Automatic Fault Identification of a Mechanical System using Genetic Algorithm

Abhishek Bal
Department of Computer Science
RCCIIT College
Kolkata, India

Nilima Paul
Department of Computer Science
RCCIIT College
Kolkata, India

Sonali Sen
Department of Computer Science
St. Xavier's College
Kolkata, India

ABSTRACT

This paper describes a fault identification technique for mechanical system which is based on genetic algorithm using training set. The real-world application of Genetic Algorithm (GA) to the key of engineering problem becomes a rapidly emerging approach in the field of control engineering and signal processing. Genetic algorithms are convenient for searching a space in multi-directional way from large spaces and poorly defined space. In this paper Genetic Algorithm is used to identify and evaluate the fault cases. Several methods are employed in the state of art in fault identification. Here one class of efficient method are investigated which is based on optimization technique. Here it is shown that Genetic Algorithm can be used to select smaller subset of features from the large set which together form a new set that can be successful for fault identification and classification tasks. The performance of this present proposed method has been verified through two types of fitness function, namely, square function and polynomial function. Finally, fault detection exercises are performed based on the training set to verify the feasibility of this proposed method. Experimental results show that the fault is distinguished with a high precision through this present work.

Keywords

Genetic Algorithm, Fault identification, Optimization, Fitness Function, Training set.

1. INTRODUCTION

This paper introduces a real time fault or damage identification [16, 18, 19] technique for mechanical system. A mechanical system manages the power to accomplish a task which involves forces and movement. It must have a controller that compares the output to a performance goal and directs the actuator input. In the real time system it is needed to quantitative expression of time to describe the behavior of system. Real time system interacts with the external world in a way that involves times where system must respond to it a certain way and before a certain deadline. If it produces the correct result after the deadline then the system is regarded as having failed. Real time system can control safety-critical devices in vehicles, hospitals, factories, power plants and so on, so fault tolerance is frequently an issue. This type of system is able to handle the worst-case scenario.

In this present work the target is to create a real time system that has the property that it can stop the mechanical system when a serious failure occurs. For example when a machine produces an unexpected sound, then this proposed system can identify the faulty location and stop the machine immediately. That is the system can halt operation without damage.

The performance of fault identification using evolutionary computational algorithms (Genetic algorithm, GA) [2, 3, 4, 5, 6, 8, 10, 11] is improved. Here the genetic algorithms [9] are used because genetic algorithm has been traditionally considered as robust techniques [6], easily applicable to almost any domain. Genetic Algorithms (GAs are adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics. Genetic algorithm used to solve optimization problems and also helpful in random search. Optimization [10, 11] is a process of finding a best or optimal [10] solution for a problem. An optimal problem is defined as finding the values of the variables that maximize or minimize the objective problem at the same time satisfying the constraints. Two fitness function- Square and polynomial function are used for optimization purpose. GA is better than conventional AI (Artificial Intelligence). It is more robust [6, 11]. Unlike older AI systems, the GA's do not break easily even if the inputs changed slightly or in the presence of reasonable noise.

Here fault identification [19, 20] technique on the mechanical system is implemented. Let say, a machine produce repeated sound like fan or industrial machine. These machines produce a rhythmic sound. If anytime it produces an abnormal or unexpected sound rather than its normal generated sound, then there must have a problem which may cause different type of damage. Now the target is to detect those faulty locations. If such a problem occurs, then automatically machine is halted or other precautions are taken. It saves the human life in industry and avoids the loss. This damage detection mechanism is fully dependent on generated mechanical sound. This present work mainly follows two steps- Creation of Training Set and Identification of faulty location.

For creating training set, a reference fault free continuous rhythmic mechanical sound is taken which is produced by a machine like fan. Then this rhythmic sound is partitioned in a number of parts with respect of one unit time. Then each consecutive pair of sound parts are taken and then apply GA on them and store the cross point and mutation point information in the training set for which GA result same with next part of the sound.

In real life genetic algorithm gives the optimal solution and it takes more time. So for this reason the training set is used to store cross point and mutation point. Now it takes a little time with compare to applying GA directly on sound files. Now just cross point and mutation point is extracted from training set to produce GA-sound which will be compared with the original sound. Using this method also the damaged sound is recovered. This means if there is a repeated sound, and for any reason some portion is missing, then it can recover those file. The performance of damage detection technique is improved using evolutionary computational algorithms (Genetic algorithm) [2, 3, 4, 5, 6, 8, 10, 11].

Let consider any instant of time a fan is working. If any time it produces an unexpected sound then the target is to stop the faulty machine like fan which does not respond to subsequent input or does not produce further output. To do this the first two consecutive part of the mechanical sound is selected and apply the GA depending on the cross point [1, 7, 11] and mutation point [1, 7, 11] which is extracted from the training set. Then compare the GA resultant sound with the next part of the mechanical sound. If GA resultant sound is not matched with the next part of the sound file then it is said that fault exist on this location, so machine has stopped or other precautions has to take immediate. Otherwise it shift one position right and take next two consecutive sound files. The above technique will continue until the fault is detected or end of the mechanical sound. After a lot of test, it is observed that the fault is distinguished with a high precision through this present technique.

The fault detection are recognized using genetic algorithm follows a systematic technique which includes repeated iterations consisting of genetic operators that are selection, crossover, mutation and reproduction until the optimal solution[9,11,12] is found.

2. LITERATURE REVIEW

In 2006, Abhinav Saxena and Ashraf Saad [21] proposed use of the Genetic Algorithm for identifying near optimal design parameters of Diagnostic systems for condition monitoring of mechanical systems. They also show that GA can be used to select a smaller subset features from the large set space.

In 2001, Chou, J.H. and Ghaboussi, J. [16] and in 1996, Mares, C. and Surace, C. [18] have represented a detection of structural damage is an inverse problem in structural engineering. In this study, optimization problem is solved by using by using Genetic Algorithm. This proposed method is able to detect the location of damage to a reasonable level of accuracy.

In 2001, S.S. Law, T.H.T. Chan, D. Wu [15] proposed a structural medaling method which is based on the concept of Damage-Detection-Oriented-Modeling, in which each super-element representing a segment of a large-scale structure.

In 2003, F.T.K Au,Y.S Cheng,L.G. Tham,Z.Z. Bai [14] have represented a procedure for detecting the structural damage based on micro-genetic algorithm using noisy model test data. This paper represent micro-genetic algorithm to quantify the damage extent by minimizing the errors between the measured data and numerical results.

In 2003, Raich, A.M. and Liszkai, T.R. [17] proposed a robust structure damage detection methodology which can handle noisy frequency response function. This proposed method describe that the IRR GA is less sensitive to noise than a SGA.

In 2004, Mehul A. Shah,Joseph M. Hellerstein,Eric Brewer [19] proposed a method that masks failures in a cluster to provide availability and fault-tolerance for long-running, parallelized dataflow. This method allows tolerating failures without sacrificing result quality and automatically recovers the lost pieces.

3. PROPOSED WORK

This section describes the method of capturing the acoustics of mechanical sound and details of the experimentation procedure.

3.1 Recording of Sound

Here microphone is used as a sound recorder to record mechanical reference sound in a closed room that is fault free and save the sound using audio file format (.wav). Then the amplitude values of the mechanical sound files are stored in excel sheet. As the sound are recorded using stereo type recorder, so left and right channel amplitude value is produced. So any one channel is chosen as amplitude values to process. For instant mechanical sound also microphone is used as a sound recorder and apply previous process for extraction the features of the instant mechanical sound.

3.2 Operators of genetic Algorithm

Genetic operators are used to maintain the genetic diversity. In GA, three operators are used that are selection, crossover and mutation. In the selection process two parent chromosomes are selected from the population according to their fitness that is better fitness, bigger chance to be selected. In the present approach the selection operator is used to select each two consecutive parts of the mechanical sound at each time instead of concentrating on fitness value. Crossover is a genetic operator that combines two chromosomes or parents to produce a new chromosome or offspring. The actual idea behind the crossover is that new offspring may better than both of the parents if it take the best characteristics from the each of the parents. Crossover involves choosing a random position and swapping the bits that occur after this position. In this proposed technique one point [1, 11] crossover is used. Mutation is the last genetic operator which is needed to maintain the genetic diversity from one generation to the next generation. Mutation alters one or more gene values in a chromosome from the initial state. It helps to prevent the population from stagnating at any local optimum. Here flip bit [1, 11] mutation technique is used.

3.3 Fault Identification of a Mechanical system Using Genetic Algorithm

A mechanical system can manage the power to accomplish a task which involves the forces and movement. It consists of a system of mechanisms that shape the actuator input to achieve a specific output. This type of system must have a controller that compares the specific output to a performance goal and then directs the actuator input. So, it is require to monitoring the system and to identifying the most portable fault leading to failure.

Let consider a machine produces a continuous rhythmic mechanical fault free sounds. Now this rhythmic sound is partitioned in a following set of parts shown in Figure1 with respect of one unit time. These entire sound files have some common features because all of these parts are created by a single source of machine.

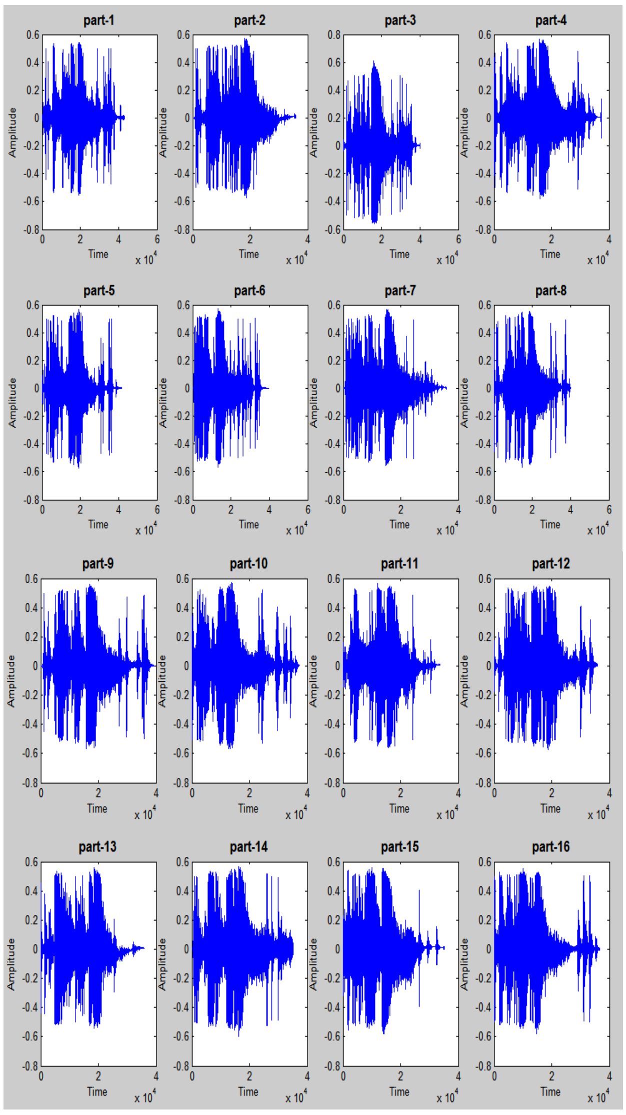

Fig 1. Set of mechanical sound sample

Due to different types of reason it may produces an abnormal or unexpected sound rather than its normal generated sound, which may create the faults or damages. Then the target is to detect those faulty locations. If such a problem occurs, then the system stops the machine immediately. That is the system can halt operation like without damage.

3.3.1 Training Set creation:

Training set is a portion of data used to train a model for prediction or classification of value that are known in the training set but unknown in other data. Training set is used in the genetic programming to discover potentially predictive relationship.

At first training set is created depending on above set of mechanical damage free sound files shown in Figure1 so that at any instant time if this machine produces an abnormal sound rather than its normal generated sound, then the present technique can detect that there must have a problem which may cause different type of the damage.

The Genetic algorithm finds the optimal solution, but if GA is applied directly on sound files when fault detection technique is being processed it takes more time to find out the cross point and mutation point according to fitness function. So for reducing the time before processing the training set is created first time. Now when fault detection process is running it just extract cross point and mutation point from the training set and then apply GA on sound files.

Table 1. Training set for cross and mutation point

1st file	2nd file	Cross point	Mutation Point	GA-Result	Compare With
Part1	Part2	7	5	GA3	Part3
Part2	Part3	4	4	GA4	Part4
Part3	Part4	4	5	GA5	Part5
Part4	Part5	6	4	GA6	Part6
Part5	Part6	7	5	GA7	Part7
Part6	Part7	4	4	GA8	Part8
Part7	Part8	5	5	GA9	Part9
Part8	Part9	8	4	GA10	Part10
Part9	Part10	8	5	GA11	Part11
Part10	Part11	5	9	GA12	Part12
Part11	Part12	4	4	GA13	Part13
Part12	Part13	6	7	GA14	Part14
Part13	Part14	9	4	GA15	Part15
Part14	Part15	5	4	GA16	Part16
Part15	Part16	8	7	GA1	Part1

First two consecutive sound parts are taken from set of sound file which are shown in Figure. 1 let say part_i and part_(i+1). Now the target is to produce sound file which is same with next one that is part_(i+2).So apply GA on part_i and part_(i+1). By applying genetic algorithm, find out cross point and mutation point for which GA_(i+2) is same with the part_(i+2). At first cross point and mutation point is calculated by function $\log_2(n+1)$ where n is the number of bits in the individuals. This cross point and mutation point moves until it get the GA result which is same with the next sound file (cross point and mutation point movement possible until there is no combination left for cross point and mutation point).

From this Table 1 it is shown that cross point is 7 and mutation point is 5 for part_1 and part_2 and genetic algorithm produce the result that is GA_3 which is same with part_3.So that GA_3 is the alternative of part_3.

Then it shift one position right, take part_2 and part_3 and apply genetic algorithm on them. Here for cross point is 4 and mutation point 4 genetic algorithm produce the result that is GA_4 which is same with part_4.

Again it shift one position right and apply same technique. In this way, for every two consecutive sounds the previous process is applied. At last two end consecutive sounds are talen i.e. part_(k-1) and part_k and apply genetic algorithm and find out suitable cross point and mutation point for which genetic algorithm produce a result which is same with part_1. For each two consecutive sound files cross point and mutation point are stored in the training set.

3.3.2 Two Types of Cross point and Mutation Point:

In this implementation two methods are used for find out cross point and mutation point that are (n/2) [1] and $\log_2(n+1)$ [1] where n is number of bits in the amplitude value of the sound file. Then result of these two methods is compared to show which one gives better result.

Here, for creating the training set each of these two methods are applied. For each method GA is also applied on each two consecutive sound files then compares the GA result with the next sound file for find out the exact cross point and mutation point.

From this experimental it is observed that if $\log_2(n+1)$ function is used for find out cross point and mutation point then percentage of matching increased than the other method and also increased the performance of the system.

3.3.3 Two Types of Fitness Function:

The fitness function is a particular type of objective function that is used to summarize, to determine how close a given design solution is to achieving the set aims. The fitness function plays a very useful role in guiding GA to find out the best solutions within a large search space. Good fitness functions will help GA to explore the search space more effectively and efficiently. In the present work two type of fitness function are used then compare the result of these two methods. The comparison part is discussed in the experimental portion.

a) Maximize the function ($f(x) = x^2$)

This function [11] could be solved by a variety of traditional methods such as a hill-climbing. This function start from any real number x in the domain of f and evaluate at the point x. Then pick a new x which is at a small distance in the direction from current x.

b) Polynomial function ($f(x) = a_0 + a_1x + a_2x^2 + a_3x^3$)

Suppose for some function p(x) let find the third degree polynomial that best fit this function at x = 0.

Let's call this polynomial

$$f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 \quad (1)$$

To determine f, it is need to find values for the four coefficients a0, a1, a2, a3.

At first this present work have to define what is mean by the "best" fit to p at x=0. There are four unknowns coefficient. One condition is that the graph of f should pass through the point (0, p(0)). But this is equivalent to f(0) = p(0). Since f(0) = a0 ,so a0 = p(0).

It can be said that f(x) is the best fit to p(x) at the point x = 0 if and only if $f(0) = p(0)$, $f'(0) = p'(0)$, $f''(0) = p''(0)$, $f^{(3)}(0) = p^{(3)}(0)$.From here it is observed that the derivatives of f can easily express in terms of a_1, a_2, a_3 .So the derivatives are

$$f'(x) = a_1 + 2a_2 x + 3a_3 x^2 \quad (2)$$
$$f''(x) = 2a_2 + 3.2\, a_3 x \quad (3)$$
$$f^{(3)}(x) = 3.2\, a_3 \quad (4)$$

Thus $f(0)=a_0$, $f'(0)= a_2$, $f''(0) = 2a_2$, $f^{(3)}(0) = 3 \cdot 2\, a_3$.

Finally, P (0) = f (0), it can be solved for the coefficients of P (x):

$$a_k = \sum_{k=0}^{3} \left(\frac{p^{(k)}(0)}{k!}\right) \quad (5)$$

Now it can be written an explicit formula for the third degree polynomial which best fits p(x) at x = 0:

$$f(x) = p(0) + p'(0)x + \frac{p''(0)}{2!}x^2 + \frac{p'''(0)}{3!}x^3 \quad (6)$$

The above equation can be expressed more compactly using the Σ–notation that is :

$$f(x) = \sum_{k=0}^{3} \left(\frac{p^{(k)}(0)}{k!}x^k\right) \quad (7)$$

Consider p(x) =sin(x).So if a_0, a1, a2, a3 are calculated depending on p(x) =sin(x) then $a_0=0$, $a_1=1$, $a_2=0$, $a_3=-0.166$.

From this present work it is observed that if polynomial fitness function that is $f(x) = a_0 + a_1x + a_2x^2 + a_3x^3$ is used then performance of this approach more accurate rather than using the fitness function $f(x) = x^2$.

4. ALGORITHM

4.1 Algorithm for Training Set Creation

```
Algorithm TrainingSet_creation
Steps:
    1) n←total no. of sound files
    2) in1←sound1 //first sound file
    3) in2←sound2 //second sound file
    4) pos← POSITION(sound2)
    5) while(pos<=n) do
    6) if(pos=n) then
    7) in3←sound1
    8) else
    9) pos← pos+1
   10) n3←soundpos
   11) end if
   12) crosspoint←log2(x+1)
   13) mutapoint← log2(x+1)
   14) repeat
        //crossover and mutation section
   15) offspring1←CROSSOVER(in1,in2,crosspoint)
   16) offspring2←MUTATION(offspring1,mutapoint)
   17) if(offspring2=in3) then
   18) break
   19) else
   20) crosspoint←MOVE1(crosspoint)
   21) mutapoint←MOVE2(mutapoint)
   22) end if
   23) until(cross and mutation point movement possible)
   24) STORE_TRAIN(crosspoint,mutapoint)
   25) in1←in2
   26) in2←in3
   27) done
   28) Stop.
```

In the above algorithm, **POSITION()** method is used to find out the position the sound file from the set of sound files. **CROSSOVER()** method is basically used for crossover between two sound files and **MUTATION()** is used for mutation operation. **MOVE1()** method is used for movement of the cross point and **MOVE2()** method is used for movement of the mutation point.

STORE_TRAIN() method is used for storing the cross point and mutation point into the training set for each two consecutive sound files. Here **x** is the number of bits which is used to represent each amplitude value.In the above algorithm $\log_2()$ function is used for finding the cross point and mutation point and for each two consecutive sound parts cross point and mutation point are stored in the training set.

4.2 Algorithm for Fault Identification

Algorithm Fault_Prediction
Steps:

```
1)  c←0
2)  i←1
3)  in1←$sound_i$ // initialize a sound
4)  i←i+1
5)  in2←$sound_i$ // initialize next sound
6)  repeat
7)  cosspoint←EXTRACT_1(in1,in2)
8)  mutapoint←EXTRACT_2(in1,in2)
9)  offspring1←CROSSOVER(in1,in2,crosspoint)
10) offspring2←MUTATION(offspring1,mutapoint)
11) i←i+1
12) in3←$sound_i$
13) if(offspring2!=in3) then
14) c←1
15) break
16) end if
17) in1←in2
18) in2←in3
19) until(end of the sound creation)
20) if(c=1) then
21) Write "Damage Occurred"
22) else
23) Write "NO Damage Occurred"
24) end if
25) Stop
```

In above algorithm, **EXTRACT_1()** method is used for extracting the cross point from the training set for each two consecutive sound files and **EXTRACT_2()** method is used for extracting the mutation point from the training set for each two consecutive sound files. **CROSSOVER()** method is basically used for crossover between two sound files and **MUTATION()** is used for mutation operation.

5. EXPERIMENTAL RESULT AND DISCUSSION

In this section the results of some test cases are discussed using the present approach with the use of Genetic Algorithm. Now some example is shown to illustrate the experimental result.

5.1. Experiment 1:

In this experimental part the present approach is illustrated with an instant mechanical sound which is sub divided into 16 parts which are shown in Figure 2.

Let consider at any instant time the same machine produce a continuous rhythmic mechanical sounds shown in Figure 2. Now the task is to check the machine is working fine or not. If it produces the abnormal or unexpected sounds then automatically machine is stopped or other precautions are taken.

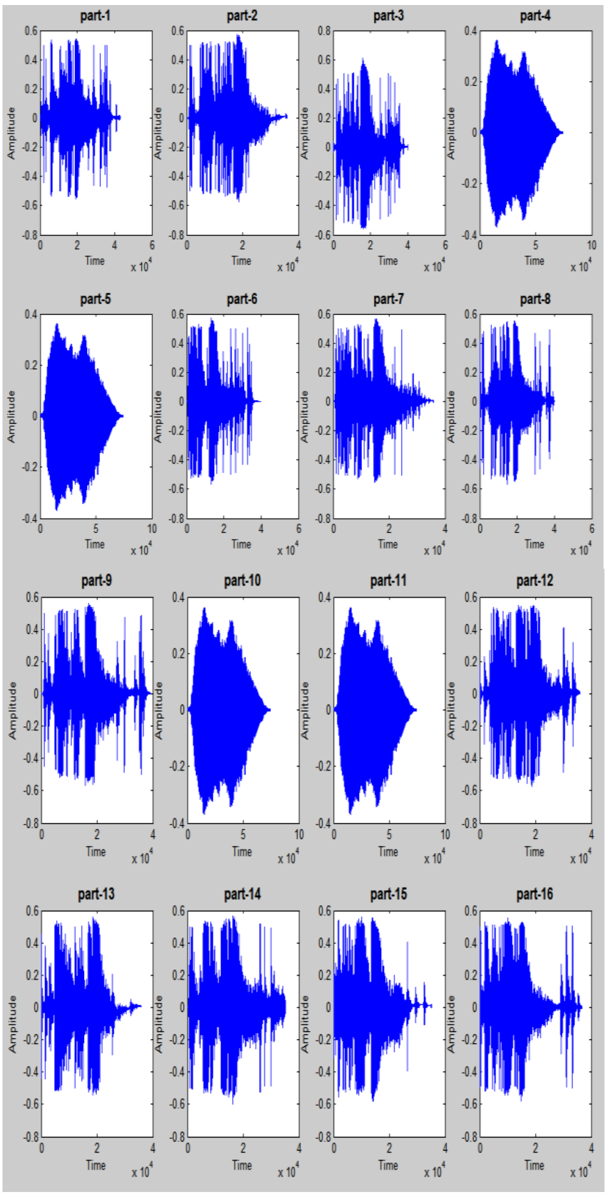

Fig 2. Set of mechanical instant sound sample

To do this first two consecutive sound files are taken and extract the cross point and mutation point from the training set for these two sound files. Then apply GA on these two sound files depending on selected cross point and mutation point. If GA result same with third part of the sound file then there is no any damage or fault occurred in third part of the sound file. Else damage occurs in the third part. If there is no damage occurred in the third part of the sound file then automatically the next two consecutive sound file is chosen and continue the same process.

Let consider the part_1 and part_2 parts from the instant mechanical sound and extract the cross point and mutation point for these two sound file from the training set. Then apply GA on them and compare the GA result with part_3 which is shown in Figure 3. Here for cross point 7 and mutation point 5 GA produce a sound file which is same with the part_3. So no fault or damage occurs in part_3.

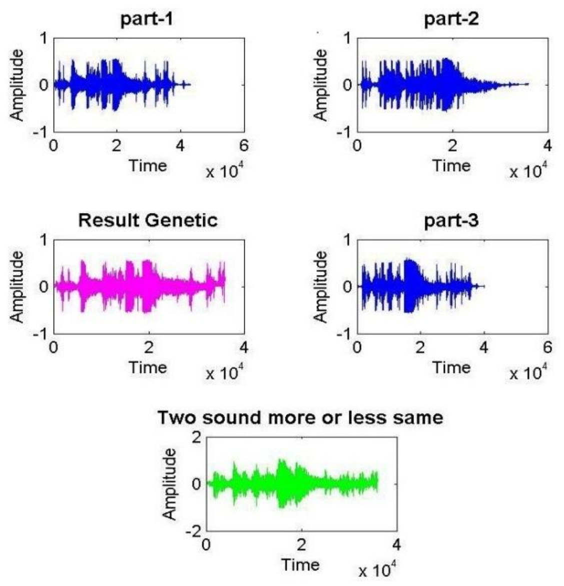

Fig 3. GA between part-1 and part-2

Then the fault prediction algorithm choose the next two consecutive sound files that are part_2 and part_3 and extract the cross point and mutation point for these two sound file from the training set. Then apply GA on them and compare the GA result with part_4 which is shown in Figure 4.

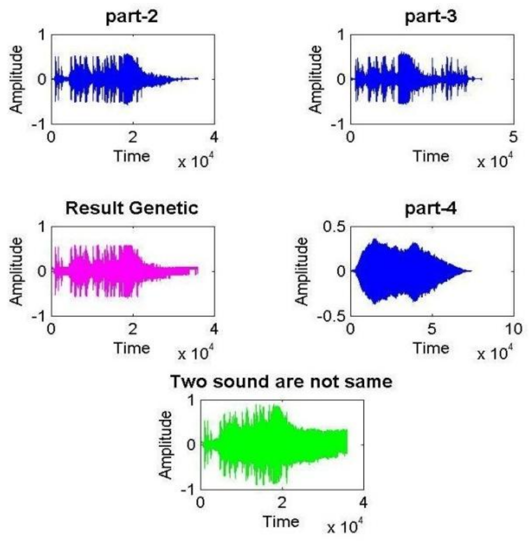

Fig 4. GA between part-2 and part-3

In this Figure 4, it is seen that for cross point 4 and mutation point 4 GA produce a sound file which is not same with the part_4. So from the proposed method it is said that there is a fault or damage occurs in the part_4 sound part and machine automatically is stopped.

5.2. Experiment 2:

5.1.1 Comparison between Two Types of Cross point and Mutation Point

In this implementation two methods are used for find out cross point and mutation point that are (n/2) and $\log_2(n+1)$ [1] where n is number of bits in largest amplitude value of the sound files. For each of these two methods the present work is implemented and also compares result of these two methods in Table 2.

In Table 2, for each of these two methods GA is applied on each two consecutive sound files then compare the GA result with the next sound file and observe the difference between two methods.

Table 2: Comparison between two types of Cross point and Mutation Point

GA result	Compare with	% of same using cross and mutation point (n/2)	% of same using cross and mutation point $\log_2(n+1)$
GA_3	Part3	55.17	71.83
GA_4	Part4	54.27	66.25
GA_5	Part5	58.06	72.79
GA_6	Part6	52.03	69.62
GA_7	Part7	56.06	63.18
GA_8	Part8	56.73	71.28
GA_9	Part9	57.14	73.32
GA_10	Part10	52.08	68.37
GA_11	Part11	57.16	75.97
GA_12	Part12	54.57	60.06
GA_13	Part13	57.94	71.77
GA_14	Part14	52.74	70.62
GA_15	Part15	52.69	68.25
GA_16	Part16	54.73	72.84
GA_1	Part1	67.25	77.25

From Table 2, it is conclude that if $\log_2(n+1)$ function is used for find out cross point and mutation point then percentage of matching increased than the other method.

5.1.2 Comparison between two fitness function

The performance of the present approach has been verified through two types of fitness function, namely, square function and polynomial function. Already the details about these function is described in the proposed work section.

Here, the of these two functions are compared and try to find out the best fitness function between two functions because fitness function plays a very useful role in guiding GA to find out the best solutions within a large search space. Good fitness function will help GA to explore the search space more effectively and efficiently.

Table 3: Experimental value for square function

GA result	Compare with	$f(x)=x^2$	
		% of same	Euclidean
GA_3	Part3	71.83	25.80
GA_4	Part4	66.25	30.15
GA_5	Part5	72.79	23.12
GA_6	Part6	69.62	28.85
GA_7	Part7	63.18	33.25
GA_8	Part8	71.28	25.97
GA_9	Part9	73.32	21.45
GA_10	Part10	68.37	29.85
GA_11	Part11	75.97	18.21
GA_12	Part12	60.06	35.12
GA_13	Part13	71.77	25.80
GA_14	Part14	70.62	23.87
GA_15	Part15	68.25	28.67
GA_16	Part16	72.84	23.84
GA_1	Part1	77.25	15.98

In Table 3, the fitness function $f(x)=x^2$ is described. Then compare the GA result with the next sound file. Here two types of comparison methods is used. One is using Threshold

Value method and another is using Euclidean Distance method.

Table 4: Experimental value for polynomial function

GA result	Compare with	$f(x)=a_0+a_1x+a_2x^2+a_3x^3$	
		% of same	Euclidean
GA_3	Part3	74.50	21.7172
GA_4	Part4	69.77	28.2418
GA_5	Part5	78.57	18.5472
GA_6	Part6	71.74	25.7159
GA_7	Part7	69.78	28.2418
GA_8	Part8	72.16	23.5478
GA_9	Part9	75.46	18.9875
GA_10	Part10	70.54	23.8742
GA_11	Part11	77.85	16.8801
GA_12	Part12	62.32	33.2863
GA_13	Part13	73.17	21.4571
GA_14	Part14	74.74	22.8453
GA_15	Part15	69.21	28.9902
GA_16	Part16	73.18	21.4571
GA_1	Part1	78.99	13.2574

In Table 4, the same approach is discussed using polynomial function. Then compare the GA result with the next sound file. Here also two types of comparison methods is used. One is using Threshold Value method and another is using Euclidean Distance method.

From above Table 3 and Table 4, it is conclude that if the polynomial fitness function $f(x) = a_0 + a_1x+a_2x^2+a_3x^3$ is used then performance of the present approach more accurate rather than using the fitness function $f(x) =x^2$.

6. CONCLUSION

In this paper, an improved GA-based fault identification algorithm is discussed using two type of fitness function. First a training set is created using genetic algorithm. Then depending on the training set, the present work can detect the faulty location and stop the machine immediately.

From the experimental results, it is observed that the proposed technique can accurately predict the fault and to identify the faulty location in the mechanical sound. Furthermore, the proposed method can also be used to overcome the problem of error in selection the measurement point for damage detection. However, this approach is little time consuming which can reduce the efficiency for the damage detection. Hence a little improvement to this present proposed approach will be focus of future studies.

7. REFERENCES

[1] Abhishek Bal, Nilima Paul, Suvasree Chakraborty, Sonali Sen, ”Voice Matching using Genetic Algorithm”, International Journal of Advanced Computer Research, Volume-4 Number-1 Issue-14,pages 305-312 March-2014.

[2] Georges R. Harik, Fernando G. Lobo, and David E. Goldberg ,”The Compact Genetic Algorithm”, University of Illinois at Urbana-Champaign Urbana,IL 61801,IlliGAL Report No. 97006,August 1997.

[3] K. F. Man, Member, IEEE, K. S. Tang, and S. Kwong, “Genetic Algorithms: Concepts and Applications”, Member, IEEE Transaction on Industrial Electronics, Vol-43, No-5, October 1996.

[4] Kumara Sastry(2), David Goldberg(2), Graham Kendall(3), “GENETIC ALGORITHMS”,Springer,2. University of Illinois, USA, 3.University of Nottingham, UK, 2005.

[5] Davis, L. D. (ed),” Genetic Algorithms and Simulated Annealing, Pitman publishing”, London,1987.

[6] Sastry, K., “Evaluation-relaxation schemes for genetic and evolutionary algorithms”, Master’s Thesis, General Engineering Department, University of Illinois at Urbana-Champaign, Urbana, IL,2001.

[7] Sastry, K. and Goldberg, D. E., 2, Let’s get ready to rumble: Crossover versus mutation head to head, in: Proc. 2004 Genetic and Evolutionary Computation Conf. II, Lecture Notes in Computer Science, Vol. 3103, Springer, Berlin, pp. 126–137,2004.

[8] Srivastava, R. and Goldberg, D. E., Verification of the theory of genetic and evolutionary continuation, in: Proc. Genetic and Evolutionary Computation Conf., pp. 551–558, 2001.

[9] David E. Goldberg, Addison-Wesley, “Genetic Algorithms in Search Optimization and Machine Learning”, chapter 1-8, page 1-432, 1989.

[10] Randy L Haupt, “Practical genetic Algorithm”, John Wiley and Sons Inc, chapter 1-7, page 1-251,2004.

[11] Artificial Intelligence: Course Content, Lecture hours–42, RC Chakraborty, June 01, 2010.Available: http://www.myreaders.info/html/artificialintelligence.html.

[12] Goldberg, D. E., "Genetic algorithms in Search, Optimization & Machine Learning”, Addison-Wesley, New York, 1999.

[13] L. D. Goh, N. Bakharya, A. A. Rahmana, B. H. Ahmada, ”Prediction of Unmeasured Mode Shape Using Artificial Neural Network for Damage Detection ”,Goh et al. / Jurnal Teknologi (Sciences & Engineering) 61:1,vol 61 ,No 1,pages 57–66,12 February 2013 .

[14] F.T.K Au,Y.S Cheng,L.G. Tham,Z.Z. Bai, ”Structural Damage Detection Based on a Micro-genetic Algorithm Using Incomplete and Noisy Model Test Data”, Journal of Sound and Vibration.259:1081-1094,2003.

[15] S.S. Law, T.H.T. Chan, D. Wu, ”Efficient numerical model for the damage detection of large scale structure”, Engineering Structures, Volume 23, Issue 5, Pages 436-451,2001.

[16] Chou, J.H. and Ghaboussi, J., “Genetic Algorithm in Structural Damage Detection”, Computers and Structures, Vol. 79, pp. 1335-1353, 2001.

[17] Raich, A.M. and Liszkai, T.R., “Benefits of Applying and Implicit Redundant Representation Genetic Algorithm for Structural Damage Detection in Noisy Environments”, Genetic and Evolutionary Computation Conference, Vol. 2724, pp. 2418-2419, 2003

[18] Mares, C. and Surace, C., "An application of genetic algorithms to identify damage in elastic structures", Journal of Sound and Vibration, Vol. 195, No. 2, pp. 195-215, 1996.

[19] Mehul A. Shah,Joseph M. Hellerstein,Eric Brewer.,"Highly Available, Fault-Tolerant, Parallel Dataflows",SIGMOD 2004 June 13-18, 2004.

[20] Klaus Echtle , Irence Eusgeld, “A Genetic Algorithm for Fault-Tolerant System Design”,springer,LNCS 2847,pp. 197-213,2003

[21] Abhinav Saxena ,Ashraf Saad,“Genetic Algorithms for Artificial Neural Net-based Condition Monitoring System Design for Rotating Mechanical Systems”,Journal of Applied Soft Computing, Volume 34,pp 135-149,2006.