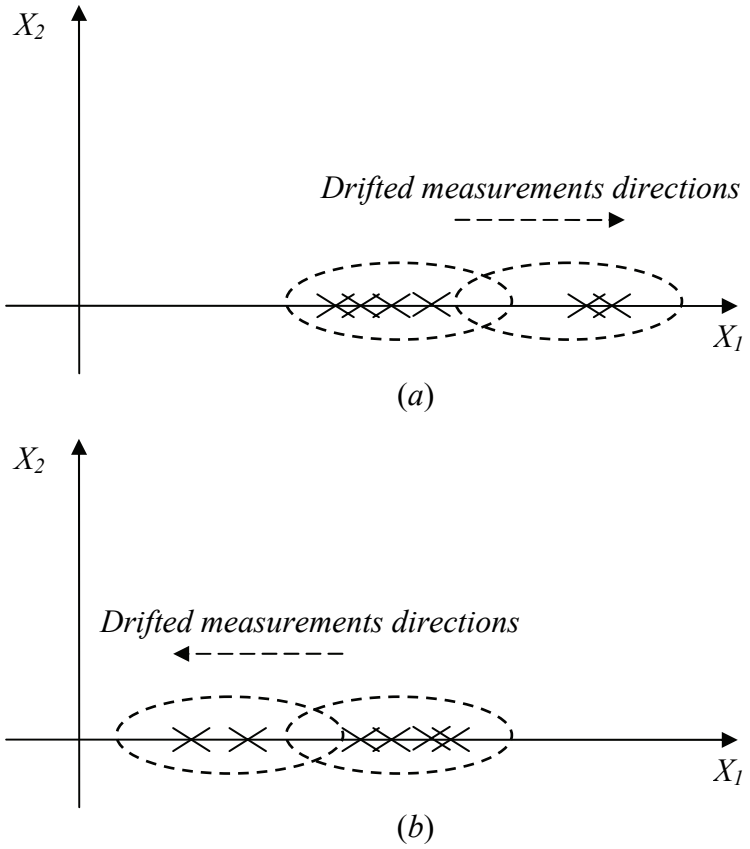


Fig. 2.6. Partitioning for the measurements with one direction for the drifted measurements, (*a*) to the right and (*b*) to the left.

So, it is very important to know the right cluster numbers at each time instant during the real-time system operation. For this purpose non-parametric density estimation is used to give the right decision for the cluster number required for the input data at each

time instant, where no parametric assumption required during the estimation process. The smooth kernel density estimator (SKDE) given by the following equations is used to achieve this goal [42-44]:

$$P(S) = \frac{1}{mh} \sum_{i=1}^{m} K(\frac{s - s_i}{h}) \tag{2.15}$$

$$K(s) = \frac{1}{(2\pi)^{1/2}} \exp(-\frac{1}{2} s^T s) \tag{2.16}$$

where h is the length of the estimation window. The result from applying the smooth kernel density estimator (SKDE) for a given inputs of measurements is shown in figure 2.7, where the maximum peak in the figure is denoted as *Max* and located by the dotted lines.

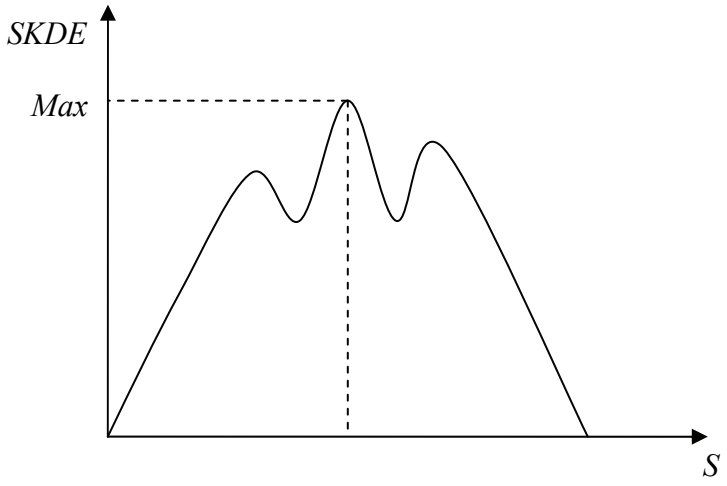


Fig. 2.7. Smooth Kernel Density Estimator (SKDE).

Subsequently, the number of clusters is determined according to the following two rules, based on the maximum peak location of the kernel estimator:

- If the maximum peak of the kernel estimator is left or right skewed then $C= 2$.
- If the maximum peak of the kernel estimator is centered then $C = 3$.

VI. FUSION ALGORITHM

Figure 2.8 shows the block diagram for the presented total fusion approach [45, 46]. Initially, the system will collect the input data measured by the m sensors, and then clusters the collected data using the fuzzy c-means clustering algorithm. The number of the resulted clusters depends on the direction of the drifted measurements which provided by the SKDE. After that the system generates the labeling binary codes for each resulted cluster. Finally these binary codes are processed by the fusion engine to produce the fused output S_f for the given input data.

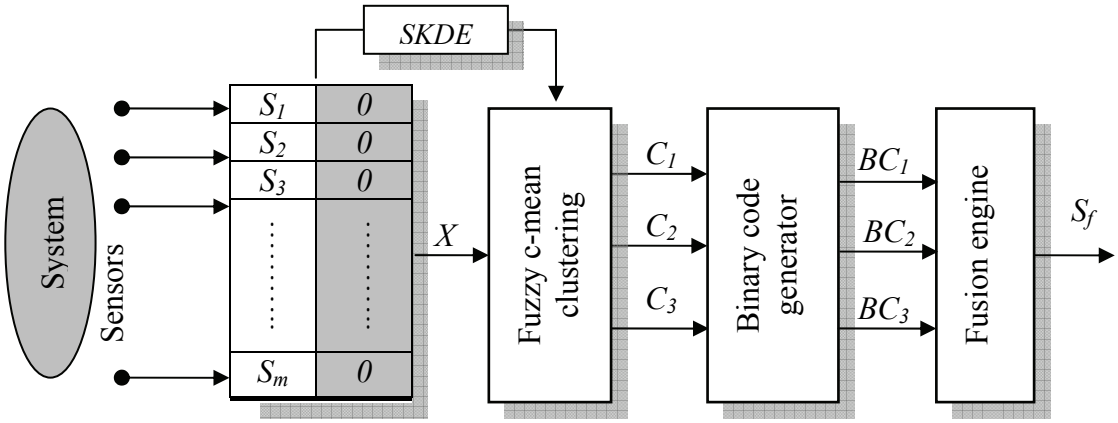


Fig.2.8. Block diagram for the total fusion process.

The fusion algorithm from the previous section and block diagram can be specified in the following steps:

1- Collect the input values S.

2- Compute C, by using (SKDE).

3- Initialize cluster centers V, ε.

4- Compute the membership function μ_{ci} using equation 2.9.

5- Update the cluster centers V, using equation 2.11.

6- Test for convergence: If $max\ \{|V^t - V^{t-1}|\} < \varepsilon$ stops, Else go to step *4*, where ε is a Threshold for convergence criteria has been initialized in step *3*.

7- Compute α.

8- Generate the binary codes BC_i .

9- Find the fused output S_f, *which* minimize the argument 2.14.

VII. HYBRID FAULT DETECTION SYSTEM

In this section the proposed fault detector based on artificial immune system (AIS) is presented. The artificial immune systems have drawn a significant attention recently in wide areas of applications, such as:

- Feature extraction [47].
- Pattern recognition [48,49].
- Learning techniques [50, 51].

- System control [52-55].
- Fault detection [56,57].

The artificial immune system main immunological principles are shown on the block diagram in figure 2.9 [58], which include the following categories [58-60]:

- The mechanisms of negative selection.
- Immune network theory.
- Clonal selection principle.

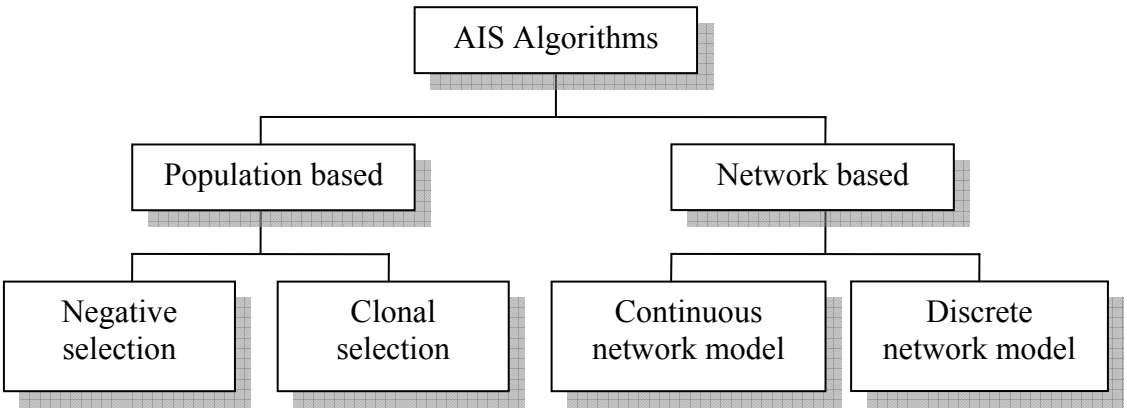
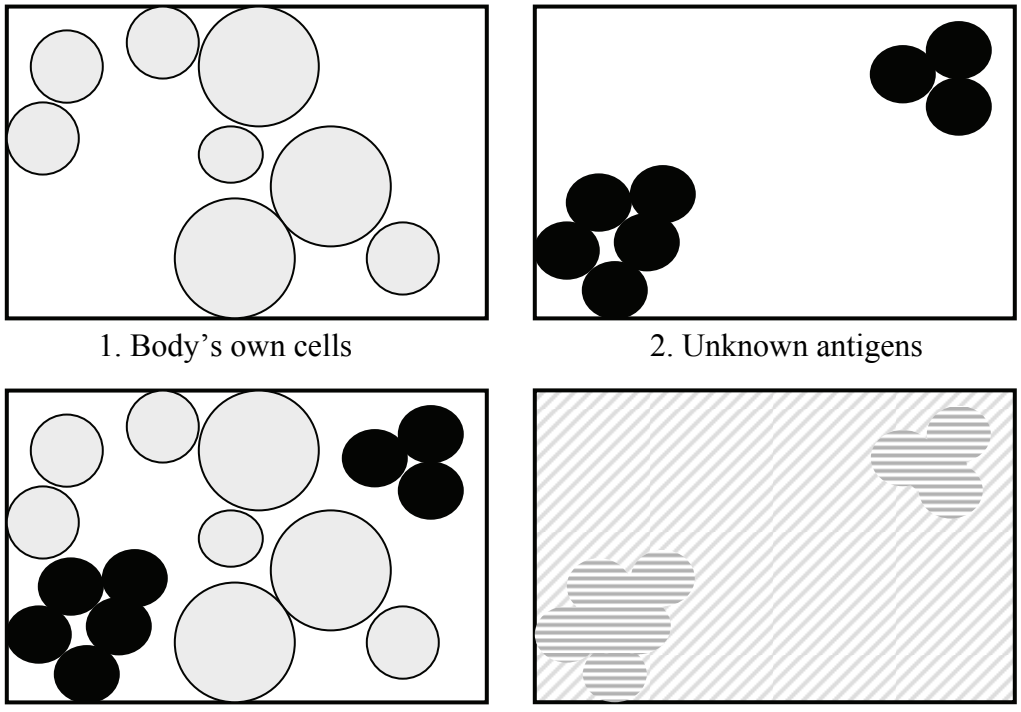


Fig.2.9. The Artificial Immune System (AIS) main immunological taxonomy.

The immune network theory is based on the Jerne's network theory [59], which suggests that the immune system maintains a network of interconnected immune cells (B-cells) for antigen recognition [59].

The clonel selection principle is based on the basic immune system adaptive response properties to antigens. In this principle only those immune cells which are able to recognize the antigens will be proliferated and differentiated into effectors cells [59].

The first artificial immune system based on negative selection mechanism was proposed by Fosset [59,61], which depends on the immune system ability to recognize unknown foreign antigens as a non-self body's cells from those of the body's own cells. The main principle of this proposed system is based on normal/abnormal discrimination of the system behavior which is similar to the immune system self/non-self classifications for the body cells.



1. Body's own cells

2. Unknown antigens

3. Body's own cells with unknown cells.

4. Negative selection for the unknown cells.

Fig. 2.10. Self/non-self discrimination negative selection mechanisms.

The negative selection main idea is illustrated in figure 2.10 [62]. As it can be noticed from the figure, part 1 shows the body's own cells as gray circles, and the unknown antigens are shown in a black circles in part 2. Part 3 shows the instance of time when the unknown antigen appears in the body. The immune system negative selection approach will be able to recognize those unknown cells from the body's own cells as shown in part 4 of the figure, where the self part or region during this detection process is characterized by slanted gray pattern lines, while the non-self part or region is characterized with gray horizontal pattern lines.

Considering the multiple sensor system as shown in Figure 2.11, we have a multiple input, single output system at all time instants. The system output is the correct fusion for the normal fault free system behavior, while there could be deviation for some inputs from this fused output which can be considered as abnormal behavior for these deviated inputs.

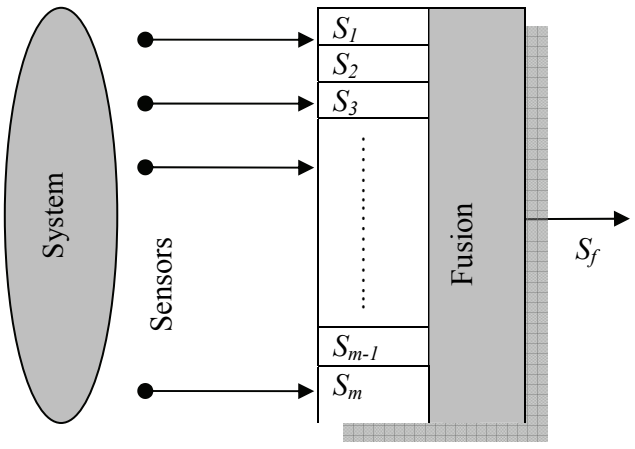


Fig. 2.11. Multiple sensors fusion system block diagram.

A fault detector can be defined based on the artificial immune system negative selection mechanism [59, 61, 63, 64], which is based on the immune system ability to recognize unknown antigens as non-self state from the body's own cells as a self state. In the proposed fusion system, the single fused output is considered as the self state, while the space complement over the detection window or period is considered as the non-self states. Figure 2.12 describes the detector implementation for both the self/non-self state during the detection window, where the detector will match a self state if the fused output remains within the detector self range; otherwise a signal will be generated to indicate a non-self match case outside the detector self range [64]. As it is noticed from figure 2.12, the faulty sensor signal is shown as a dotted line and it matches a non-self state after the third detection window, as a result a fault is detected at that window.

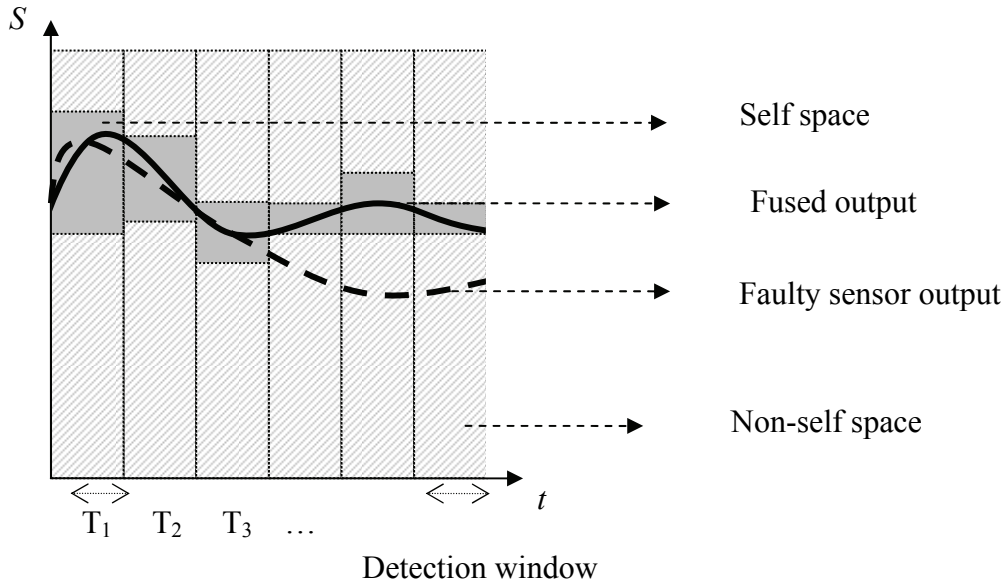


Fig. 2.12. Self/ non-self detected regions.

According to the previous definition of the negative selection mechanism for fault detection, the mapping table 2.1 is adopted for mapping the immune system negative selection mechanism to a fault detector [65].

Table 2.1

Immune system negative selection mechanism to hybrid system for fault detection mapping

Immune system	Hybrid system
Self	Normal operation (fused output)
Non-self (antigen)	Faulty operation (faulty measured values)
Antibody	Detector and matching conditions
Inactivation of antigen	Faulty signal and normal operation for the system.

As it is clear from the table, the immune self state is mapped to the system fused output, while the deviated or drifted measurements from this fused output are considered as the immune non-self state or antigen. The detection process and the matching rules are equivalent to the antibody working principle, while the fault detection signal and the isolation of the fault effect are equivalent to the inactivation of the antigen effect on the immune system.

So the main steps in the negative selection mechanism for fault detection are: defining the self normal system behavior, the detector implementation and matching conditions or rules, and finally monitoring faulty matching cases.

For the detector implementation, the detector sensitivity depends on the detection period such that the smaller the detection period the better is the sensitivity, the detector width is defined by the detection period or window. The detector self range or height can be determined by the fused output and the input data elements which do not belong to the fused binary code. The detector heights defined by the upper and lower self bounds are given according to the following equations:

$$Detector\_upper\_bound = S_f + \gamma \min_i \| S_f - s_i \|, s_i \notin S_f \tag{2.17}$$

$$Detector\_lower\_bound = S_f - \gamma \min_i \| S_f - s_i \|, s_i \notin S_f \tag{2.18}$$

Where γ is a scaling factor, $\gamma \in (0, 1)$, selected between zero and one according to the following; Let ΔD is the difference between the upper and lower detector boundaries as follows:

$$\begin{aligned} \Delta D &= Upper\_bound - Lower\_bound. \\ \Delta D &= 2\gamma \min_i \| S_f - s_i \|. \end{aligned} \tag{2.19}$$

Let ΔS is the difference between the fused output S_f and any of the measured values s_i:

$$\Delta S = S_f - s_i. \tag{2.20}$$

This measured value is faulty, if ΔS is greater than half ΔD value, where it will be outside the self detection region for the detector:

$$\begin{aligned} &\frac{\Delta D}{2} < \Delta S \\ &\gamma \min_i \| S_f - s_i \| < S_f - s_i \\ &\gamma < \frac{S_f - s_i}{\min_i \| S_f - s_i \|} \end{aligned} \tag{2.21}$$

So γ should be less than the relation shown above, if s_i is the same in the nominator and the dominator, γ can be selected to be any value less than one:

$$\gamma < 1 \tag{2.22}$$

Figure 2.13 describes the detector components [64], the self detector region or body is shown as gray box, while non-self region is the complement of the self region over the detection window is shown as slanted gray pattern lined box for both of the detection windows.

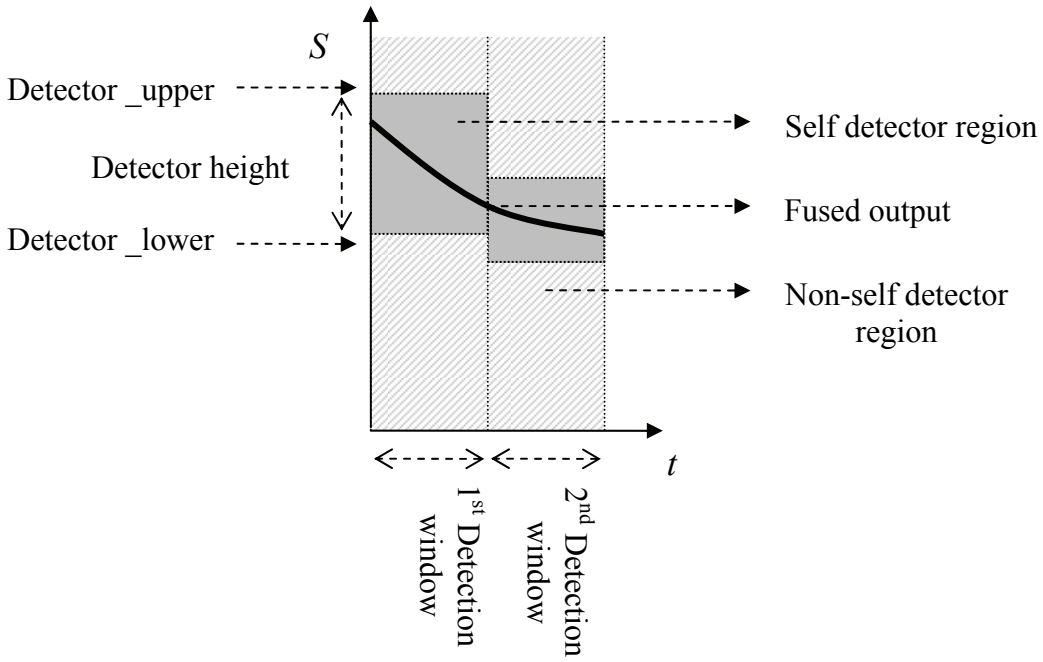


Fig. 2.13. Detector components.

The detection process starts by generating a blank detection window, and then the fused output from the fusion engine is applied to the detector as well as the input data in order to generate the detector upper and lower boundaries. As a result the detector self region is created.

Once the boundary of the self region is complete, the artificial immune system enters the monitoring process to detect any input faults. When any of the input data set fails to match the self detector region, a fault is detected at this operating instant and a signal is sent to the display panel to visualize the occurring faults.

These steps are demonstrated in figure 2.14 [64], where in part *a* the initial detection window was created, while in parts *b* and *c* the fused output was applied, as a result the

detector self and non-self regions was implemented. Finally in part *d* the monitoring process for fault detection is started.

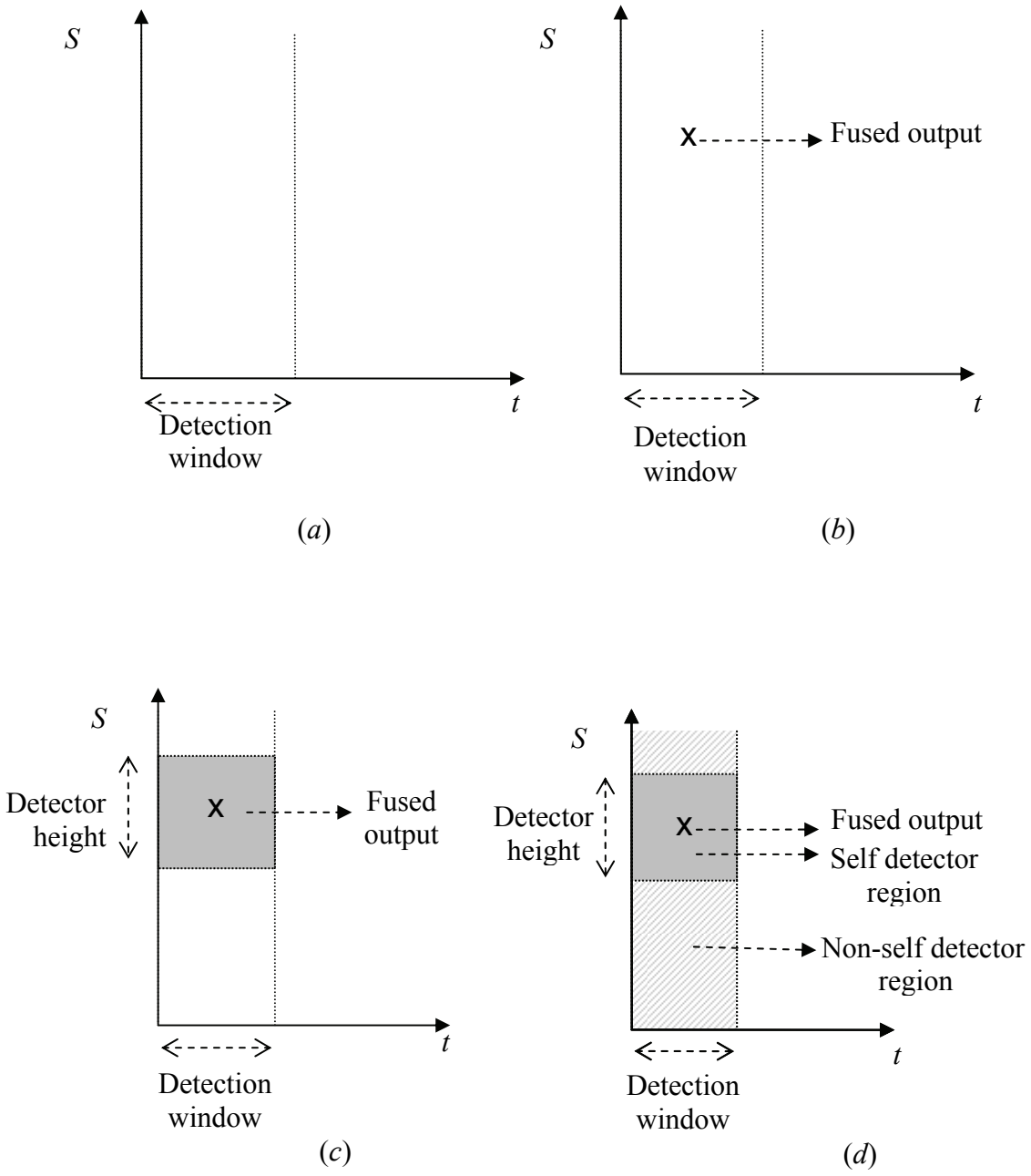


Fig. 2.14. Illustration for the detector implementation steps.

The final implemented hybrid system is shown in figure 2.15 [66]. The figure shows the block diagram for the total presented hybrid approach, starting with the measurement process and ending with the fault detection process based on the fused output generated by the fusion engine.

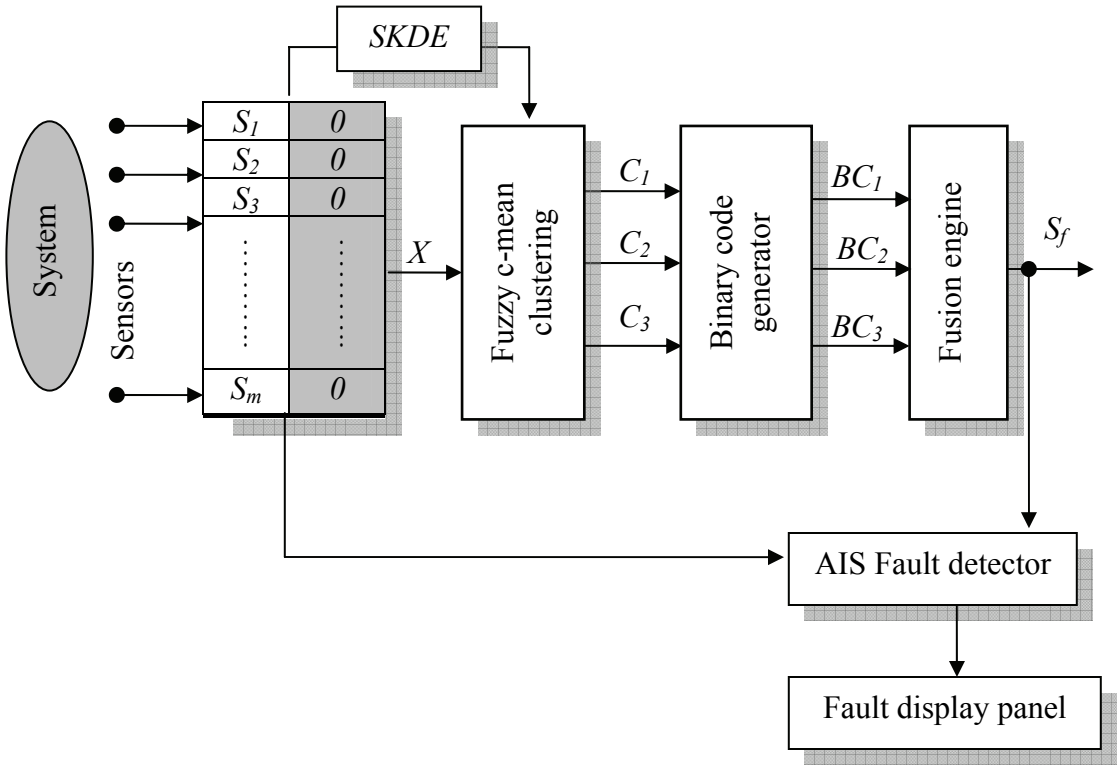


Fig. 2.15. Hybrid system block diagram.

The hybrid system mainly consists of two parts; the first part is the fusion part which includes the input part that collects the measured data for the clustering part. Next the system will cluster these data and generates the labeling binary codes for each resulted cluster. Finally, the fusion engine will produce the fused output for the given inputs. The

second part is the fault detection part based on the negative selection mechanism, which includes the fault detection unit and the fault display unit. A fault counter can be integrated in the final unit to count the number of detected faults during the operation process. This counter will generate the required signal for the system administrator to take the required maintenance actions when the number exceeds (m-1)/2, where *m* is the number of inputs.

CHAPTER III

PERFORMANCE EVALUATION

I. INTRODUCTION

This chapter focuses on the performance evaluation of the hybrid system approach. For this purpose, several tests had been conducted based on simulated experimental data and real time experiments. In the first test the results from a simulated target tracking or a navigation application by multiple sensors is presented. The results from this simulation are compared with other fusion and fault detection approaches for the evaluation purpose. In the second performance evaluation a mobile robot localization experiment was conducted to determine the mobile robot location in a predefined path based on the information available from the sensors mounted onboard. A distributed wireless sensor network is addressed in the final performance evaluation for measuring indoor light intensity through distributed sensing nodes.

This chapter is organized as follows; the conducted simulations and results are covered in section two. In the sequel an evaluation for the hybrid system performance is presented in section three by comparing the hybrid system results with other approaches. Finally, in section four and five the conducted mobile robot localization and the distributed wireless sensor network experiments and their results are demonstrated.

II. PERFORMANCE EVALUATION I

The proposed approach will be confirmed by using the following simulation for signal fusion of a target tracking or a navigation application, which is a modified simulation for multiple sensors system that has been used in [19].

Here, five input noisy commensurate sensors are used to measure an object motion which is assumed to be linear during the simulation time respectively. In fact, some of the input sensors are assumed to have non-linear characteristics to agree with the non-linear characteristics of the navigation sensors [19].The sensors mathematically are modeled as follows:

$$S_1 = x + n_1 \tag{3.1}$$

$$S_2 = x^{1.02} + n_2 \tag{3.2}$$

$$S_3 = x + \sin(0.3x) + n_3 \tag{3.3}$$

$$S_4 = x \times e^{-0.003x} + n_4 \tag{3.4}$$

$$S_5 = x^{1.08} + n_5 \tag{3.5}$$

where S_i is the sensor real measurement, x is the sensor actual measurement, and n_i is the normal random zero mean additive noise, i = 1, …, 5. The five sensor outputs are shown in figure 3.1.

As it can be noticed from the figure the fault free case is plotted, where the five sensors track the same target which moves in a linear manner, the non-linear characteristics for some of the sensors are clearly seen from the plots.

To evaluate the proposed approach in critical real life situations, some of these sensors are subjected to faults at different time instants, respectively. Were the sensors are affected with different type of faults which made the measured values deviated from the assumed correct measurements. To make the simulation relatively a critical example, we assumed that some of those sensors have problems with sensing the real measured value which is subjected to 75% drop compared to the correct measurement, while others are subjected to parameter changes during the simulation run time.

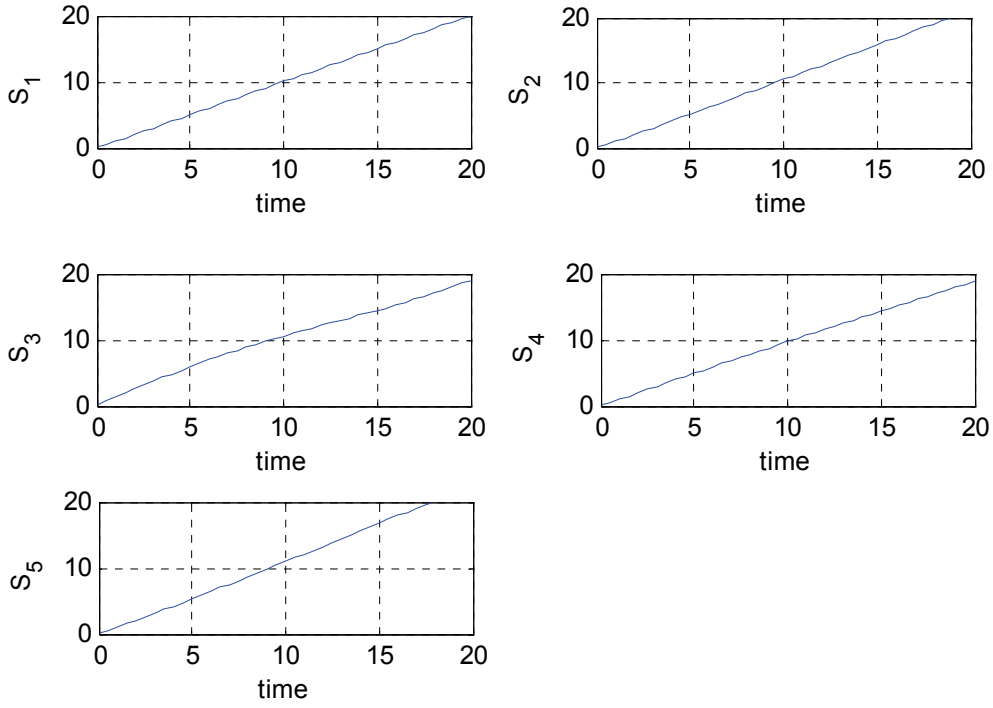


Fig. 3.1. The five sensor outputs with no faults.

The two sensors that are subjected to these two faults are sensors number four and five (S_4, S_5). S_4 is subjected to a 75% drop in the measured value, while S_5 is subjected to

parameter changes; the other sensors values are not changed. These changes are demonstrated in figure 3.2, after comparing this figure with figure 3.1, the resulted changes in the measured values of sensors S_4 and S_5 are clearly noticed.

The above sensors models are simulated as input signals to the hybrid system for multiple sensor fusion and fault detection. The resulted fused output after applying these inputs from the sensors is shown in figure 3.3; the fused output is denoted as S_f shown in the plot as solid line, while the detector upper and lower boundary lines are shown in dashed lines.

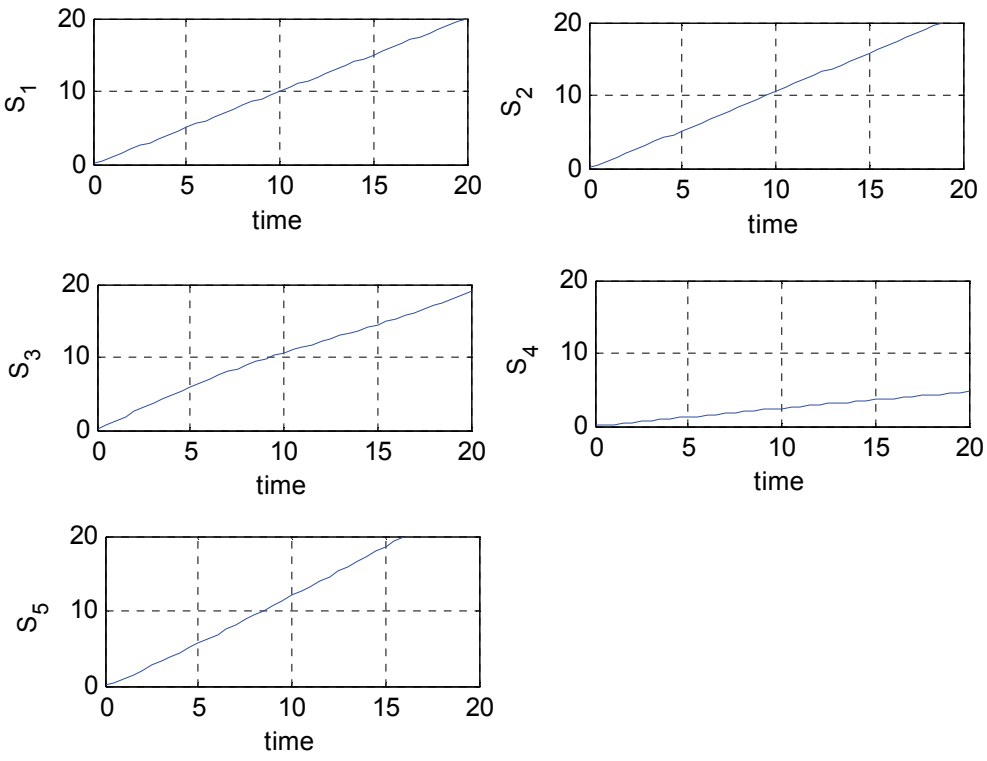


Fig. 3.2. The five sensor outputs with two faults at sensors four and five (S_4, S_5).

When comparing the fused output figure 3.3, with the measured values in figure 3.1, and the sensors faulty measured values in figure 3.2, we can see, as expected, that the presented approach is able to minimize the effect of the deviated or drifted measured values on the final fused output. The hybrid system is able to give the correct fusion for the tracked target from the sensors measured values even with more than one faulty sensor input in a very satisfactory way.

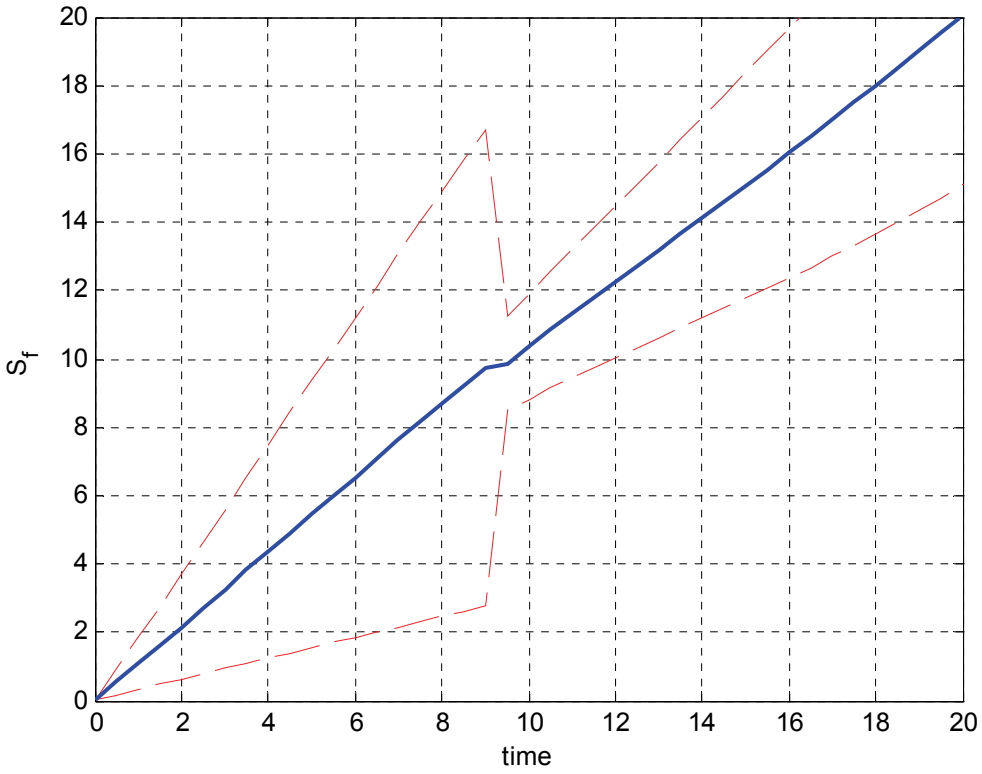


Fig. 3.3. The fused output S_f, the detector upper and lower boundaries.

The figure shows also the fault detector upper and lower bounds which are defined by equations 2.16 and 2.17. The bounds are shown as dashed lines above and below the

fused output. This figure agrees with the fault detector implementation figure 2.13 shown in chapter II, where the area between the two lines defines the self region, and the complement region outside the two dashed lines is the non-self region.

The generated signals for the detected faults are presented in a time progress display panel showing the detected faults. The detected faults are presented as a yellow bar, which started at the time instance when the fault was detected by the AIS detector and its size increases as long as the fault is still detected based on the matching with the non-self region defined in the figure 3.3. This display panel is presented in figure 3.4.

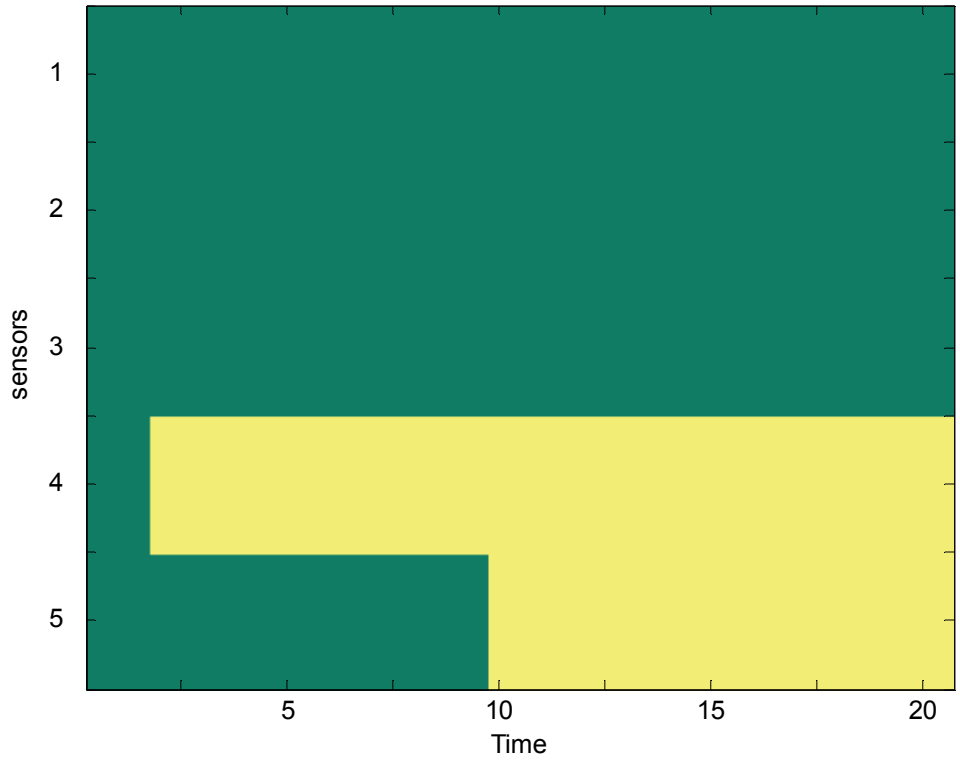


Fig. 3.4. The display panel for the detected faults.

By examining figure 3.4, we can see that two faults were detected during the simulation time. This result agrees with the two simulated sensors faults (S_4, S_5) during the run time. After comparing figure 3.4 with the simulated inputs sensor signal in figure 3.2, we can see that the first fault was early detected at time equals to *2* seconds which was caused by S_4. This result agrees with the sensor four (S_4) measurements faulty trend, when it's started to deviate before *t* equals to 5 seconds, as it is clear from sensor four (S_4) plot shown in figure 3.2.

From the display panel we can notice the second fault detection process started at time equals to *10* seconds which was caused by sensor number five (S_5). This detected fault agrees with the measured values trend for this sensor. When comparing the fault free case of figure 3.1 and the faulty measured values of figure 3.2 with the fused output in figure 3.3, we can notice that the measured values by this sensor tend to deviate rapidly above the fused output in a clear manner after a time instant equals to *10* seconds.

As it is indicated from the simulation results, the hybrid system is able to give the correct fused output for the tracked target in a satisfactory manner during the simulation time, even with the presence of two faulty input sensors. The AIS detector was able to detect these faults and gave the right indication to the administrated process for any required maintains actions afterwards. These indications were presented in the fault display panel (figure 3.4).

The number of clusters used during the simulation time by the hybrid system is shown in the plot figure 3.5. As it can be noticed from the plot, the rules used to determine the

right number of clusters during the fusion process according to the deviation direction in the sensors measurements are working as good as expected.

With referring to the detected faults presented in the display panel in figure 3.4 and the faulty sensors plots in figure 3.2, we find that only one direction for the deviated measurements from the actual measured value during the first *10* seconds of the simulation time is noticed. This deviation was caused by sensor number four (S_4), as a result two clusters are required to model the input data, and this agrees with the plotted result in figure 3.5 for the first part of the simulation.

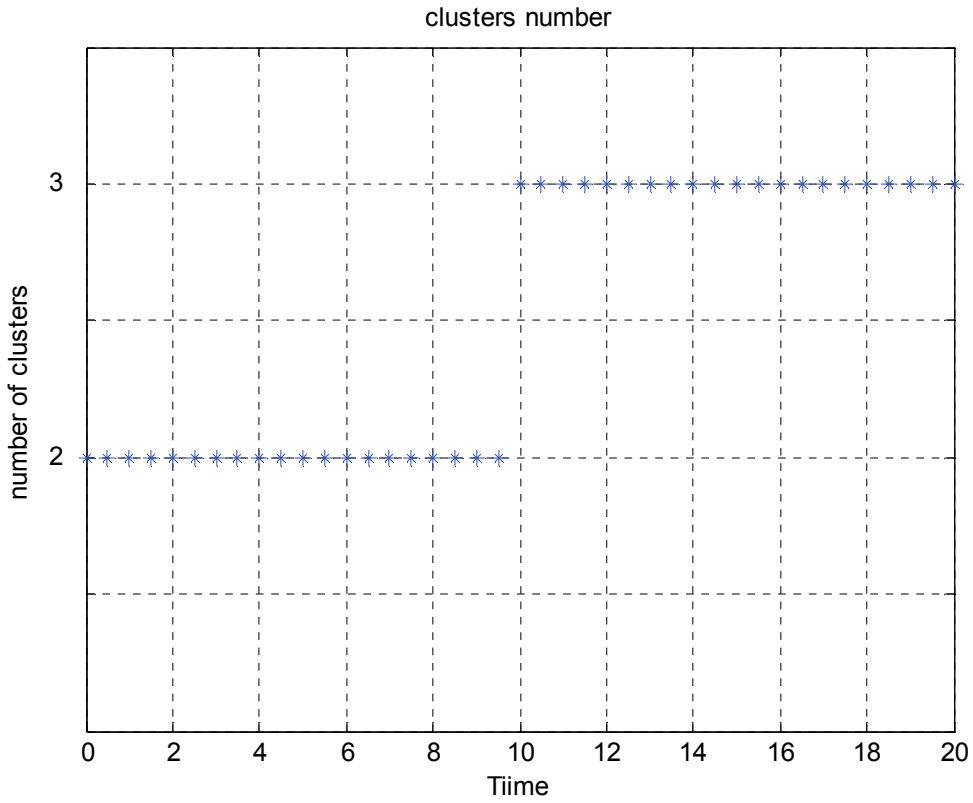


Fig. 3.5. The required number of clusters.

After the first 10 seconds, the first fault was still detected and the second fault can be noticed now. This fault is caused by sensor number five (S_5), deviated above the actual measurements. As a result three clusters are required to model the input data, and this also agrees with the clusters number determined by the rules based on the SKDE through out the simulation time. As it is presented in figure 3.5 after the first *10* seconds.

The calculated α values by equation 2.13, during the simulation time for the binary code generation process, are shown in figure 3.6. As it can be noticed from the plot the values are less than one which agrees with the membership values definition to be between zero and one.

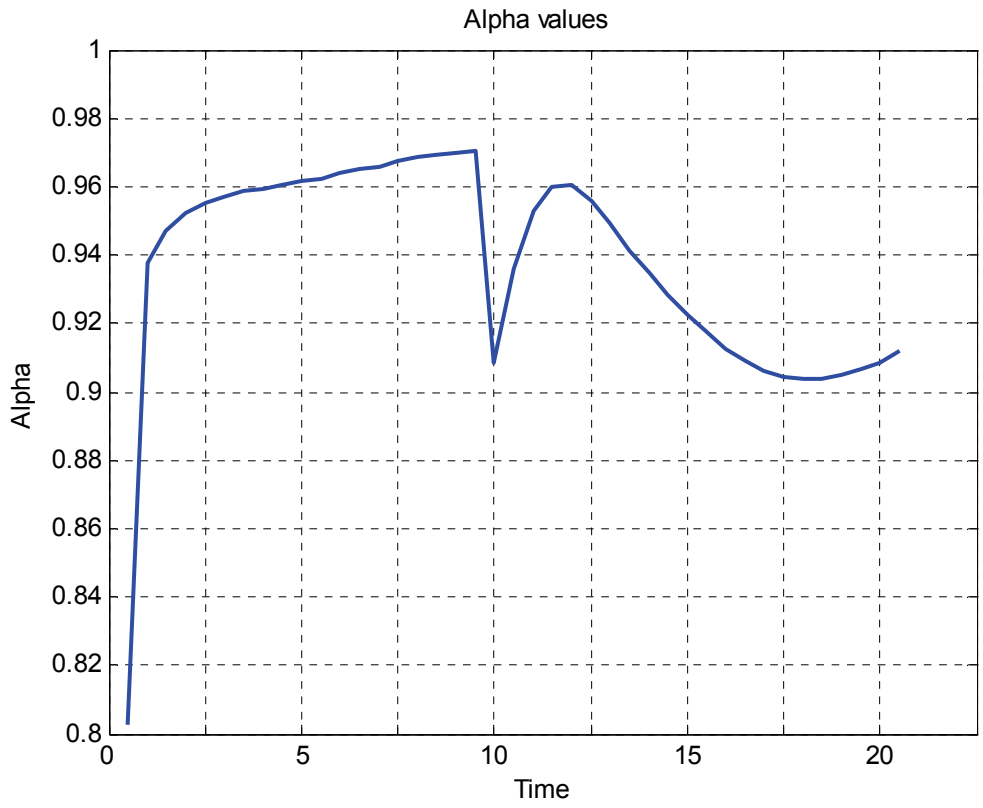


Fig. 3.6. α values plot.

The over all resulted plots from the simulation are grouped in one figure for a better demonstration, as shown in figure 3.7. The figure shows the five sensors input signals (S_1 - S_5), the fused output (S_f), the detector upper and lowers boundaries, the fault display panel, and finally the cluster numbers used through out the simulation time.

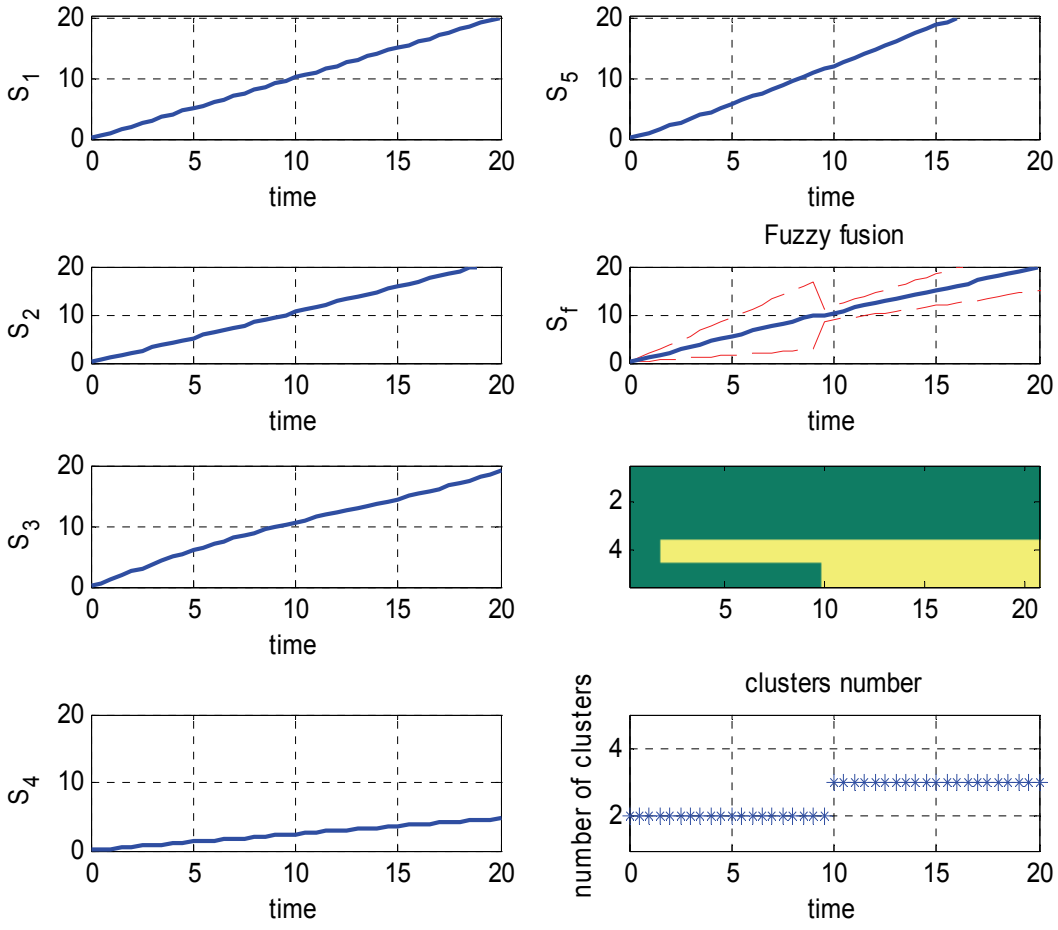


Fig. 3.7. Demonstration for the simulation results. Shows the five sensors input signals (S_1 - S_5), the fused output (S_f), the detector upper and lowers boundaries, the fault display panel, and finally the cluster numbers used through out the simulation time.

As it was expected from the presented approach, the effect of the deviated input measurements was minimized in the fused output, which was a very good fusion for the correct measured values, even with more than one faulty input which was detected by the AIS fault detector in a very reasonable way.

III. PERFORMANCE EVALUATION II

The main focus of this section is to demonstrate the efficiency of the presented hybrid system approach by comparing the performance with other fusion algorithms as well as fault detection approaches.

For this purpose the new hybrid system approach is compared to one of the popular sensor fusion methods, for instance the ordinary linear averaging fusion, and to one of the new approaches, i.e. nonlinear attractor fusion [9, 67, 68].

For the linear averaging fusion, the fused output x_{fl} is given by the ordinary arithmetic mean value for the given set of input sensors data as in equation 3.6. The nonlinear attractor fusion is looking to solve the state equation in 3.7 to provide the fused output value x_{fn}.

$$x_{fl} = \frac{1}{m}\sum_{1}^{m} x_i \tag{3.6}$$

$$\dot{x}_{fn} = \sum_{1}^{m} \lambda_i (x_i - x_{fn}) f_i (x_{fn}) \tag{3.7}$$

where λ is the sensor estimation relative strength and f is a nonlinear function defines the local attraction range for the sensor. This function can be defined for example as a Gaussian or Bell-shaped function.

The simulated sensors inputs presented in section II (equations 3.1-3.5) are used during the performance evaluations to run several combinations of sensors failure scenarios. The root mean squared error (RMSE) is used as an evaluation measure for the simulation results between these approaches. Three main scenarios are conducted;

- All five sensors are valid with no faulty sensors.
- Four sensors are valid and one sensor is faulty.
- Three sensor measurements are valid while the other two sensors are faulty.

The resulted root mean squared error (RMSE) values for different simulation scenarios are shown in Table 3.1.

Table 3.1

Root mean squared error (RMSE) values for different simulation scenarios

Simulated sensors scenario	***RMSE***		
	Hybrid system	Linear averaging	Non-Linear averaging
5 valid sensors	0.0238	0.0886	0.0920
4 valid & 1 faulty sensors	0.0231	0.0855	0.0941
3 valid & 2 faulty sensors	0.0478	0.6267	0.6230

By comparing the root mean squared error values in the table for the three fusion approaches, the hybrid system, the linear fusion approach, and the nonlinear fusion, we can notice that the hybrid system is able to maintain minimum error values compared to the other approaches during all the simulation scenarios. In all sensor valid scenario the hybrid system fusion got the minimum RMSE value which is *0.0238*, while the linear fusion approach got *0.0886* and the nonlinear fusion approach got *0.0920.*

A similar trend is achieved in the other two scenarios for the RMSE values as shown in the table. The hybrid system is able to give the best fused output for the given input sensors measurement and to maintain minimum RMSE values.

In the last simulated scenario addressing the two faulty sensors problem, the hybrid system is maintaining the best performance in spite of the RMSE value increase, while the other approaches suffer from a remarkable increase in the RMSE values, compared to the one faulty sensor case addressed in the second simulation scenario. For instance, the linear fusion approach increase from *0.0855* for the one faulty sensor to *0.6267* for the two faulty sensors case. As it can be noticed from the demonstrated results in table 3.1, the appearance of more than one faulty sensor has a remarkable effect over the fusion approaches performance.

The fused outputs from the three approaches for two sensors faulty scenario are shown in figure 3.8. The error plots is generated by taking the difference between the fused output and the actual linear target motion which is assumed to be linear through out all the simulation times are shown in figure 3.9.

As it can be noticed from the plots, the fused output by the hybrid system (which is shown by stars) is able to track the actual measurement values (which are shown by dashed straight line) in more acceptable manner than the other approaches.

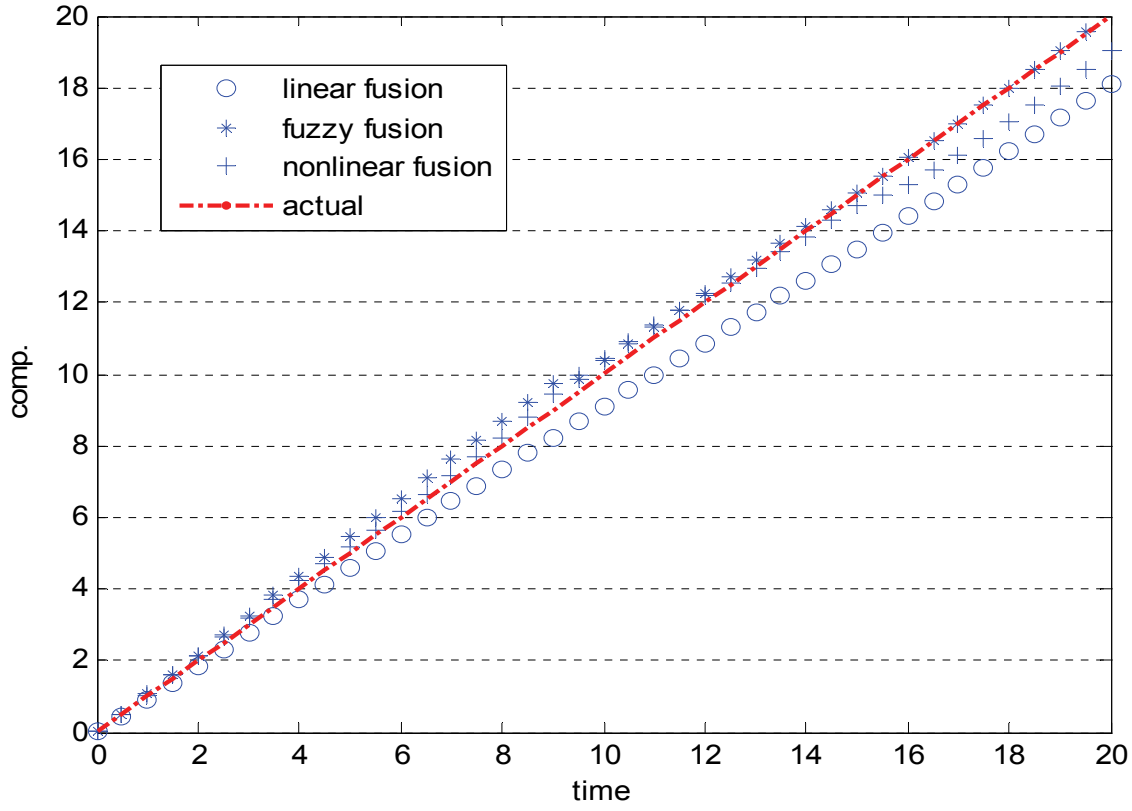


Fig. 3.8. The fused output from the three approaches.

It can be noticed from the tracked target plots by the linear averaging and non-linear approaches fused outputs (denoted by circles and plus signs respectively in the figure), had deviated from the actual straight line motion. And this deviation is clearer after the

presence of the second faulty sensor measurements towards the end of the simulation time.

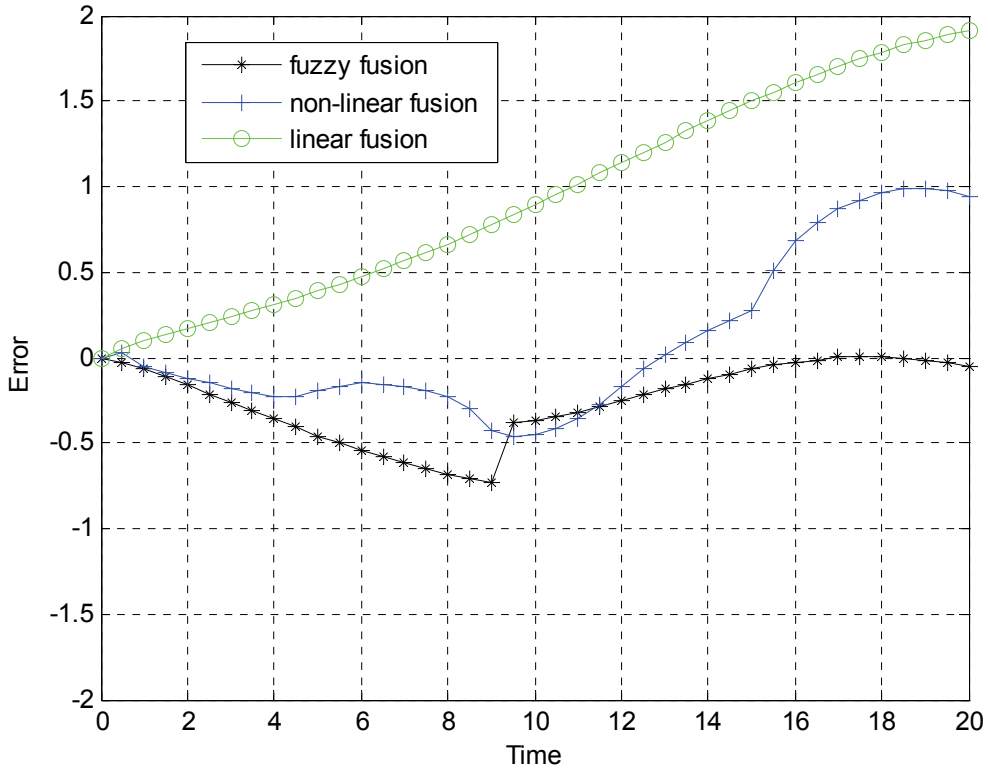


Fig. 3.9. Error plots.

Also, the error plots shown in figure 3.9, agree that the hybrid system was able to reduce the error, while the error in the other approaches increases towards the end of the simulation time. In figure 3.9 the error plot for the hybrid system is shown as stars, while the error plot for the linear fused output is shown as circles and for the nonlinear fusion as plus signs.

In the next part of this section the fault detection process is evaluated by comparing the performance with another approach for fault detection. Where a residual generator is used to monitor the changes in the sensors measured values based on the mean fused value for a given set of inputs [25,69], as shown in figure 3.10.

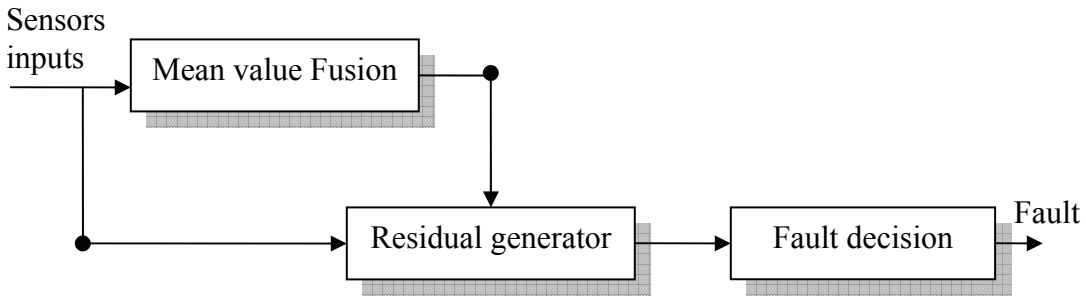


Fig. 3.10. Residual generator fault detection approach.

As shown in figure 3.10, five residual signals (R_1-R_5) are generated for the five input sensors during the simulation time. The linear averaging fusion S_{fl} given by equation 3.6 is used for the fusion process; as a result the residual signals are given as follows:

$$R_i = x_i - S_{fl}, \quad i = 1,...,m. \tag{3.8}$$

Where i is the number of the input sensors, and x_i is the measured value by sensor S_i. If any of the residual signals exceeds a predefined threshold d at fault free conditions a fault is detected by the fault decision block as shown in figure 3.10, a fault free measured input is defined by the following equation:

$$-d \le R_i \le d, \quad i = 1,...,m. \tag{3.9}$$

The five sensor inputs are applied to the fault detector presented in figure 3.10, addressing the case of two sensors are faulty; sensors four and five. The resulted residual signals are shown in figure 3.11 and the predefined thresholds -*d* and *d* are shown as dashed lines.

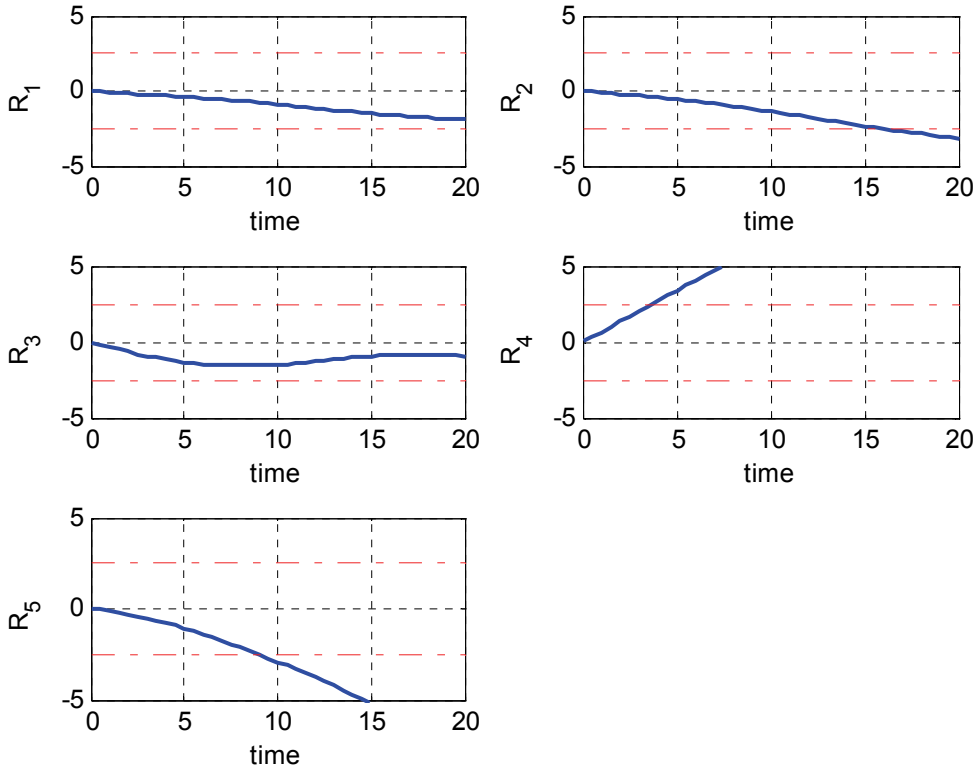


Fig. 3.11. Fault detection results with *d = 2.5*.

The threshold value is predetermined to be *2.5*. As it is demonstrated in figure 3.11 three faults are detected, two of them agree with the simulated sensor errors of sensors four and five (S_4, S_5). But there was a third fault detected belongs to sensor number two

(S_2) occurring toward the end of the simulation time as indicated from the residual plot in the figure. Since sensor two have no faults simulated during the simulation, this detection process is considered as a false alarm.

This problem is mainly raised from the determination of the predefined threshold *d* value, where increasing the threshold value to *3.5* will overcome this problem as shown in figure 3.12. The problem of predefining the threshold value can be solved by running the simulation several times addressing all the possible expected faults and finally determining the threshold level above which all the faults could be detected [25].

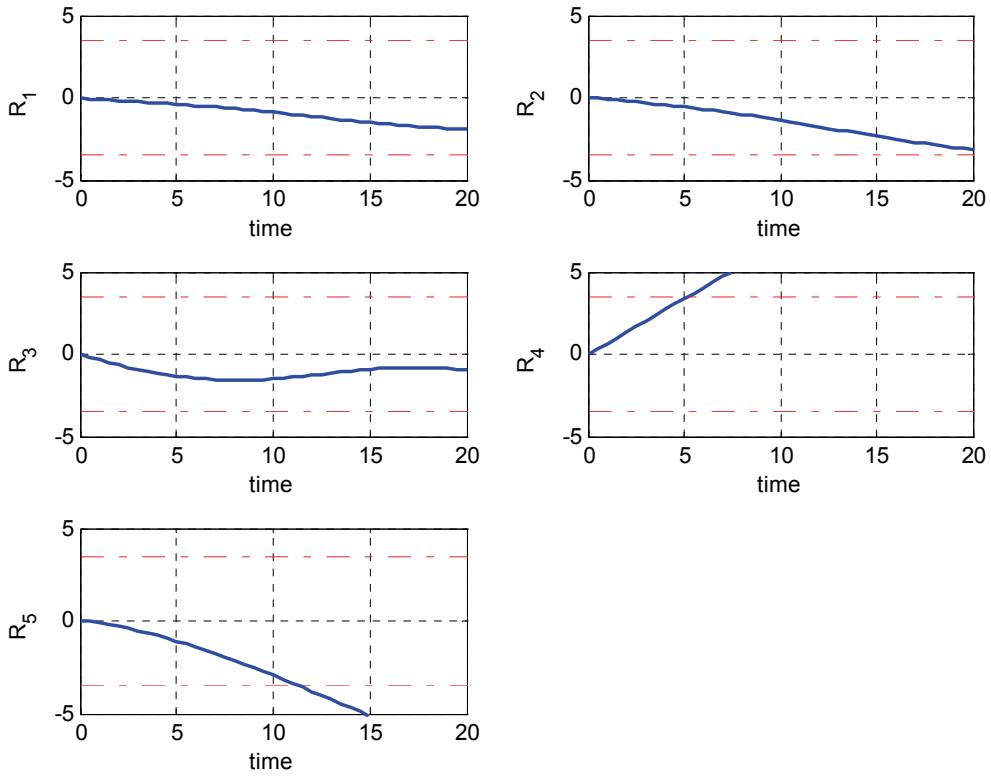


Fig. 3.12. Fault detection results with *d* = *3.5*.

As indicated from the pervious performance evaluation, it is clear that the hybrid system is able to detect the right faults as well as the correct fused output during the simulation in a very agreeable way compared to the assumed straight line motion for the tracked target through out the simulation time in spite of the simulated sensors faults.

IV. PERFORMANCE EVALUATION III

Multiple sensor fusion is widely employed in mobile robot localization as well as navigation for known and unknown robot surrounding environment, where a remarkable amount of research addressed such problem based on the collected information by the mounted sensors onboard of a mobile robot [70-74]. The mobile robots are commonly mounted with a set of sensors so that the robots can achieve the assigned tasks for them successfully. Mobile robots received a considerable amount of attention in the last decades in wide area of applications [30, 75].

These sensors could be of the same type or different types, but the main objective behind using all these sensors is to provide readable information about the robot location through out the robot navigation process. Such sensors include: ultrasonic range sensors, encoders, cameras, and laser sensors.

In this section a mobile robot localization example is addressed, by using the collected data from the onboard robot sensors to provide the correct fusion for the robot position by the hybrid system, while the robot is navigating in a known indoor environment.

For indoor mobile robot localization it's required to know the robot location with reference to a predefined coordinate system as shown in figure 3.13. According to this figure, the robot position is defined by the vector *OA*, where the head of the vector denoted as (x_A, y_A) represents the robot position in the (*X, Y*) coordinate. The robot position at point *A* can be determined through measuring the distance between the robot and the known environment boundaries in the directions of the four dotted arrows shown in the figure and denoted as *1*,*2*,*3* and *4*, where the known environment bounds in the figure are the room's walls. If the robot moves afterward to point *A'* for example, another measuring process can determine the new robot position and so on.

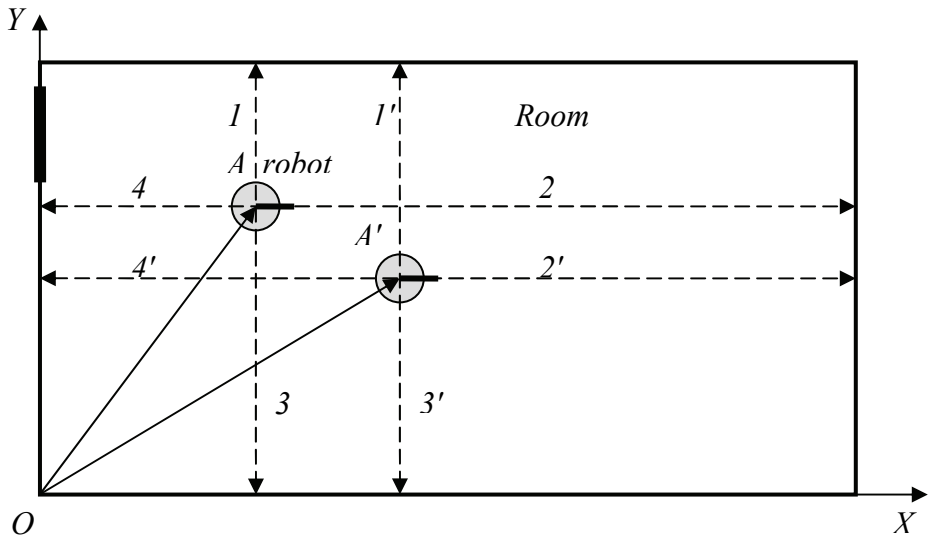


Fig. 3.13. Indoor mobile robot localization.

The robot used during the experiment is the Rug Warrior platform [76], this mobile robot is equipped with microcontroller, serial connection interface, three collision

detectors, two light sensors, infrared detector, shaft encoder, motor drive chip, and sonar rang finding system.

The measurement process was conducted by the mobile robot equipped with sonar sensor in three directions around the robot vertical axis which correspond to the three arrows *1*, *2* and *3* in figure 3.13, where the sonar sensor is commonly used for range finding tasks [30,75]. After that, the robot updated its position and performed another measurement process; this process was repeated by the mobile robot iteratively.

The layout for the experiment is shown in figure 3.14. This is a known closed room environment for the robot, where the robot travels over a predefined path shown as the dotted arrow in the figure.

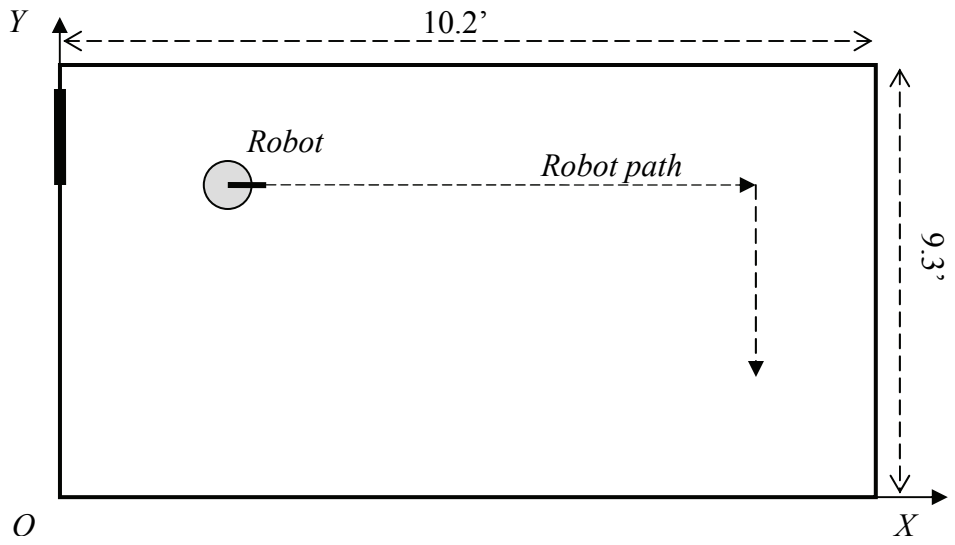


Fig. 3.14. The layout for indoor experiment showing the robot path.

According to the sonar readings and the robot position update, three pairs for the robot position are collected: (x_1, y_1), (x_2, y_2) and (x_3, y_3). The right robot position is fused

through applying the three robot position pairs to the hybrid system, where the robot fused position is denoted as (x_f, y_f).

As it was mentioned before, the sonar sensor is used to provide the distance information due to its ability to measure the time of flight of the detected echo wave. This wave results from the reflection of the initially emitted sounds wave from objects surrounding the mobile robot ahead the sonar sensor. The distance information is provided by the sonar according to the time difference between the emission and reception times of the sound wave as follows [30,75]:

$$d = \frac{1}{2} \; v \;\; \Delta t \tag{3.10}$$

where:

- d is the distance to the detected object.
- v the speed of the emitted sound wave, this speed at normal condition air medium is defined as 1138 feet per second. v is generally given as follows:

$$v \; = \; \sqrt{\gamma RT} \tag{3.11}$$

  where γ is the specific heat ratio, R is the gas constant and T the temperature in Kelvin.
- Δt is the time difference between the emitted sound wave and the reception times of detected echo. This time difference is defined as the time of flight, half of this time is required to measure the distance to the object, so the ½ appears in the equation 3.10 above.

The fused robot positions by the hybrid system are shown in figure 3.15 and 3.16, where the fused x_f values are presented in figure 3.15 by the solid line, while the fused y_f values are presented in figure 3.16 by the solid line. The upper and lower fault detector bounds are shown by the dotted lines above and below the fused position in both figures.

As it can be noticed from the figures and compared to the mobile robot path in figure 3.14, the robot initially is moving parallel to the X axis and after that turns right to continue moving down parallel to the Y axis. During the first stage, the robot position given by (x, y) will have an increasing x values while the y values will be mostly constant, after that the robot turns right where the x values will be almost constant while the y values start to increase.

This trend for the mobile robot position in the predefined path, agrees with the mobile robot fused position (x_f, y_f), shown in figures 3.15 and 3.16. To organize the robot's fused position in figures 3.15 and 3.16 so it can be more readily examined, the robot's fused position (x_f, y_f) is displayed in two coordinate configuration (X,Y), as shown in figure 3.17.

In this figure the robot fused positions (x_f, y_f) shown as a solid line, as well as the three pairs for the mobile robot position; (x_1, y_1) is represented as star signs line, (x_2, y_2) is represented as cross signs line and (x_3, y_3) is represented as plus signs line. The room's boundaries are shown as dotted lines which are the room walls, the mobile robot moving directions are shown by the arrows signing according to the mobile robot predefined path in figure 3.14.

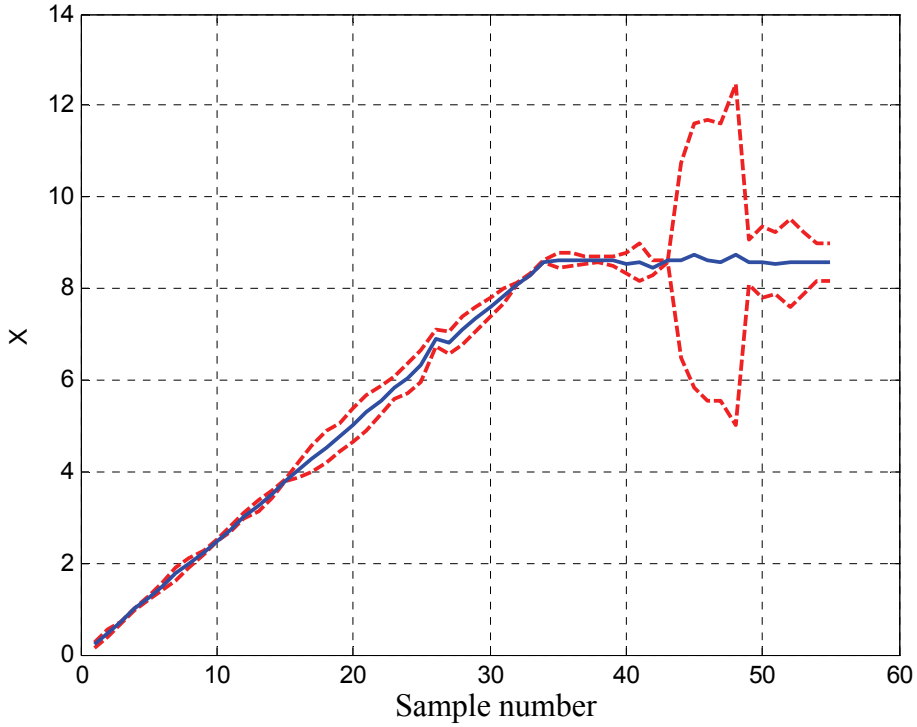


Fig. 3.15. The robot x_f fused output location.

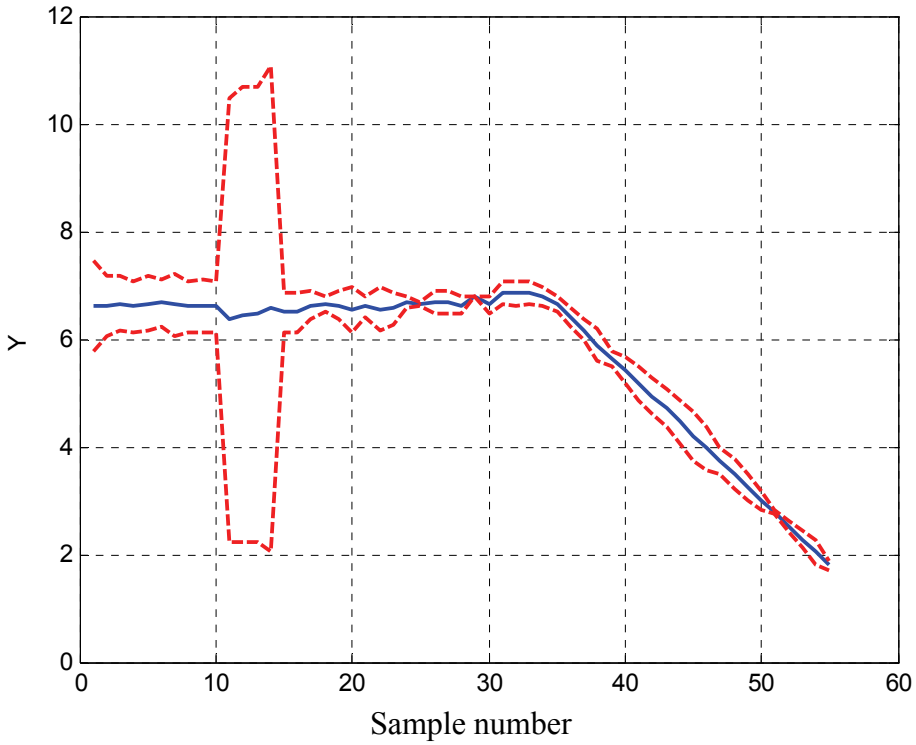


Fig. 3.16. The robot y_f fused output location.

According to the fused robot position shown in figure 3.17 and as compared to the predefined robot path in figure 3.14; we can see that the measured robot position is well represented in spite of some faulty measured value for the robot position provided as shown in the figure.

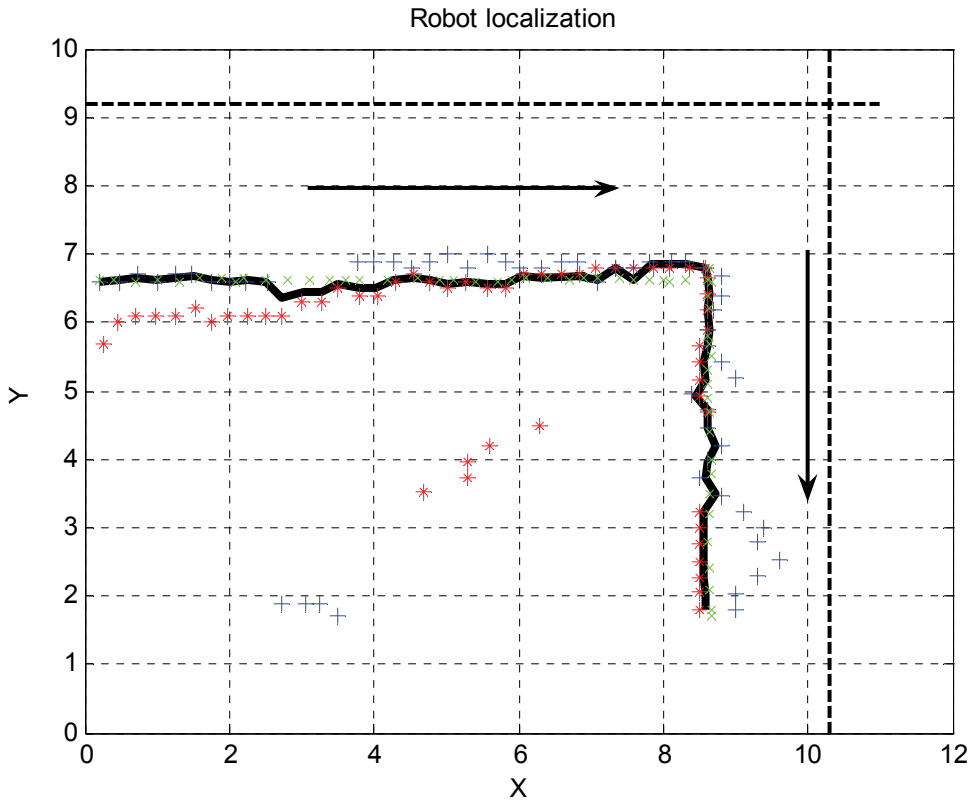


Fig. 3.17. The robot fused location (x_f, y_f) shown in solid line. The room layout shown in dashed lines, and the arrows show the robot path.

Examining figure 3.17, we can notice some of the measured values for the robot position deviated or drifted from the fused robot position, shown as the solid line in the figure, or from the predefined robot path shown in figure 3.14.

These drifted values for the robot position are detected as faulty measurements by robot sensors and reported to the display panel as shown in figure 3.18. As we can see from this figure, the distance measured values provided by the sonar sensor have three groups of faults detected, two of them in the measurement direction number *1*, and one in the measurement direction number *3*, according to figure 3.13.

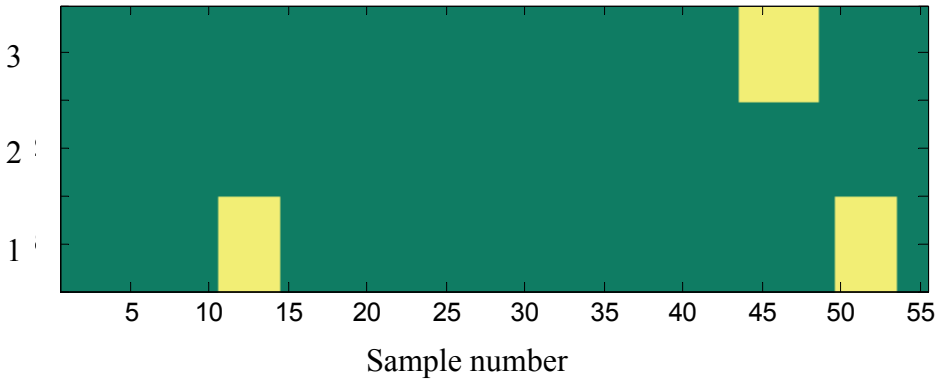


Fig. 3.18. The display panel for the detected faults shown as a yellow bar.

For the faults detected in the measurement direction number *1*, the first group is between measurements numbers 11-14 and the second faulty group is between measurements numbers 50-53. While the faults detected in the measurement direction number 3 are between measurements numbers 44-48.

These detected faults agree with the data demonstrated in figure 3.17, where a set of possible measurement faults could be due to the used sonar sensors in mobile robots, which ranges from specular reflections of the radiated sounds waves to wide propagation angle [74,77].

These two faults are arising from; firstly, the possible reflection of the emitted sound waves away from ahead object and fail to be detected by the sonar sensor again, and secondly from the fact that the propagated sound waves travel in cone shape in the surrounding robot media.

As indicated from the previous performance evaluation demonstrated results, it is clear that the hybrid system approach is able to provide the right fused mobile robot location during the experiment in a very agreeable way compared to the mobile robot predefined path through out the experiment time in spite of the detected faulty measurements collected by the robot sensors.

V. PERFORMANCE EVALUATION IV

Recently the desire for distributed wireless sensor networks has increased to solve a variety of problems in wide area of applications such as automation, control, monitoring, tracking, security, intelligent environments and health monitoring [89-93]. These networks have been called wireless sensor networks because they consist of a set of sensing nodes and processing elements distributed over the target field area, that are capable to communicate with each others with low data rate and low power constraint through a communication network [89].

Figure 3.19, explains the wireless sensor network structure. As it can be noticed from the figure, the sensing nodes can communicate with one or multiple processing elements simultaneously. The data collected by each sensing node is transmitted to the associated

processing element for processing and fusion. The processing elements can pass the collected information to higher processing elements for better cooperation and information integration and fusion [90]. The processing elements are shown as gray circles in figure 3.19 while the sensing nodes are shown as white circles.

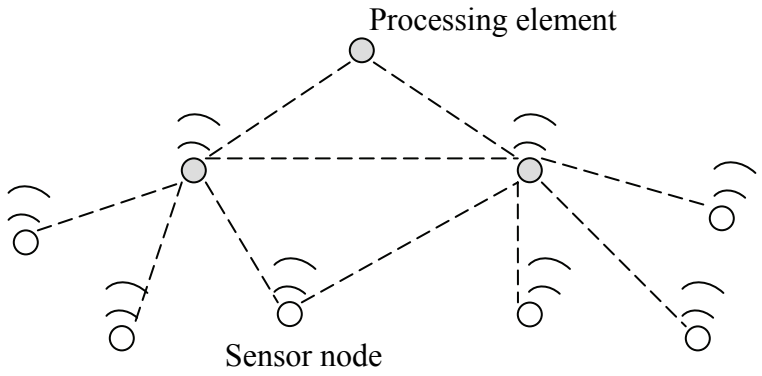


Fig.3.19. Wireless sensor network structure.

In this section the performance of the hybrid system is evaluated by using the collected data from the distributed wireless sensors network to provide the correct fusion and monitoring for indoor light intensity. For this purpose a wireless sensor network is implemented using TELOS–B experimental wireless sensor network platform which is available by Crossbow Technology [94] and runs the TinyOS open-source operating system which was developed and published by UC Berkley and supports a lot of sensor networks platforms.

The TELOS–B kit is a low power research development platform with integrated microcontroller, RAM, embedded antenna, radio-chip, onboard sensors, programming

ability through USB connector and IEEE 802.15.4 radio compliant 2.4GHz. The block diagram for the TELOS–B platform basic components and the TPR2420 kit are shown in figure 3.20 [94].

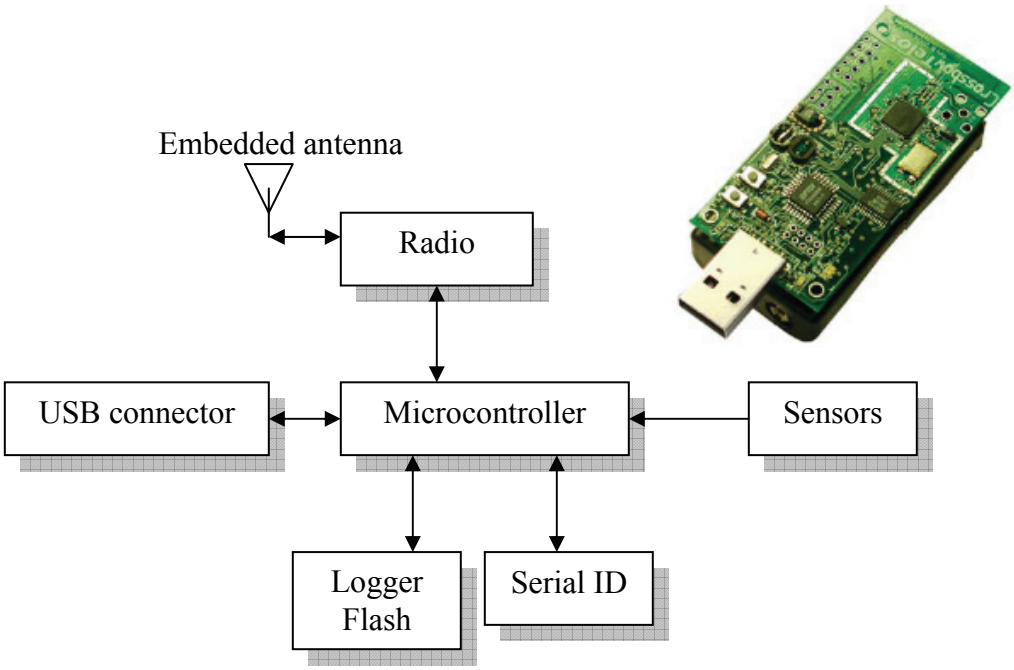


Fig. 3.20. TELOS–B platform basic components block diagram.

The deployed wireless sensor network for measuring the indoor light intensity consists of four distributed sensing nodes (TPR2420 kit) provides the light intensity measurement through the onboard light sensor to one base element (TPR2400 kit) through the wireless link, while the base element is connected to the computer as a gateway. Figure 3.21 shows the deployed wireless sensor network structure.

A network address is assigned to each sensing node module, and this process is required for the wireless communication between the sensing nodes and the gateway. Where each data packet will carry its source address which is the same as the network

address as a signature for each node, so the gateway can know which packet belongs to which sensing node. The data packet format is shown in figure 3.22, where the data packet consists of the following fields [95]:

- Destination address (2 bytes).
- Active Message handler ID (1 byte)
- Group ID (1 byte)
- Message length (1 byte)
- Source ID (2 bytes)
- Sample counter (2 bytes)
- ADC channel (2 bytes)
- ADC data readings (2 bytes each).

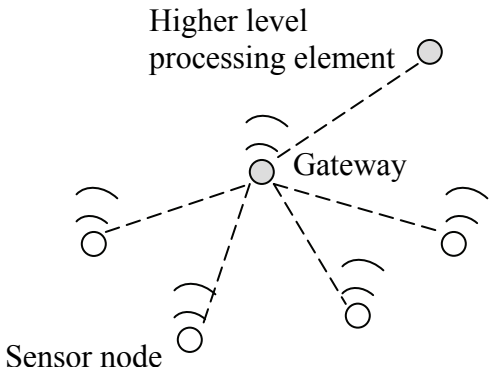


Fig.3.21. The deployed wireless sensor network structure.

Dist. addr	HandlerID	GroupID	Msglength	Source addr	Cont.	Ch	Readings

Fig. 3.22. The data packet format.

For this experiment the four sensing nodes were distributed within indoor room environment with constant light intensity. These sensing nodes transmit the collected data through the wireless network at a 250 Kbps data rate to the sensor network gateway. A fault was simulated by changing the light intensity around one of the sensing nodes by turning another light source on close to this sensing node after short time. The four sensors nodes (S_1- S_4) measurements plots are shown in figure 3.23.

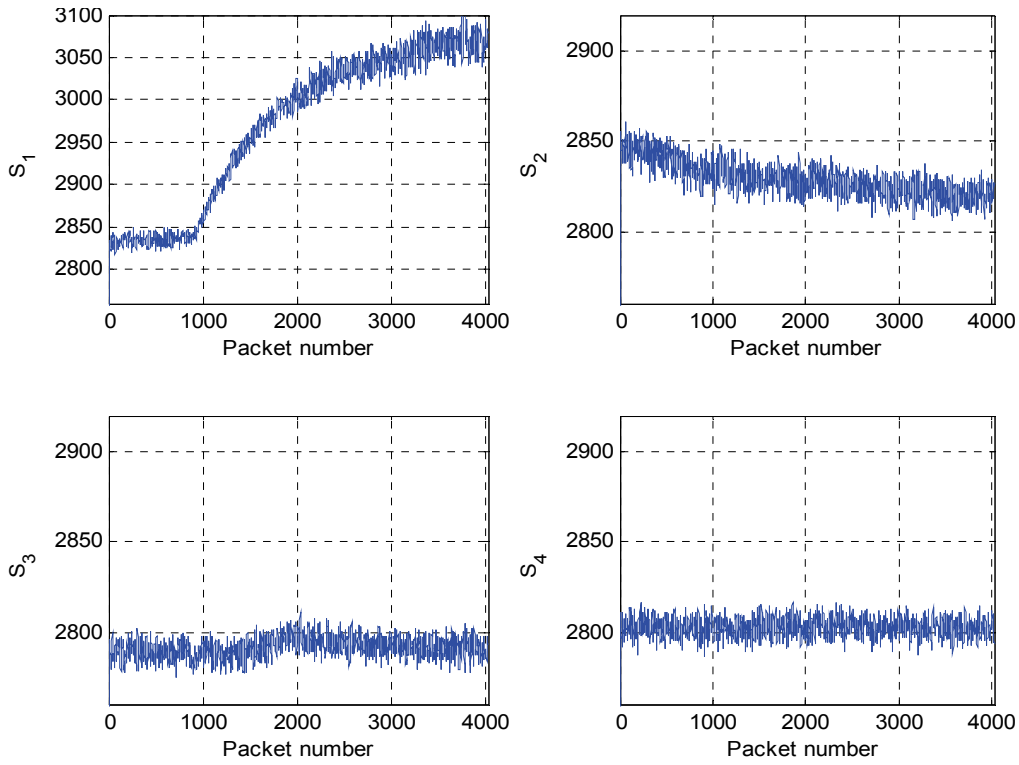


Figure 3.23. The four sensors nodes (S_1- S_4) measured values.

As it can be noticed from the figure, the light intensity measured values collected by sensor number one and reported to the gateway show a remarkable increase in the measured values after data packet number 1000. While the other sensors shows small

variability in the measured values during the experiment time as shown in sensor three and four plots, the second sensing node shows a slow variability for the measured light intensity.

The resulted fused output from the hybrid system for the wireless sensor network is shown in figure 3.24, as well as the fault display panel, and the required number of clusters plots.

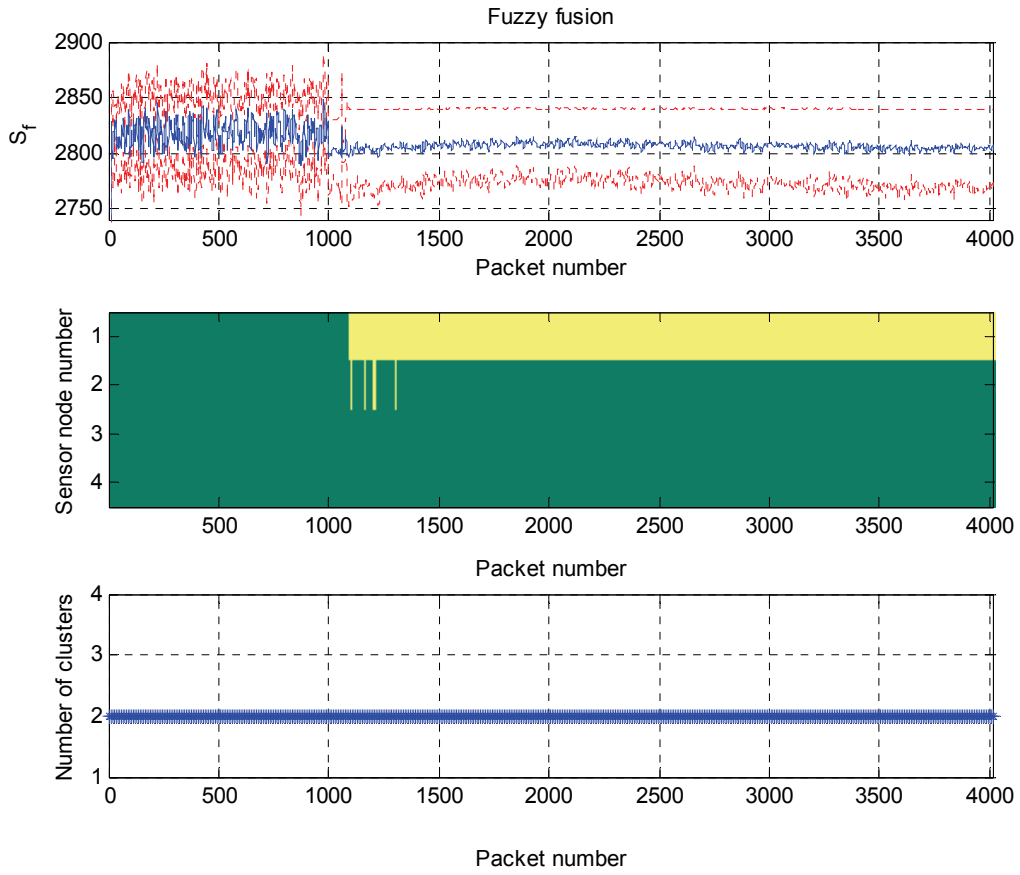


Fig. 3.24. The hybrid system resulting plots. The fused output, fault display panel and the required number of cluster plots.

The fused output denoted as S_f shown in solid blue line, while the detector upper and lower boundaries are shown in dashed lines. When comparing the resulted fused output with the collected measured values by the four sensing nodes in figure 3.23, we can see that the hybrid system is able to minimize the drifted measured values by sensor node number one S_1 in the final fused output giving the correct fusion in spite of this fault appearance.

The generated signals for the detected faults by the hybrid system are reported to the fault display panel; again the detected faults are presented as a yellow bar in this evaluation. By examining the fault display plot in figure 3.24, a fault is clearly detected after packet number 1000 belongs to sensing node number one S_1. This agrees with the simulated fault for sensor number one, during the experiment time. Comparing sensing node number one plot in figure 3.23 with the detected fault in the display panel, we can see that the drifted measured values by this sensing node is clearly noticed after packet number 1000, so the reported fault signals agree with sensor measurements faulty trend started after packet number 1000.

The required number of clusters by the hybrid system is shown in the third plot in figure 3.24, two clusters are required by the hybrid system as it is noticed from the plot and this agrees with the direction of the deviated measured values by sensing node number one S_1, and the rules used to determine the right number of clusters required by the hybrid system.

The α values calculated for the binary code generation process are shown in figure 3.25, as it can be noticed from the plot the values are less than one which agrees with the membership values definition to be between zero and one.

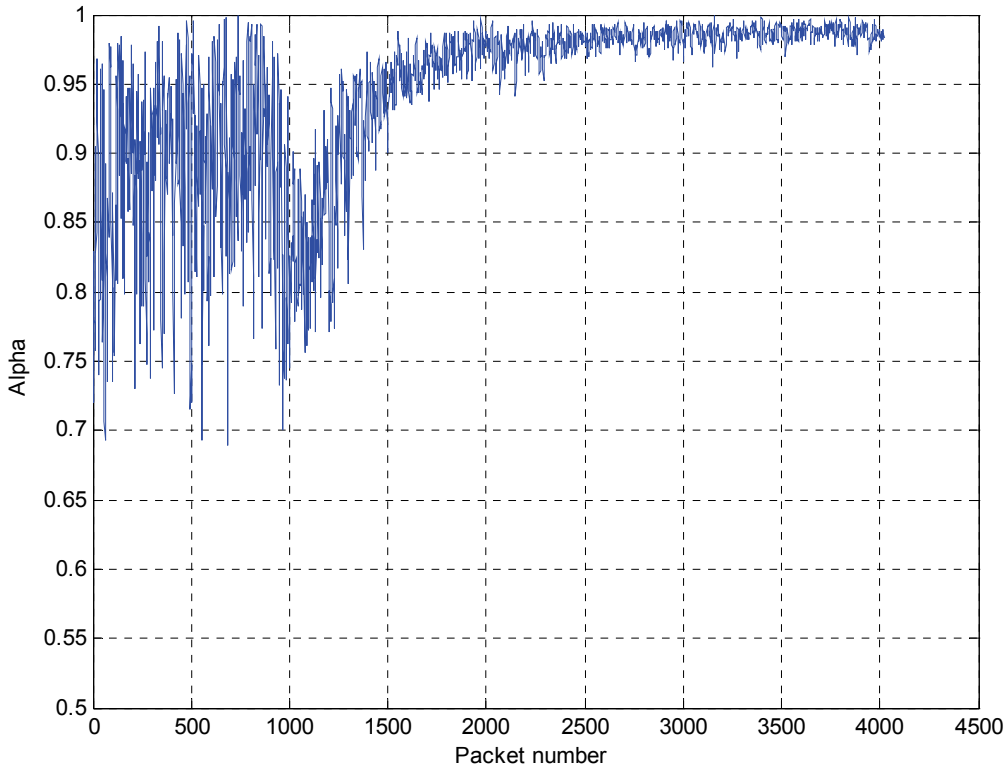


Figure 3.25. The α values.

From the demonstrated results for this experiment, it can be seen, as it was expected, the presented hybrid system is able to minimize the deviated measurement values effect in the final fused output and detect the fault source through the AIS detector in a reliable way.

CHAPTER IV

EARLY FAULT DETECTION SYSTEM

I. INTRODUCTION

Our objective on this chapter is to develop a fault detection system for critical dynamic components of rotorcraft and in particular for the H-60 Intermediate Gearbox (IGB). With this in mind, the presented work discusses this objective based on the algorithm utilized by the hybrid system implemented in the previous chapters.

This chapter is organized as follows; a preliminary introduction for this topic is discussed in section two, the hardware platform that is the subject of this study is briefly covered in section three, the developed methodology is demonstrated in section four followed by the obtained results in section five which shows the efficiency of the developed methodology.

II. PRELIMINARIES

The majority of diagnostic systems for complex mechanical systems have focused on determining the onset of critical conditions requiring maintenance actions [78-84]. Detection of early signs of failure, i.e. incipient fault conditions, however, can lead to more effective maintenance procedures. Moreover, early detection of incipient faults, formally called prognostics [85], can help prevent catastrophic failures and enhance operational reliability as shown in figure 4.1.

Prognostics generally contain two phases. The first phase focuses on the detection of the fault state, while the second phase attempts to determine the remaining useful life of the component of the system.

With this in mind, we will use the utilized hybrid system implemented in the previous chapters for early detection of fault condition based on studying a group of selected features to utilize our objective.

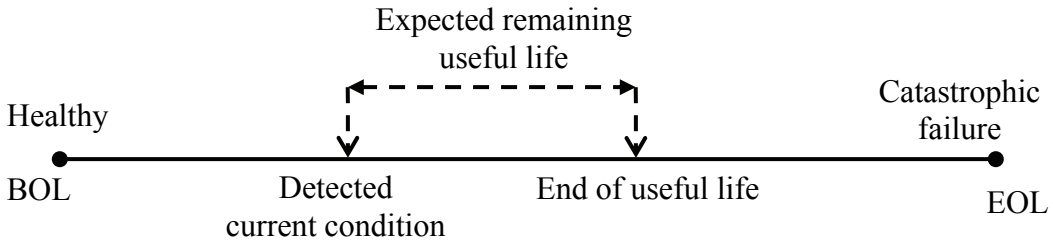


Fig. 4.1. Timeline for component prognostics.

III. HARDWARE PLATFORM

For the evaluation process an H-60 gearbox is considered. The data in this study is collected from the H-60 Intermediate Gear Box (IGB), as shown in Figure 4.2. The data set is provided by NAVAIR and includes traces that span from healthy to failure conditions. Figure 4.2 shows the tail drive system, where the accelerometers are located

at the input/output of the Intermediate Gear Box (IGB) to collect the vibration data, the IGB section is shown enlarged in figure 4.2 for more details [78,85,86].

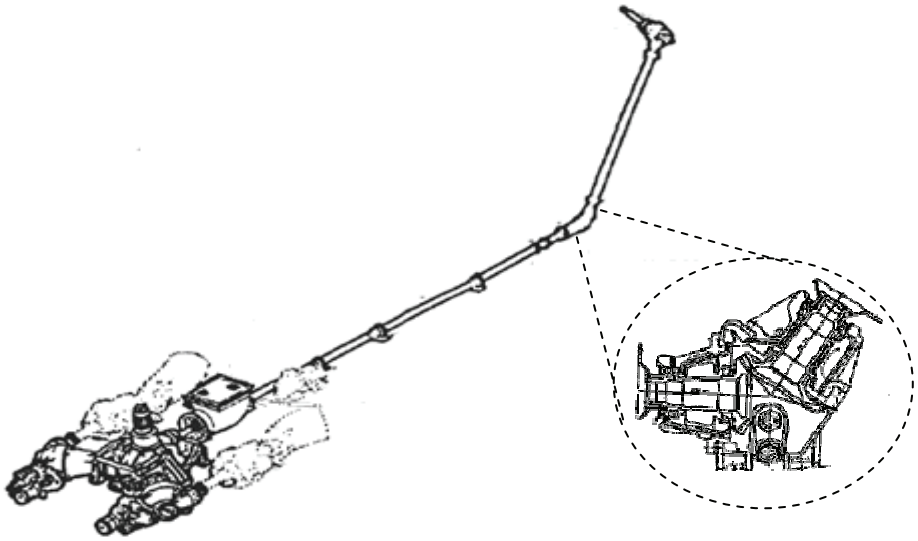


Fig. 4.2.The H-60 intermediate gear box (IGB) [78].

The test was run at full tail power, and terminated before the failure has completely occurred in the Intermediate Gear Box (IGB). The collected data was planned to be collected at the same steady state condition through out the test time. The data was collected at 100 KHz, a total of 89 records were collected spanning the test time from healthy to failure component conditions, 30 second for each recording time separated by 15 minute interval between any two subsequent recordings. A five second sample was used from each record in the following steps, which will be discussed shortly in the methodology section [85,87].

IV. METHODOLOGY

The hybrid system presented and discussed in the previous chapters will be used to detect the possible faults that could occur in H- 60 Intermediate Gearbox (IGB) based on the collected accelerometers data.

A selected group of features are used to characterize the collected test data, after that these extracted features are applied to a hybrid system for fault detection, to get the right decision for fault detection, based on the similarity/dissimilarity measures between the three extracted features, as illustrated in figure 4.3.

Figure 4.3 describes the hybrid system blocks, where the selected features are applied to the fuzzy partitioning block where the fused feature trend is generated to the AIS fault detection part for fault monitoring process.

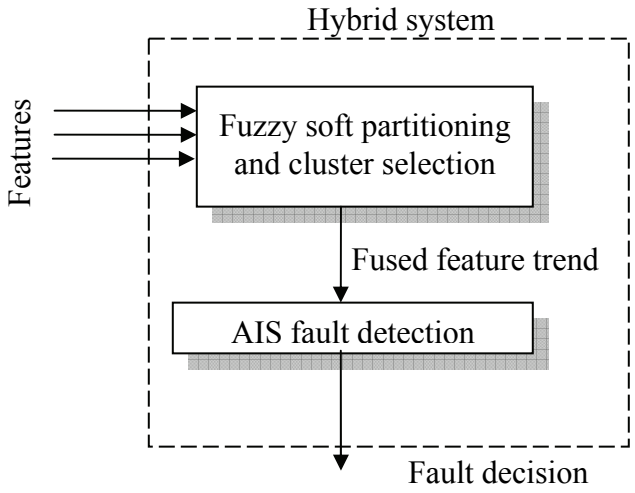


Fig. 4.3. Hybrid system block diagram using the extracted features as an input.

In general one or more preprocessing steps on the collected raw sensor data is required to extract useful characteristics or features that are necessary to implement the fault detection system.

In particular, for each data trace or record a time synchronous average is computed, by synchronizing the sampling of the vibration signal with the rotation speed of the main shaft and evaluating the average over many revolutions framed at the same angular position. The resulting signal is called time-synchronous averaging and denoted as *TSA*. This initial step reduces the effect of sources of noise and vibrations [85,88].

Following the computation of the time synchronous average, a residual signal is computed for each trace signal by eliminating the normally occurring frequencies and computing the inverse Fourier transform. As a result, the Time-Synchronous Averaging Residue (TSAR) signal is generated and used as the basis for selecting useful features for fault diagnosis [85,88].

Three features are used to characterize the resulting TSAR signal for each sample; the features [85-87]:

- RMS value of the signal defined in equation 4.1, this value gives an indication to the magnitude of the defect, and it is proportional to the energy contents in the signal :

$$RMS = \frac{\| x \|}{\sqrt{N}} \tag{4.1}$$

- The residual peak to peak X_{pp}, which gives a good indication to the peak vibration level changes for the given signal X, and the X_{pp} value is defined as follows:

$$X_{pp} = \max(X(t)) - \min(X(t)) \tag{4.2}$$

- Kurtosis which is the fourth statistical moment of the signal, given as follows:

$$K = \frac{\sum (x-\mu)^4}{N\sigma^4} \tag{4.3}$$

Where μ is the mean of the data set, σ is the variance and N is the number of data points. Kurtosis is a measure of how outlier-prone a distribution is.

A mapping step is required by scaling the other features according to the kurtosis feature and the mean difference as follows:

$$feature_i = sclaed \quad feature_i + \Delta. \tag{4.4}$$

Where Δ is the mean difference between the scaled feature mean value and the kurtosis mean value.

Next, the three extracted features are applied to a hybrid system for fault detection, to get the right decision for fault detection, based on the similarity/dissimilarity measures between the three extracted features. The fault detection process is illustrated in the block diagram shown in Figure 4.4.

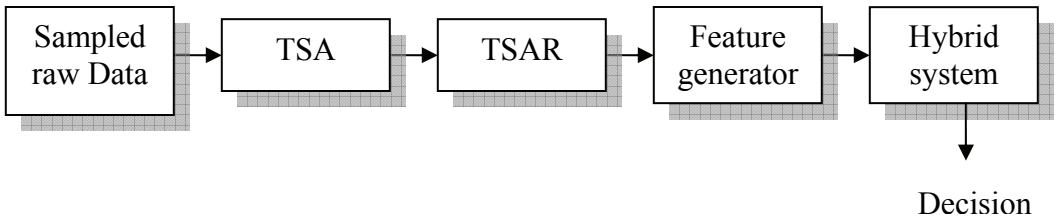


Fig. 4.4. Fault detection methodology.

The hybrid system consists of two main parts. In the first part a single decision is derived by monitoring the three extracted features, this decision is generated using the fuzzy clustering scheme. While in the second part an Artificial Immune based System (AIS) is used to generate the fault detection decision based on the fused output from the first part.

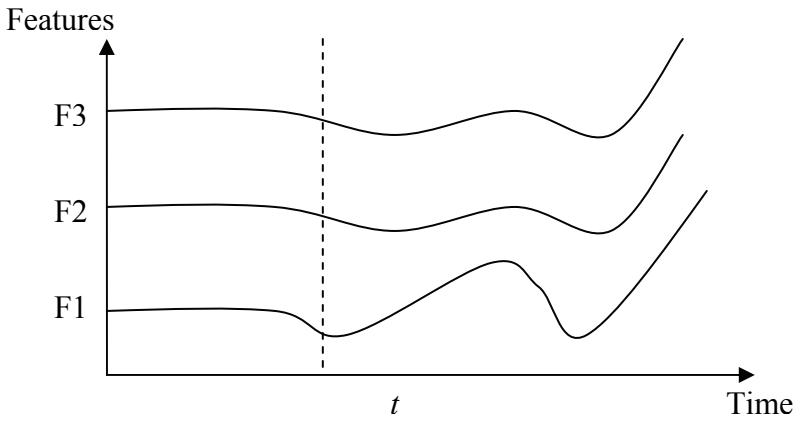


Fig. 4.5. The three extracted features trends over runtime.

As illustrated in figure 4.5, at time instant *t*, due to early fault condition, the normalized three extracted features for Time-Synchronous Averaging Residue signals will not be the same. The extracted features can have the fuzzy cluster partitioning relation shown in figure 4.6.

Fig. 4.6. Soft partitioning for the selected features at time instant t.

The fuzzy clustering will be used to extract a fusion trend for the used features in the figure above, where the fused features trend is generated by the fusion engine as discussed before in chapter II.

After that, the Artificial Immune negative selection mechanism is used to monitor the system for early fault detection based on self/non-self selection. This is mapped as normal fault free system behavior/ faulty abnormal system behavior as we mentioned before.

A fault detector is defined based on the fused feature trend considered as the self state fault free system behavior. The hybrid system will monitor the system through the features trends. It will detect any non-self match case outside the detector range to indicate a fault condition is detected by generating a fault signal and pass it to the fault display panel to present the detected abnormality in the system; the complete process is demonstrated in figure 4.7.

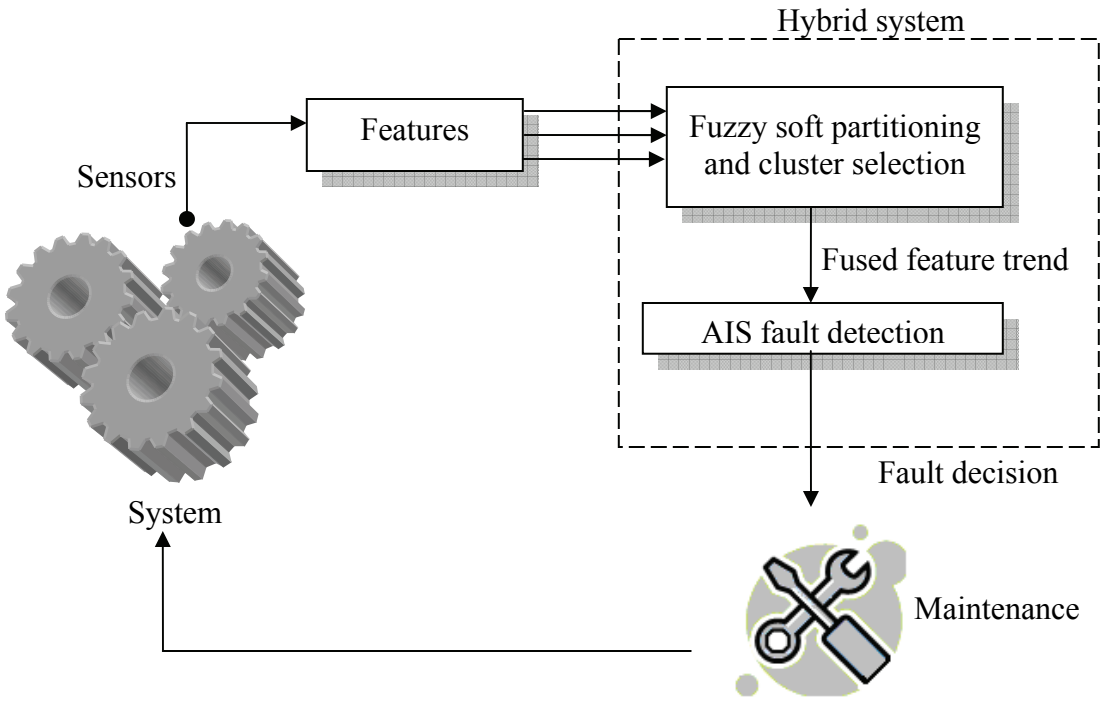


Fig. 4.7. Fault detection using the hybrid system approach.

V. EXPERIMENTAL RESULTS

The proposed procedure above is applied to a data set provided by NAVAIR. Figure 4.8 shows a five seconds sample taken from three-sample traces in the data set. These are identified as Trace or Record number 1, which is from the beginning time of collecting the data, Trace or Record number 45 from the middle and Trace or Record number 89 at the end time.

The time-synchronous averaging signals are shown in Figure 4.9 for the three recordings. As it can be noticed from comparing the three plots, a change in the vibration pattern is noticed when moving from recording number 1 to recording number 89. Also

the plots are more readable than the data demonstrated in figure 4.8, where each plot shows only one gear cycle, but these plots still lack a very good vibration pattern representation.

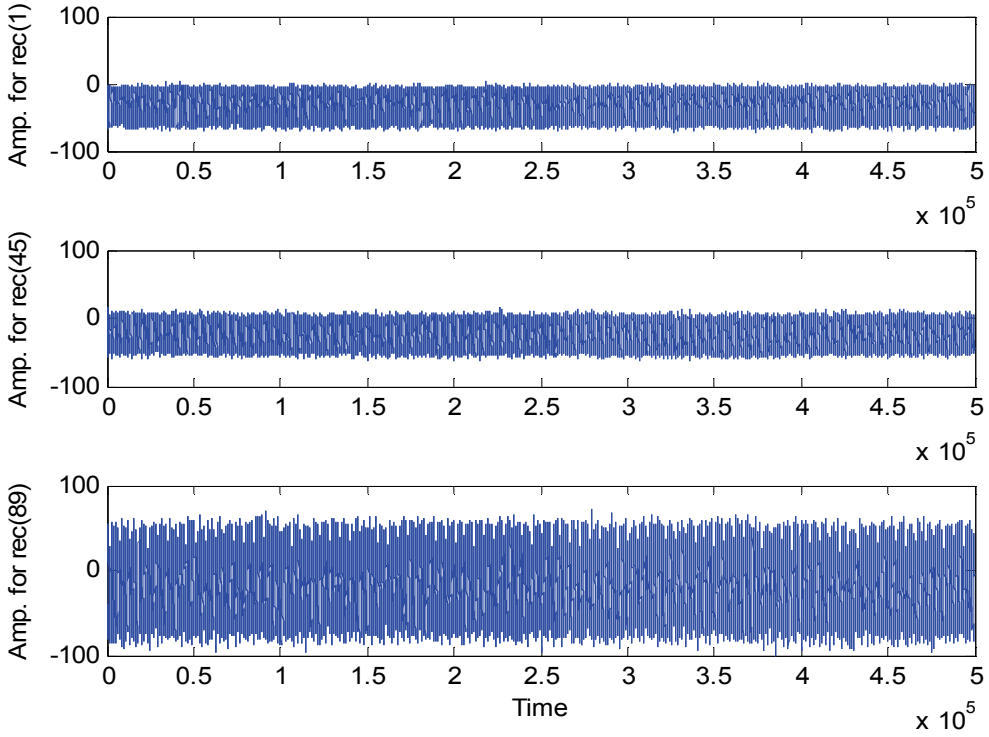


Fig. 4.8. Five second samples for Traces #1, # 45 and #89.

The amplitude spectrums of the time-synchronous averaging signals are shown in Figure 4.10. For better demonstration; the three plots are zoomed in both axes, where notable changes in the amplitude spectrums pattern are getting clearer by moving toward record number 89. Figure 4.11 shows the time-synchronous averaging residue signals plots for the same three samples.

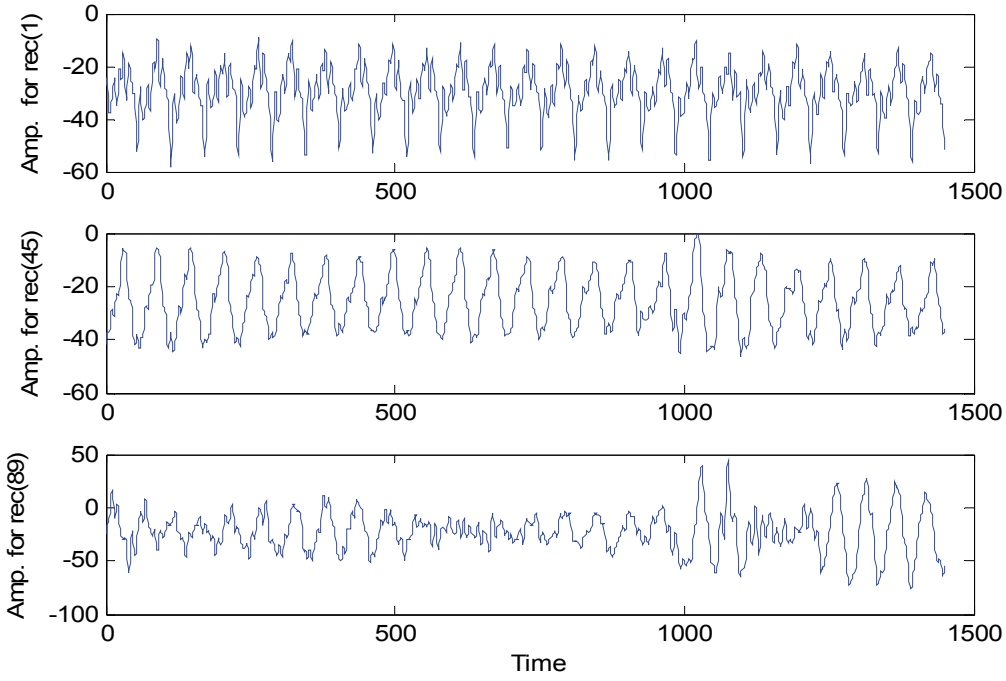


Fig. 4.9. TSA signals for Traces #1, #45 and #89.

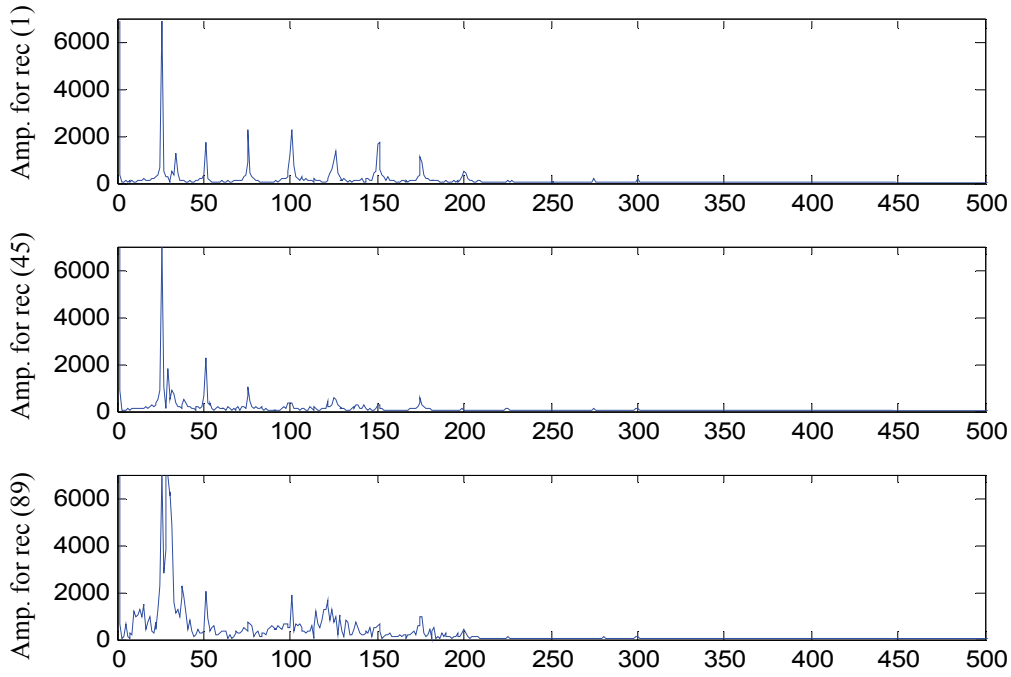


Fig. 4.10. Amplitude spectra for Traces #1, #45 and #89.

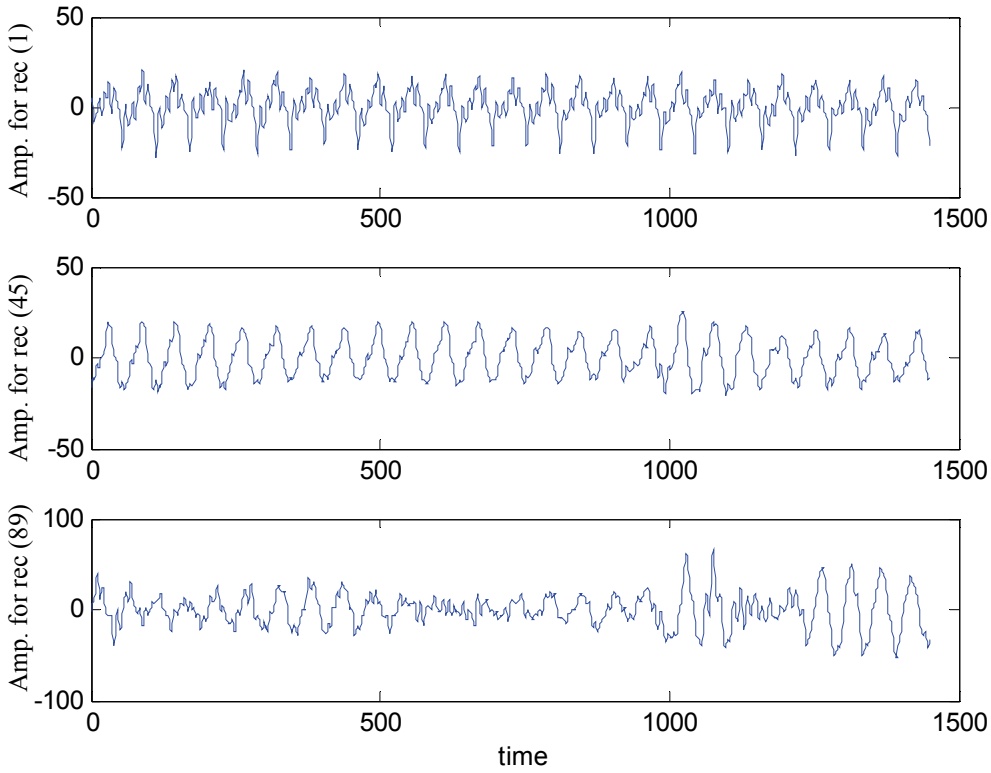


Fig. 4.11. TSAR signals for Traces #1, #45 and #89.

As it is clear from figure 4.11, the vibration pattern changes are clearer in the three plots for records number 1, 45 and 89. From the last plot (record number 89), a failure or crack damage can be expected according to the notable variability in TSAR amplitude values.

This detected variability in the amplitude values can give an expectation to the failure or the damage location, the time-synchronous averaging residue signals represents one revolution of the collected data as mentioned before, so the expected fault can be more located in the last (third) part of the tested gears.

The three extracted features plots for the entire sampled recordings are shown in Figure 4.12, which characterizes the time-synchronous averaging residue signals. Through examining these plots in the figure we can have several remarks for the three features RMS, peak to peak, and the Kurtosis trends from the initial recording toward the end of the test recordings.

By looking at the first fifty records, the same slow variation trend in the three features value can be easily noticed until record number 30, after that another level of slow variability is taking place until record number 49. In the sequel forty-nine records another remarkable increasing pattern with different variability between the features is noticed clearly.

For instance, if we have a deep examination of the Kurtosis of the TSAR signals plot, we found that, initially between records number 1-49, it was almost stable with very small variability around the value of 2. After that the Kurtosis started to increase in a remarkable way to reach a peak value at about 4.75 between records numbers 50-76 before it breaks down at record number 77.

While the other two features: the RMS feature and the peak to peak feature, shows the same stability with a slight variability between records numbers 1 to 29, after that they show a slight variability at another level between records numbers 30-50.

For the last forty-nine records, after recording number 50, both of the two features show a slow response starting at record number 50, where the Kurtosis started to increase remarkably, they show a slow increase after that, and ramped up as the fault propagated more to failure until record number 89.

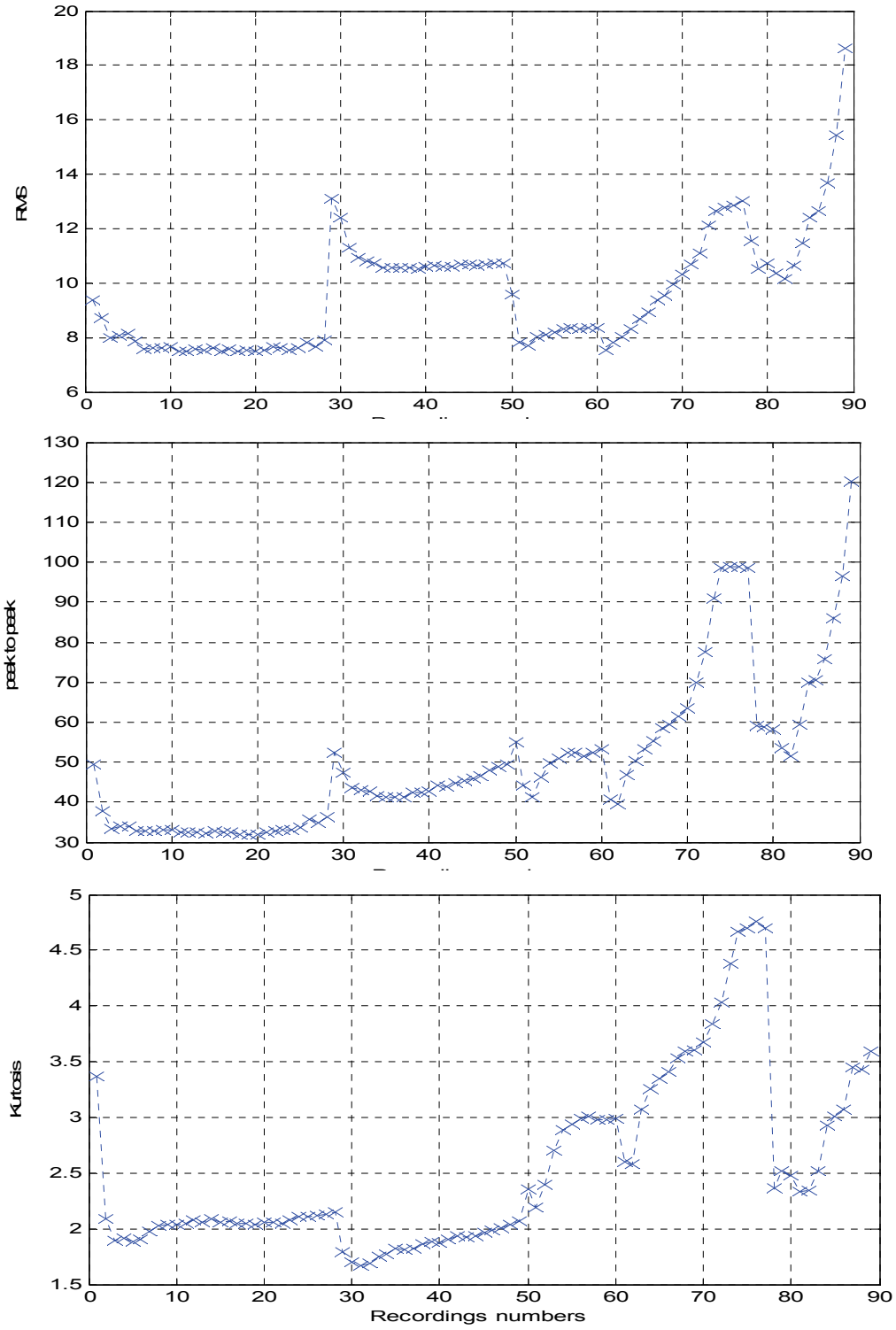


Fig. 4.12. The three extracted features plots: RMS, peak to peak, and the Kurtosis.

In spite the fact that the RMS feature and the peak to peak feature show the same slow increase compared to the Kurtosis feature , the peak to peak feature shows better and faster response than the RMS feature at these recordings when they are compared together.

These trends in the three features before record number 50 and after it are due to the changes in the torque values [87]. Initially the torque was increased during the first part of the recording process, from record number 1 to record number 49, to accelerate the failure or the damage occurring during the test [87]. The load was increased on average between the values 2350-2377 ft-lb, until record number 50 where a notable change occurred in one of the three features, which is the Kurtosis of the Time-Synchronous Averaging Residue (TSAR) signals for each record, as shown in figure 4.12.

After this notable change, the load was decreased to take a value between 2022-2045 ft-lb on average, so that the damage occurred is propagating at a steady state condition as it was planned before collecting the data [87].

The results after applying the three normalized features shown in Figure 4.13, to the hybrid system are shown in Figure 4.14. The fused feature trend (F_f), the detector upper and lowers boundaries are shown in the first plot. The detected faults are displayed in a display panel which shows the time progress for the monitored system, displaying the instant at which an early fault or damage is detected denoted in yellow over the time progress bar, while the green areas in the display panel denote that no fault was detected. As it is noticed from the figure the early fault was detected before the final damage or failure occurs.

Figure 4.13 shows the three normalized features plots; RMS denoted as F_1, peak to peak denoted as F_2, and the kurtosis denoted as F_3. The general trend for the three features can be noticed covering the condition from healthy to failure over the time progress. The general features trend can be describe as flat regions before record number 50, which progressed after that to increases rapidly at the failure stages toward the end of the of the data samples or records.

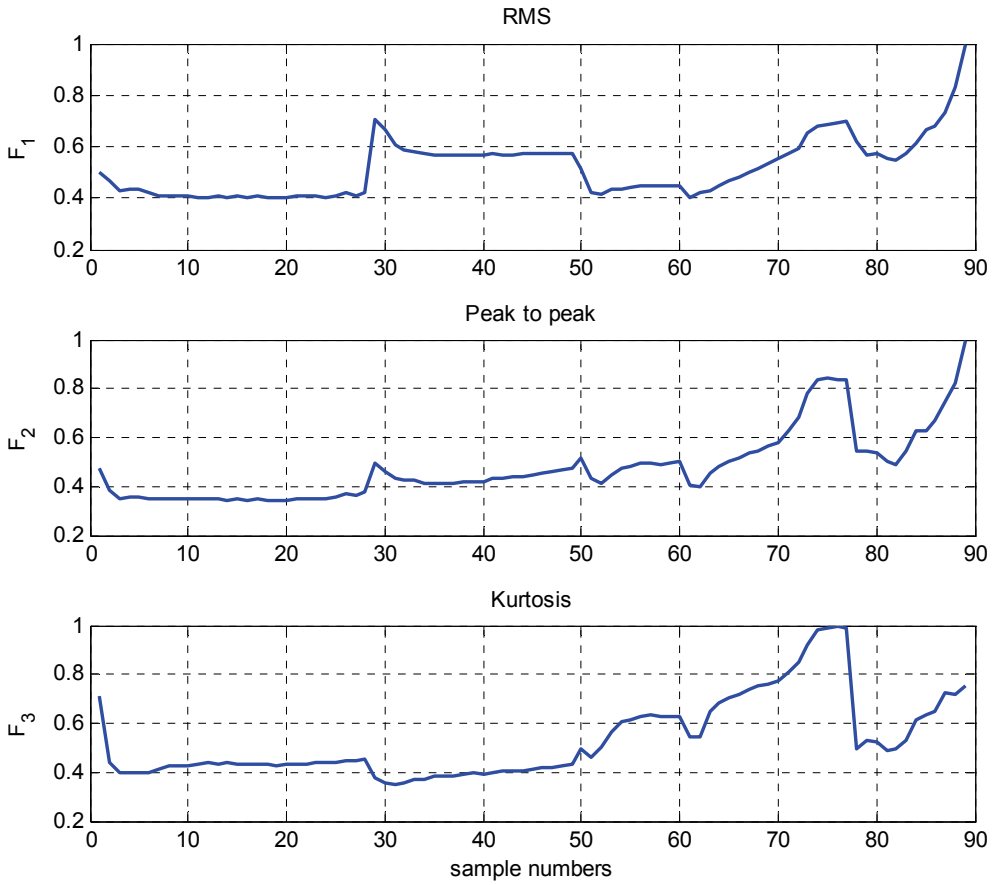


Fig. 4.13. Three normalized extracted features.

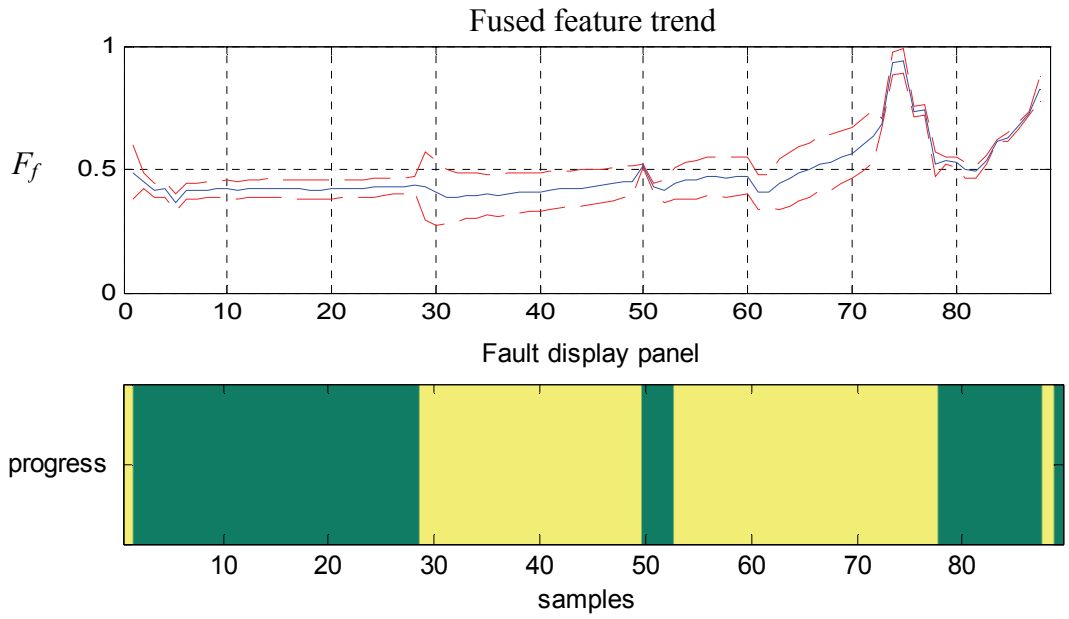


Fig. 4.14. Fused feature trend and the Fault display panel.

The kurtosis feature F_3 gives a good indication for early fault conditions before the failure occurs, where it started to rise smoothly at early stages of the test as it can be noticed after record number 50, more than the other features.

The displayed fault output in the display panel figure 4.14, agrees with the early changes in the three features discussed before, and gave an early fault signal indicating a fault condition was detected. As we can see, the detected faults presented over the time progress in figure 4.14 can be separated in two groups, before and after sample or record number 50.

For the first group, as it was mentioned before the test was planned to run at steady state condition constant torque to collect the data, the increase in the torque was required

to accelerate the early damage occurrence. This torque increase is obvious in the RMS plot of the TSAR signals figure 4.13 between records number 30-49, where a step can be noticed in this plot at this interval of the samples. Due to this change in the torque, the first fault group was detected as demonstrated in figure 4.14.

The second group of fault was detected when the damage occurred and propagated to failure at a steady state condition as it was planned before for the test conditions. As it can be noticed clearly from the time progress bar of the monitoring system, the early fault was detected at record number 53 and continues until record number 77, before the final failure is detected at record number 88.

Comparing these results to the three features used a very good similarity between the feature trends and the detected faults are noticed. As a result the hybrid system is able to provide a very good mean to detect the early conditions of failure which can propagate to component damage or failure over time.

CHAPTER V

CONCLUDING REMARKS AND FUTURE WORK

I. INTRODUCTION

This chapter has two main parts, in the first one; a brief summary for the provided work is presented, while in the second part the concluding remarks from this study are presented as well as the flow for future work.

II. SUMMARY

In this study, we have presented an efficient new hybrid approach for multiple sensor data fusion and fault detection based on the fuzzy c-means (*FCM*) clustering algorithm and the artificial immune systems negative selection mechanism (AIS), addressing the case with multiple faults in different directions.

In the first part of this study a literature review was provided through the preliminary introduction in chapter I, after that the hybrid approach implementation was detailed in chapter II and the required backgrounds were discussed too. The hybrid system performance evaluations were subsequently demonstrated through the simulated results as well as experimental results in chapter III. In this chapter the hybrid system performance was compared and evaluated to other fusion and fault detection approaches.

In chapter IV an early fault detection system was developed based on the implemented hybrid system approach, where the experimental results from applying the hybrid system showed a very good performance.

III. CONCLUDING REMARKS AND FUTURE WORK

An efficient new hybrid approach for multiple senor data fusion and fault detection was presented in this study; this new approach was developed based on the fuzzy C-means (*FCM*) clustering algorithm and the artificial immune systems negative selection mechanism (AIS), addressing the case with multiple detected faults.

For this new hybrid approach, the first part of it provides a multiple sensor data fusion passed on the fuzzy logic soft partitioning capabilities to represent the input data. The right fused output was generated through the fusion engine according to the clustering information provided by the fuzzy c-means clustering algorithm.

The rules used to determine the right number of clusters during the fusion process according to the drifted measured values direction is working as good as expected from the simulated results.

In the second part of the hybrid system, a new fault detection based on the artificial immune systems negative selection mechanism was presented. The new hybrid approach is an unsupervised approach for both multiple sensor data fusion and fault detection where no learning or training phases are required.

The presented simulation and the conducted experiments for multiple sensor systems have confirmed the strength of the new approach for online fusing and fault detection by handling different problems such as noisy sensor signals and multiple faulty sensors, where the hybrid system gives a fault tolerance (i.e. the system ability to keep working and providing an output in spite of the presented faults). This makes the new hybrid approach attractive for solving such fusion problems and detecting faults during real time operation.

The hybrid system was extended for early fault detection in complex mechanical systems as in rotorcraft intermediate gearbox system. The hybrid system was able to detect the onset of fault conditions which could lead to critical damages or failures. This early detection for the failure signs can provide more effective information for any maintenance actions or corrective procedures.

From the presented experimental results; we can see that the hybrid system can be applied to many systems which open the way for further work to apply the presented approach in many areas such as image processing and distributed sensor network for monitoring purposes.

For the early fault detection more development for the system can be done, so that the system can detect the starting of any fault condition and an estimation for the useful remaining life time (i.e. prognostics), which can provide more effective information for any maintenance decision needed afterwards.

REFERENCES

[1] T. Runkler, M. Sturm, and H. Hellendoorn, "Model based sensor fusion with fuzzy clustering," in *Proceedings of the 1998 IEEE International Conference on Fuzzy Systems, IEEE World Congress on Computational Intelligence*, Anchorage, Alaska, USA, vol. 2, 1998, pp.1377-1382.

[2] H. Durrant-Whyte, "Elements of sensor fusion," in *IEE Colloquium on Intelligent Control*, London, 1991, pp. 5/1 – 5/2.

[3] H. David and L. Jamas, eds., *Handbook of Multisensor Data Fusion*, CRC Press, New York, 2001.

[4] R. Luo and M. Kay, "A tutorial on multisensor integration and fusion," in *16th Annual Conference of IEEE Industrial Electronics Society IECON '90*, Pacific Grove, California, USA, 1990, pp.707-722.

[5] R. Luo, C. Yih and K. Su, "Multisensor fusion and integration: approaches, applications, and future research directions," *IEEE Sensors Journal*, vol. 2, pp.107-119, 2002.

[6] C. Lee and Y. Xu, "Theoretical study on a new multi-sensor system," in *Proceedings of the First ISA/IEEE Conference Sensor for Industry*, Rosemount, Illinois, USA, 2001, pp.187-191.

[7] M. Abderahman and P. Kandasamy, "Integration of multiple sensor fusion in controller design," in *Proceedings of the American Control Conference*, Anchorage, Alaska, USA, vol. 4, 2002, pp. 2609-2614.

[8] Y. Vershinin, "A data fusion algorithm for multisensor systems,"in *Proceedings of*

the Fifth International Conference on Information Fusion, Annapolis, Maryland, USA, vol. 1, 2002, pp. 341-345.

[9] S. Suranthiran and S. Jayasuriya, "Nonlinear averaging of multi-sensor data," in *ASME International DETC'03, Proceedings of the 19th International Biennial Conference on Mechanical Vibration and Noise (VIB)*, Chicago, Illinois, USA, 2003, pp. 1/48512- 6/48512.

[10] S. Akkihal and G. Wise, "Data fusion with a faulty sensor," in *Proceedings of the American Control Conference*, Seattle, Washington, USA, vol.1, 1995, pp.638 – 642.

[11] P. Odgaard, J. Stoustrup, P. Andersen and H. Mikkelsen, "Estimating focus and radial distances, and fault residuals from CD player sensor signals by the use of a kalman estimator," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii, USA, vol. 2, 2003, pp. 1962 – 1967.

[12] Q. Gan and C. Harris, "Comparison of two measurement fusion methods for Kalman filter based multisensor data fusion," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, pp. 273 – 279, 2001.

[13] A. Al-Dhaher and D. Mackesy, "Multi sensor data fusion architecture," in *Proceedings of the 3rd IEEE International Workshop on Haptic, Audio and Visual Environments and Their Applications, HAVE 2004*, Ottawa, Canada, 2004, pp. 159 – 163.

[14] S. Niwa, T. Masuda and Y. Sezaki, "Kalman filter with time variable gain for multi sensor fusion system," in *Proceedings of the International Conference on Multisensor Fusion and Integration for Intelligent Systems, MFI '99*, Taipei, Taiwan, 1999, pp. 56 – 61.

[15] J. Sasiadek and P. Hartana, "Sensor data fusion using Kalman filter," in *Proceedings of the Third International Conference on Information Fusion*, Paris, France, vol. 2, 2000, pp. 5/19 - 5/25.

[16] A. Mahajan, K. Wang, and P. Ray, "Multisensor integration and fusion model that uses a fuzzy inference system," *IEEE/ASME Transactions on Mechatronics*, vol. 6, no. 2, pp. 188-196, 2001.

[17] M. Lee, K. Stanley, and Q. Wu, "Implementation of sensor selection and fusion using fuzzy logic," in *IFSA World Congress and 20th NAFIPS International Conference,* Vancouver, Canada, vol. 1, 2001, pp.328-333.

[18] D. Shahmirzadi, C. Lucas, and R. Langari, "Intelligent signal fusion algorithm using BEL-Brain emotional learning," in *Proceedings of Brain-Like Computer Architecture,7th Joint Conference on Information Sciences JCIS'03*, Cary, North Carolina, USA, 2003, pp. 1743-1746.

[19] P. Prajitno and N. Mort, "A fuzzy model-based multi-sensor data fusion system," in *Proceedings of SPIE*, vol. 4385, Orlando, Florida, USA, 2001. Sensor Fusion: Architectures, Algorithms, and Applications V, pp. 301-312.

[20] L. Perlovsky, "Mlans neural network for sensor fusion," in *Proceedings of the IEEE 1993 National Aerospace and Electronics Conference, NAECON 1993*,

Dayton, Ohio, USA, 1993, pp. 880 – 884.

[21] O. Postoache, P. Girao, H. Ramos and J. Pereira, "A temperature sensor fault detector as a artificial neural network application," in *9th Mediterranean Electrotechnical Conference, MELECON 98*, vol. 1, Tel-Aviv, Israel, pp. 678 – 682, 1998.

[22] S. Pui and A. Dexter, " A fuzzy approach to fault diagnosis in the presence of sensor bias," in *The 10th IEEE International Conference on Fuzzy Systems*, vol. 3, Melbourne, Australia, 2001, pp. 1585 – 1588.

[23] E. Gaura and M. Kraft, "Are neural network techniques the solution to measurement validation. Monitoring and automatic diagnosis of sensor faults?," in *Proceedings of the 41st SICE Annual Conference, SICE 2002*, vol. 3, Osaka , Japan, 2002, pp. 2052 – 2057.

[24] P. Ambrosio and N. Mort, "A hybrid Kalman filter-fuzzy logic architecture for multisensor data fusion," in *Proceedings of the IEEE International Symposium on Intelligent Control (ISIC '01)*, Mexico City, Mexico, 2001, pp. 364-369.

[25] S. Simani, C. Fantuzzi, and S. Beghelli ,"Identification and fault diagnosis of nonlinear dynamic processes using hybrid models," in *Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 3, Sydney Australia, 2000, pp. 2621-2626.

[26] L. David and A. Sonya, *Mathematical Techniques in Multisensor Data Fusion*, Boston: Artech House, 2004.

[27] S. Muldoon, M. Kowalczyk and J. Shen, "Vehicle fault diagnostics using a sensor

fusion approach," in *Proceedings of IEEE Sensors*, vol. 2, Orlando, Florida, USA, 2002, pp. 1591 – 1596.

[28] A. Vemuri and M. Polycarpou, "On the use of on-line approximators for sensor fault diagnosis," in *Proceedings of the American Control Conference*, vol. 5, Philadelphia, Pennsylvania, USA, 1998, pp.2857-2861.

[29] B. Yang, P. He, and B. Wang, "Research on fuzzy model and algorithms for data fusion," in *Proceedings of the Second International Conference on Machine Learning and Cybernetics*, vol. 4, Xian, China, 2003, pp. 2480 – 2484.

[30] A. Khan and M. Zohdy, "A genetic algorithm for selection of noisy sensor data in multisensor data fusion," in *Proceedings of the American Control Conference*, Albuquerque, New Mexico, USA, 1997, pp. 2256-2262.

[31] R. Siegwart and I. Nourbakhsh, *Introduction to Autonomous Mobile Robots*, Cambridge, Massachusetts: MIT Press, 2004.

[32] Active robots, AmigoBot mobile robot platform, available: http://www.activrobots.com/ROBOTS/index.html, 2005.

[33] L. Zadeh, "Fuzzy sets," in *Fuzzy Models for Pattern Recognition: Methods That Search for Structures in Data*, edited by J. Bezdek and S. Pal, New York: IEEE Press, 1992, pp. 35- 45.

[34] R. Hampel, M. Wagenknecht and N. Chaker, eds, *Fuzzy Control Theory and Practice*, New York: Springer, 2000.

[35] J. Yen and R. Langari, *Fuzzy Logic: Intelligence, Control, and Information*,

Englewood Cliffs, New Jersey: Prentice Hall, 1999.

[36] H. Ying, *Fuzzy Control and Modeling Analytical Foundations and Applications*, New York: IEEE Press, 2000.

[37] W. Zhong-dong, X. Wei-xin and Y. Jian, "Fuzzy c-mean clustering algorithm based on kernel method," in *Proceedings of the Fifth International Conference on Computational Intelligence and Multimedia Applications, ICCIMA*, Xian, China, 2003, pp. 49 – 54.

[38] J. Bezdek, R. Ehrlich and W. Full, "FCM: the fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, pp.191-198, 1984.

[39] J. Bezdek, R. Hathaway, M. Sabin and W. Tucker, "Convergence theory for fuzzy c-means: counterexamples and repairs," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 17, pp. 873-877, 1987.

[40] J. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*, New York: Plenum Press, 1981.

[41] F. Hoppner, F. Klawonn, R. Kruse and T. Runkler, *Fuzzy Cluster Analysis Methods for Classification and Image Recognition*, New York: John Wiley, 1999.

[42] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, San Diego, California: Academic Press, 1999.

[43] R. Duda, P. Hart and D. Stork, *Pattern Classification***,** New York: Wiley, 2001.

[44] C. Bishop, *Neural Networks for Pattern Recognition***,** New York: Oxford University Press, 1995.

[45] M. Jaradat and R. Langari, "An efficient real-time multisensor fusion engine based on fuzzy clustering approach," in *Fifth International Conference on Intelligent Technologies, InTech'04*, Houston, Texas, USA, 2004, pp. 1/3- 4/3.

[46] M. Jaradat and R. Langari, "An efficient real-time fusion engine with multiple directions of faulty sensor based on conventional fuzzy clustering approach," in *Fifth International Conference on Intelligent Technologies, InTech'04*, Houston, Texas, USA, 2004, pp. 1/4- 6/4.

[47] M. Gadallah, E. Soleit, and A. Mahran, "Noise immune speech recognition system," in *Proceedings of the Sixteenth National Radio Science Conference* , *NRSC '99*, Cairo, Egypt, 1999, pp. C21/1 - C21/8.

[48] Z. Lianying and L. Fengyu, "Research on computer network security based on pattern recognition," in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 2 , Washington, D.C., USA, 2003, pp. 278 – 1283.

[49] A. Tarakanov and V. Skormin, "Pattern recognition by immunocomputing," in *Proceedings of the 2002 Congress on Evolutionary Computation CEC '02*, Honolulu, Hawaii, USA, 2002, pp. 938 – 943.

[50] B. Wang, S. Wang, and J. Zhuang, "A distributed immune algorithm for learning experience in complex industrial process control," in *International Conference on Machine Learning and Cybernetics*, vol. 4, Xian, China, 2003, pp. 2138 – 2141.

[51] M. Sasaki, M. Kawafuku, and K. Takahashi, "An immune feedback mechanism based adaptive learning of neural network controller," in *Proceedings 6th International Conference on Neural Information, ICONIP '99,*vol. 2, Perth, Australia, 1999, pp. 502 – 507.

[52] K. Kumar and J. Neidhoefer, "Immunized adaptive critics for level 2 intelligent control," in *IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation*, Orlando, Florida, USA, 1997, pp. 856 – 861.

[53] K. Takahashi and T. Yamada, "A self-tuning immune feedback controller for controlling mechanical systems," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics '97*, Tokyo, Japan, 1997, pp. 101.

[54] K. Dong Hwa, "Tuning of a PID controller using an artificial immune network model and local fuzzy set," in *Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, vol. 5, Vancouver, British Columbia, Canada, 2001, pp. 2698 – 2703.

[55] Q. Zhen-Qiang, H. Guang-Da, Y. Zhao-Hua, and Z. Fu-En, "A novel control algorithm based on immune feedback principle," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, vol. 2, Beijing, China, 2002, pp. 1089 – 1092.

[56] M. Araujo, J. Aguilar, and H. Aponte, "Fault detection system in gas lift well based on artificial immune system," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, Portland, Oregon, USA, 2003, pp. 1673 – 1677.

[57] S. Liu, J. Zhang, W. Shi and W. Huang, "Negative selection algorithm based approach for fault diagnosis of rotary machinery," in *Proceedings of the 2002 American Control Conference*, vol. 5, Anchorage, Alaska, USA, 2002, pp. 3955 – 3960.

[58] D. Dasgupta, Z. Ji, and F. Gonzalez, "Artificial immune system (AIS) research in the last five years," in *The 2003 Congress on Evolutionary Computation CEC '03*, Canberra, Australia, 2003, pp. 123 – 130.

[59] L. de Castro, and J. Timmis, "Artificial immune systems as a novel soft computing paradigm," *Soft Computing Journal*, vol. 7, pp. 526-544, 2003.

[60] D. Dasgupta and N. Attoh-Okine, "Immunity based systems: a survey," in *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 1, Orlando, Florida, USA, 1997, pp. 369 – 374.

[61] S. Forrest, A. Perelson, L. Allen, and R. Cherukuri, "Self-nonself discrimination in a computer," in *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*, Oakland, California, USA, 1994, pp. 202 – 212.

[62] N. Fernando and B. Oscar, "A change detection software agent based on immune mixed selection," in *IEEE Proceedings of the 2002 Congress on Evolutionary Computation, CEC '02*, vol. 1, Honolulu, Hawaii, USA, 2002, pp. 693 – 698.

[63] L. de Castro, and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*, London, UK: Springer, 2002.

[64] R. Canham, H. Jackson, A. Tyrrell, "Robot error detection using an artificial immune system," in *Proceedings of the 2003 NASA/DoD Conference on Evolvable Hardware*, Chicago, Illinois, USA, 2003, pp. 199 – 207.

[65] D. Bradley and A. Tyrrell, "The architecture for a hardware immune system," in *Proceedings of the Third NASA/DoD Workshop on Evolvable Hardware*, Long Beach, California, USA, 2001, pp. 193 – 200.

[66] M. Jaradat and R. Langari, "A hybrid real-time system for fault detection and sensor fusion based on conventional fuzzy clustering approach," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, Reno, Nevada, USA, 2005, pp. 189-194.

[67] A. Steinhage, "Nonlinear attractor dynamics: a new approach to sensor fusion," *Proceedings of the SPIE*, vol. 3839, Boston, MA, USA, 1999. Sensor Fusion and Decentralized Control in Robotic Systems II, pp. 31-42.

[68] G. Schoner, M. Dose and C. Engels, "Dynamics of behavior: theory and applications for autonomous robot architectures," *Robotics and Autonomous Systems Journal*, vol. 16, pp.213-245, 1995.

[69] J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*, New York**:** Marcel Dekker, 1998.

[70] K. Moshe, Z. Xiaoxun and K. Paul, "Sensor fusion for mobile robot navigation," *Proceedings of the IEEE*, vol. 85, pp. 108 – 119, 1997.

[71] C. Tarin, H. Brugger, R. Moscardo, B. Tibken and E. Hofer, "Low level sensor fusion for autonomous mobile robot navigation," in *Proceedings of the 16th IEEE*

Instrumentation and Measurement Technology Conference, vol. 3, Venice, Italy, 1999, pp.1377 – 1382.

[72] C. Lei and W. Yongji, "Localization of the autonomous mobile robot based on sensor fusion," in *IEEE International Symposium on Intelligent Control*, Houston, Texas, USA, 2003, pp. 822 – 826.

[73] F. Lizarralde, E. Nunes, H. Liu and J. Wen, "Mobile robot navigation using sensor fusion," in *Proceedings IEEE International Conference on Robotics and Automation ICRA '03*, vol. 1, Taipei, Taiwan, 2003, pp. 458 – 463.

[74] Y. Zou, Y. Ho, S. Chua Chin and X. Zhou, "Multi- ultrasonic sensor fusion for mobile robots," in *Proceedings of the IEEE Intelligent Vehicles Symposium IV 2000*, Dearborn, Michigan, USA, 2000, pp. 387 – 391.

[75] D. Gregory and J. Michael, *Computational Principles of Mobile Robotics*, Cambridge, U.K.: Cambridge University Press, 2000.

[76] J. Jones, B. Seiger and A. Flynn, *Mobile Robot Inspiration to Implementation*, Natick, Massachusetts: A.K. Peters, 1999.

[77] J. Borenstein and Y. Koren, "Obstacle avoidance with ultrasonic sensors," *IEEE Journal of Robotics and Automation*, vol. 4, no. 2, pp. 213-218, 1988.

[78] H. William, H. Andrew and S. Jonathan, "SH-60 helicopter integrated diagnostic system (HIDS) program-diagnostic and prognostic development experience," in *IEEE Aerospace Conference Proceedings*, vol. 2, Manhattan Beach, California, 1999, pp. 473 - 491.

[79] J. Michael, "Diagnostics for light helicopters," in *IEEE Aerospace Conference Proceedings*, vol. 6, Manhattan Beach, California, 2000, pp. 229-235.

[80] P. Monsen, P. Kazlas, and J. Michael, "Neural network-based helicopter gearbox health monitoring system," in *Proceeedings of the IEEE Workshop on Neural Networks for Signal* , Linthicum Heights, Maryland, USA, 1993, pp. 431-440.

[81] T. Monsen, E. Manolakos, and M. Dzwonczyk, "Helicopter gearbox fault detection and diagnosis using analog neural networks," in *Proceedings of the 27th Assilomar Conference on Signals, Systems and Computers*, vol. 1, Pacific Grove, California, USA, 1993, pp. 381-385.

[82] M. Dzwonczyk and E. Huff, "Helicopter transmission health monitoring using real-time neural computing methods," in *Proceedings of the AIAA/IEEE Digital Avionics Systems Conference*, Phoenix, Arizona, USA, 1994, pp. 359-364.

[83] V. Jammu, K. Danai, and D. Lewicki, "Diagnosis of helicopter gearboxes using structure-based network," in *Proceedings of the American Control Conference*, vol. 3, Seattle, Washington, USA, 1995, pp. 1623-1627.

[84] W. Wang and P. McFadden , "Application of wavelets to gearbox vibration signals for fault detection," *Journal of Sound and Vibration*, vol. 192, pp. 927-939, 1995.

[85] S. Engel, B. Gilmartin, K. Bongort , and A. Hess, "Prognostics, the real issues involved with predicting life remaining," in *Proceedings of the IEEE Aerospace Conference*, vol. 6, Big Sky, Montana, USA, 2000, pp. 457-469.

[86] G. Kacprzynski, A. Sarlashkar, M. Roemer, B. Lamirand , A. Hess, and B. Hardman, "Calibration of failure mechanism-based prognosis with vibratory state

awareness applied to the H-60 gearbox," in *Proceedings of the IEEE Aerospace Conference*, Big Sky, Montana, USA, 2003, pp. 3271-3278.

[87] W. Hardman, "Mechanical and propulsion systems prognostics: U.S. navy strategy and demonstration (Overview)," *JOM Journal*, vol. 55, pp. 21-27, 2004.

[88] G. Dalpiaz , A. Rivola, and R. Rubini, "Effectiveness and sensitivity of vibration processing techniques for local fault detection in gears," *Mechanical Systems and Signal Processing Journal*, vol. 14, pp. 387-412, 2000.

[89] E. Callaway, *Wireless Sensor Networks Architectures and Protocols*, Boca Raton, Florida: Auerbach Publications, 2004.

[90] H. Qi, S. Iyengar, and K. Chakrabarty, "Distributed sensor networks – a review of recent research," *Journal of the Franklin Institute*, pp. 655-668, 2001.

[91] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," *2002 ACM International Workshop on Wireless Sensor Networks and Applications*, Atlanta, Georgia, USA, available: http://www.tinyos.net/media.html, 2002.

[92] R. Gupta and S. Das, "Tracking moving targets in a smart sensor network," in *Proceedings of the IEEE 58th Vehicular Technology Conference VTC 2003-Fall*, Orlando, Florida, USA, 2003, pp. 3035 –3039.

[93] M. Dubberley, A. Agogino, and A. Horvath, "Life-cycle assessment of an intelligent lighting system using a distributed wireless mote network," in *IEEE International Symposium on Electronics and the Environment*, Scottsdale, Arizona, USA, 2004, pp. 122 – 127.

[94] Crossbow Technology, Inc., “Wireless sensor networks,” TelosB research platform and TPR2400 Datasheet, available: http:/www.xbow.com, 2005.

[95] TinyOS community forum, Tutorial and Documentations, available: http://www.tinyos.net, 2003.

VITA

Mohammed Abdel Kareem Rasheed Jaradat was born in Irbid City, Jordan 1976. After graduation from high school, he attended the Jordan University of Science and Technology, 1994. He received the Bachelor of Science degree in electrical engineering majoring in communication and electronics in 1999. He worked as a service engineer for JTMS, Jordan, 1999. He worked as a communication engineer for Telecommunication Regulatory Commission (TRC), Jordan, 2000-2001.

In January 2001, he came to Texas A&M University as a graduate student in the mechanical engineering department and received his Master of Science degree in 2002. After that he began his Ph. D. program, receiving his degree in 2005. During his graduate studies, he received several awards such as: First place award, Student Research Week, Texas A&M University, 2005, and second place award, Student Research Week, Texas A&M University, 2004.

His areas of interest are mechatronics, robotics, intelligent systems and control, where he has several publications.

Mohmmad Jaradat can be reached at:

Mechanical Engineering Department

Jordan University of Science and Technology,

Irbid, Joradn

mjaradat@neo.tamu.edu