Condition Monitoring and Diagnosis of an IC Engine using Vibration Recognition

Rishab R. Kannamvar, Onkar V. Repe, Dr. Pradip P. Patil, Nitin Shetty, Yash R. Maurya

Department of Mechanical Engineering,
SIES Graduate School of Technology, Navi Mumbai 400706, India

Abstract: This paper presents a model-based approach for thecondition monitoring and fault diagnosis of an IC engine usingthe MATLAB software. The vibration signatures of normally aspirated engine contain valuable information about the health of the engine. So, to extract these features a Simulink model consistingof an engine and gearbox is constructed and then the vibrationaldata has been recorded using an accelerometer connection. Healthy and fault induced data are extracted and are further processed to match the input requirements of the MATLAB Software libraries.21 significant features in both the time and spectraldomain have been extracted using the diagnostic feature designer. Which are then used to train the classifier which results in a function, which can be used to predict the nature ofnew untested data signals.

***Index Terms* – Predictive Maintenance, IC Engine, Vibration signature, Simulink model, MATLAB**

I. INTRODUCTION

In today's world we depend on a wide range of machines to do our work. But while using such high range of machines we know that over a period every machine eventually breaks down, unless it is being maintained. So, to increase the performance and reduce costs. Companies use different ways of maintenance programs. Out of which one way is reactive maintenance, where the machine is used to its limits and repairs are performed only after machine failure.

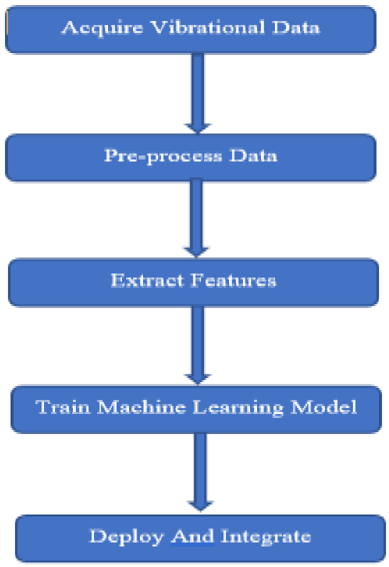
Suppose for example, if you're dealing with a light bulb it is used to its limit, then it may make sense to go with the reactive approach. But now if you are dealing with car's, bike or an airplane engine which have complex part and components and there are some components which are expensive too. You cannot really risk running it to failure, as it will be extremely costly to repair highly damaged parts. But, more importantly, it is a safety issue.

So, to solve this kind of problem predictive maintenance can be used. It helps you estimate time before the failure occurs. It also pinpoints problems in your complex machinery and helps you identify what parts need to be fixed. This way, you can minimize downtime and maximize equipment lifetime.

Hence in this paper we present a model-based approach for condition monitoring and diagnostic of an IC engine using vibration and sound recognition technique.

II. METHODOLOGY

The following flow chart shows a step-by-step process followed while conducting this research.



2.1 Acquire Vibrational Data

2.1.1 Initial Plan

The initial plan was to visit a automobile workshop and record the vibrational data of an single cylinder 4 stroke engine, by placing a Digital Vibration Meter DIGI-VIBRO Model 1332B accelerometer on multiple positions.

Figure 1 Accelerometer

Figure 2 Single Cylinder 4-stroke Engine

The basic idea was to record healthy data and fault induced data from the given engine. Faults to be detected and how we plan to induce it-

i. Cam Chain Noise: The cam chain is the element within the engine which transfers the drive from the crank shaft to the cam shaft. The tension on the cam chain can be adjusted by pushing the slider inwards or outwards. This can be done using the tension adjuster, which is accessible. Whenever cam chain is under tension, it produces the cam chain noise. All Cam Chain noise faults can be seeded by varying the tension on the Cam Chain using the tension adjuster.

ii. Piston Slap: The normal piston clearance is 0.05 mm. After the tests with normal conditions, the cylinder can be bored with two oversized clearances (3 times and 6 times normal piston clearance) to simulate piston slap faults. 3 times normal clearance can represent moderate piston slap faults and 6 times normal clearance can represent severe piston slap faults.

iii. Cylinder Head Noise: In an internal combustion engine, the cylinder head sits above the cylinders and consists of a platform containing part of the combustion chamber and the location of the valves and spark plugs. Any noise emanating from the cylinder head which is not produced by the tappet clearance is termed as the cylinder head noise. The component other than the valve and rocker arm assembly within the cylinder head is the cam shaft. Any defect on the cam shaft known as cam-lobe tends to produce the cylinder head noise.

iv. Fuel Injectors: The engine can be run with two series of healthy and faulty injectors. Injectors which were used for a period and did not spray well, will be chosen as faulty injectors.

But due to the covid reasons, it was not viable for us to go out to workshops to obtain the data and hence we decided to revise the plan and rather build a Simulink model and generate the required healthy and fault induced data.

2.1.2 Revised Plan

Simulink2020a version was used to build the below model which was used to imitate an IC engine with a connected gear box and differential. There were many built in readily available powertrain and differential models, but we created our own model from scratch to accommodate the three-axis accelerometer in Simulink.

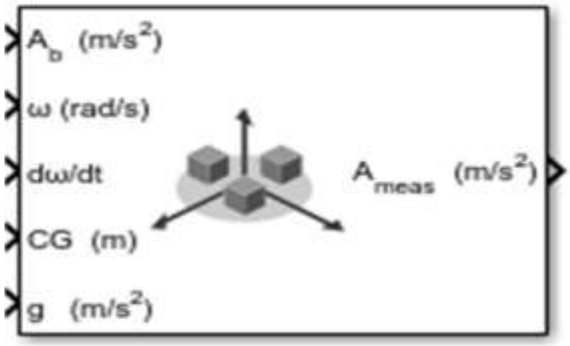


Figure 3 Three Axis Accelerometer in Simulink

This accelerometer has 5 input ports, all of which accept only 3-d inputs. The 2nd port represents the angular velocity i.e., the wheel rpm and hence a whole gearbox and differential block imitating a 6speed gear box was constructed to convert the engine torque and rpm input from the engine block into wheel rpm and torque.

The model consists of a display unit which shows the connection of the engine unit, gearbox and differential unit, and the accelerometer. It also contains the display scopes of the variables like wheel torque, wheel rpm, engine rpm, engine torque and the accelerometer output.

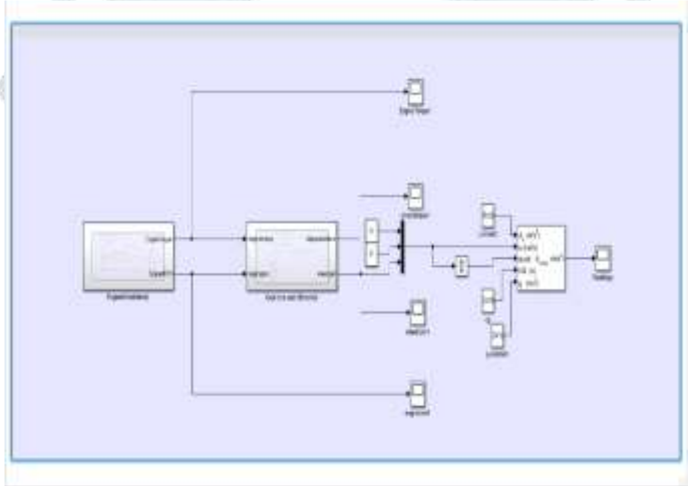


Figure 4 Display Unit

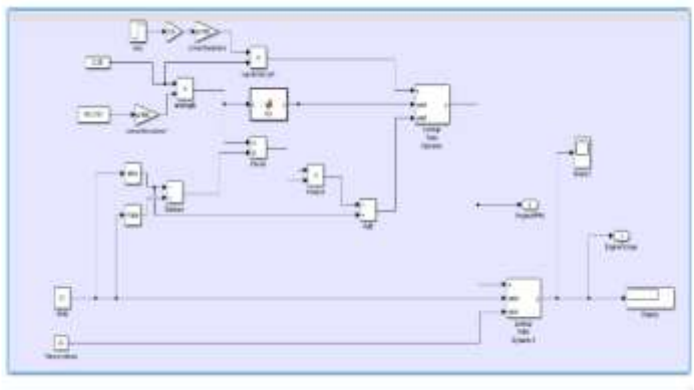


Figure 5 Engine Model

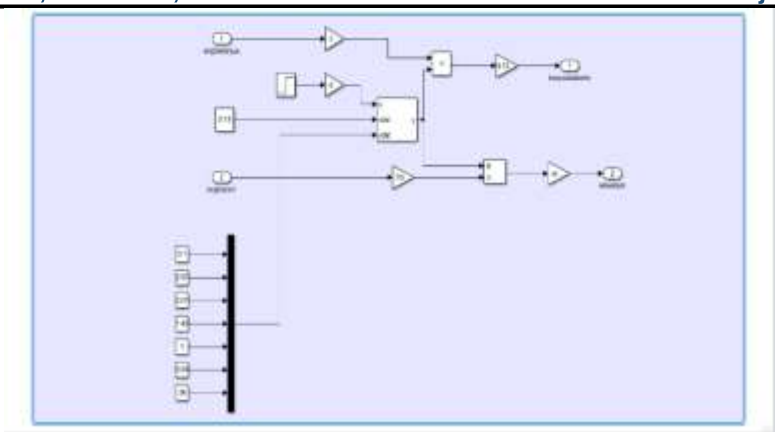
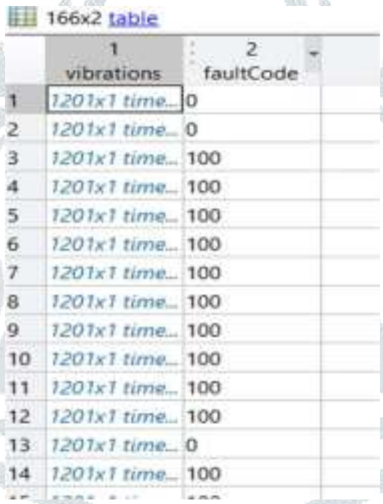


Figure 6 Gearbox and Differential

2.2 Data Pre-processing

The extracted data from the model must be in readable format for the MATLAB app, it must consist of three variables i.e.:
Data variables — The main content of the ensemble members, including measured data and derived data that you use for analysis and development of predictive maintenance algorithms. For example, in the illustrated gear-box ensembles, Vibration and Tachometer are the data variables. Data variables can also include derived values, such as the mean value of a signal, or the frequency of the peak magnitude in a signal spectrum. In our case Vibrations is the data variable
Independent variables — The variables that identify or order the members in an ensemble, such as timestamps, number of operating hours, or machine identifiers. In the ensemble of measured gear-box data, Age is an independent variable. In our case Time is the Independent variable
Condition variables — The variables that describe the fault condition or operating condition of the ensemble member. Condition variables can record the presence or absence of a fault state, or other operating conditions such as ambient temperature. Condition variables can also be derived values, such as a single scalar value that encodes multiple fault and operating conditions. In our case Fault code is the condition variable.



166x2 table

	1 vibrations	2 faultCode
1	1201x1 time...	0
2	1201x1 time...	0
3	1201x1 time...	100
4	1201x1 time...	100
5	1201x1 time...	100
6	1201x1 time...	100
7	1201x1 time...	100
8	1201x1 time...	100
9	1201x1 time...	100
10	1201x1 time...	100
11	1201x1 time...	100
12	1201x1 time...	100
13	1201x1 time...	0
14	1201x1 time...	100

Figure 7 Pre-processed Data

Fault codes	Health condition
0	Healthy
1	Fault in part1
10	Fault in part1
100	Fault in part1
1000	Fault in part1

Table 1 Fault Code Representation

The fault codes are stored in binary codes and each fault code represents a particular simulated health condition of the engine.

2.3 Extract Features

For the purpose of feature extraction, we used the feature designer tool available in MATLAB2020a.

2.3.1 Import and Visualize

The processed data is stored in the variable "Engdata" in the table format. This data is imported into the tool and it is visualized in the time and frequency domains.

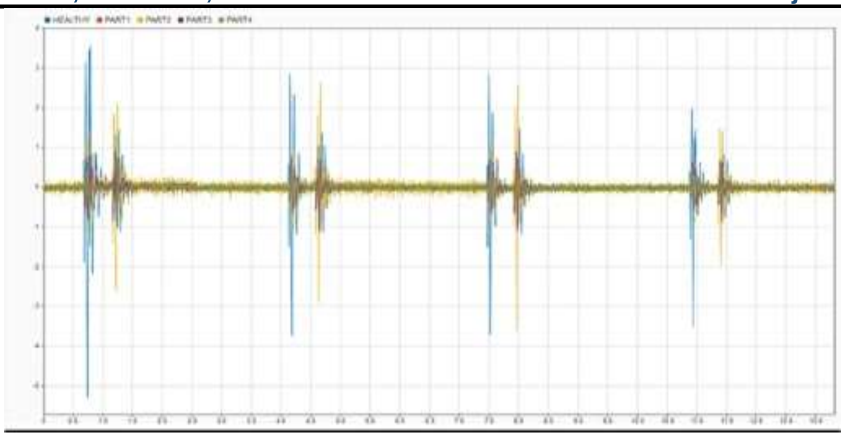


Figure 8 Data Visualization using MATLAB software

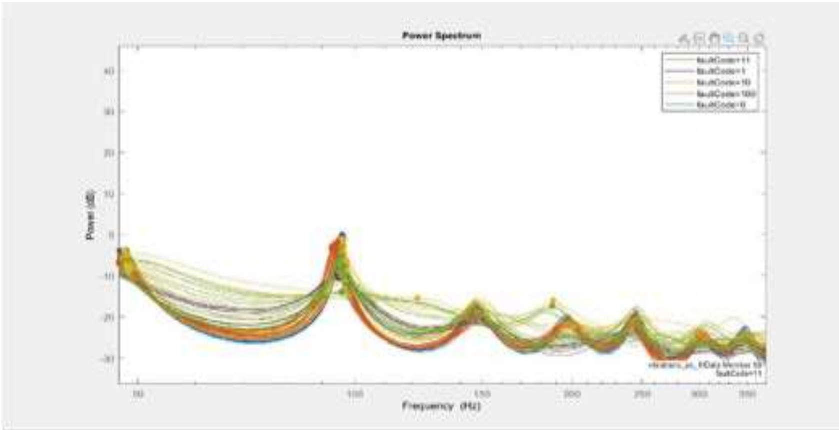


Figure 9 Power Spectrum

2.3.2 Extract Time and Spectral Features

Primary goal of a reliable monitoring system is to obtain stable, significant and non-redundant data, also called ‘‘features” from the system signature. The feature could be a parameter, i.e. real number or a two-three-dimensional graph. Two dimensional could be spectrum, time- synchronous average, mean-instantaneous power, discrete wavelet transform detail, Empirical Mode Decomposition results, cestrum; three-dimensional graph could be continuous wavelet transform, spectral correlation density, Wigner Ville distribution. This data can then be used as input by a decisional algorithm which performs both monitoring and diagnostic activities. In our case we first extract 13 features in time domain such as the mean, rms, crest factor etc. after the extraction. The histograms are plotted for each of these time domain features.

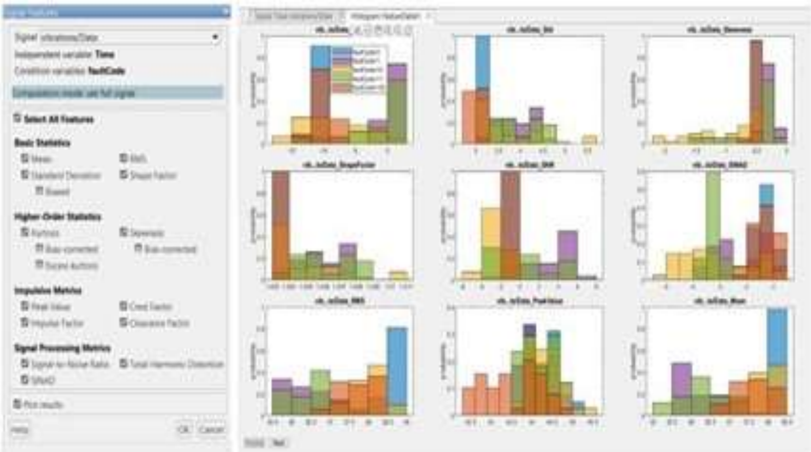


Figure 10 Time Domain Features

After the extraction and plotting of the features in the time domain we extract some more features from the frequency domain. Why do we need more features? We know that machine learning models can work with a large set of features, and when trained with many features, they can do better predictions. However, this is only true if we have useful and distinctive features that can uniquely set different fault types apart. If we have many features but those are not useful or distinctive it would just add to the noise in the model resulting in the inaccurate predictions.

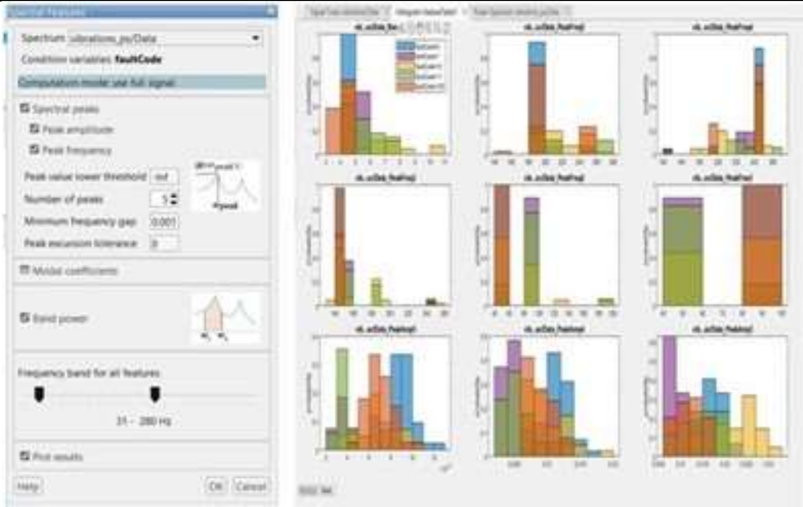


Figure 11 Frequency Domain Features

In the spectral feature extraction, only the first five peaks are considered for the features like amplitude, power etc. because the peaks in the region higher than 280 hz are mostly noise and disturbance and would prove derogatory to the nature of predictions.

2.3.3 Select Useful Features

As mentioned earlier not all extracted features can be beneficial to the machine learning model and selection of useful features is an important step as it determines the accuracy of the prediction made. The features extracted are ranked according to the one-way anova method which is a non-parametric method for testing whether samples originate from the same distribution. It is used for comparing two or more independent samples of equal or different sample sizes. Each feature is ranked and plotted according to the anova score, higher the anova score more useful and distinctive the feature will be for our machine learning model. In our tests the time features rms and mean and the spectral feature peak amp1 make up the top 3 with a score of 147,136 and 137 respectively. For training our model we ignore the features which have a score of less than 10 and thereby exclude the exportation of features like Peakamp3,4,5 which have scores less than 5. We thereby use the top 21 features for training our model and their ANOVA scores range from 147.24 to 20.72.

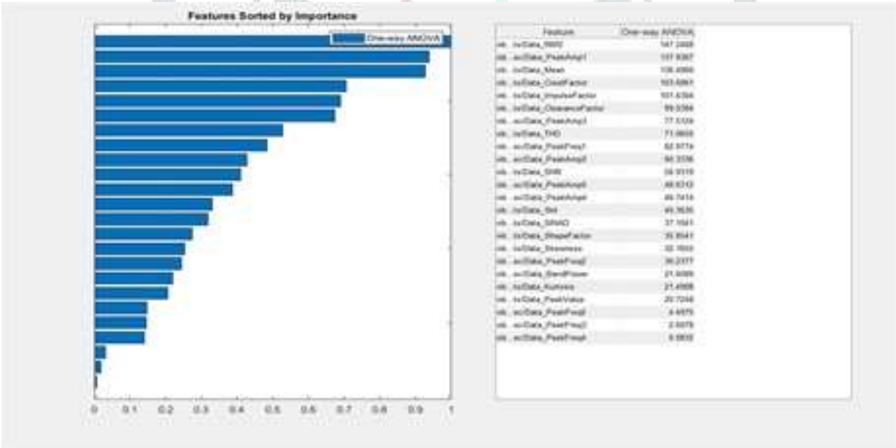


Figure 12 Feature Ranking using ANOVA scores

2.4 Train Machine Learning Model

Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention. The features extracted earlier are imported into the classifier learner app in MATLAB. Our case of machine learning is a multiclass classifier in which we have about 5 classes i.e. engine conditions for the variable (vibrations) to be classified into. Once the features are imported every classifier model available is trained using the parfor function which enables parallel computing and thereby decreasing the amount of time required for training the models by a substantial amount.

Figure 13 Classifier Model Training

As inferred from the above figure, 24 multiclass classifier models were trained using all the 21 features and the accuracy of these models ranged from 92.7% to 26.8%. Even though the range is high, most of the models have accuracy higher than 80 % and three models namely the tree, naïve bayes and ensemble models have the highest accuracy. We selected the tree model for our classifier and a function named "trained model()" is generated.

2.5 Deploy and Integrate

As mentioned earlier the generated function "trained model()", which is the code for the entire classifier training and execution is exported to our workspace and stored in a mat file. The function for feature extraction "diagnostic features()" from the extract features step is also exported and saved in the MATLAB workspace. This function takes in a table or matrix with variables similar to the training data and outputs the feature table which in turn is an input to the classifier function. Hence a display function named "final result()"is created which incorporates both the functions and the output statements by comparing the output of the trained model function which will be in terms of the variable "fault code" and type categorical. Thereby for comparing the output, 5 variables of the type categorical are created with each assigned a fault code and then the output statements are returned accordingly.

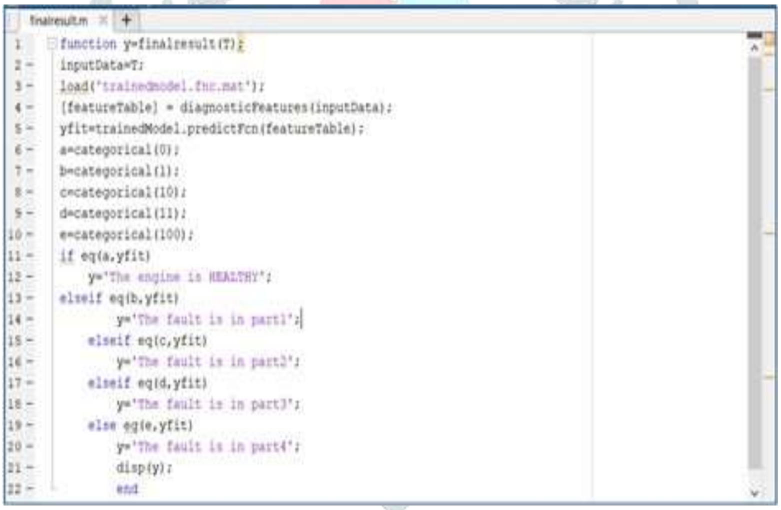


Figure 14 Display Function

This function takes in a table which is an input to the feature extraction function and outputs the condition of the engine.

III. Results

The trained classifier model i.e. the decision tree model predicts the engine health by an accuracy of 92.7%. For testing and validating the classifier model, while training the models 25% of the data was upheld and a confusion matrix of the tree classifier model was plotted for the training dataset.

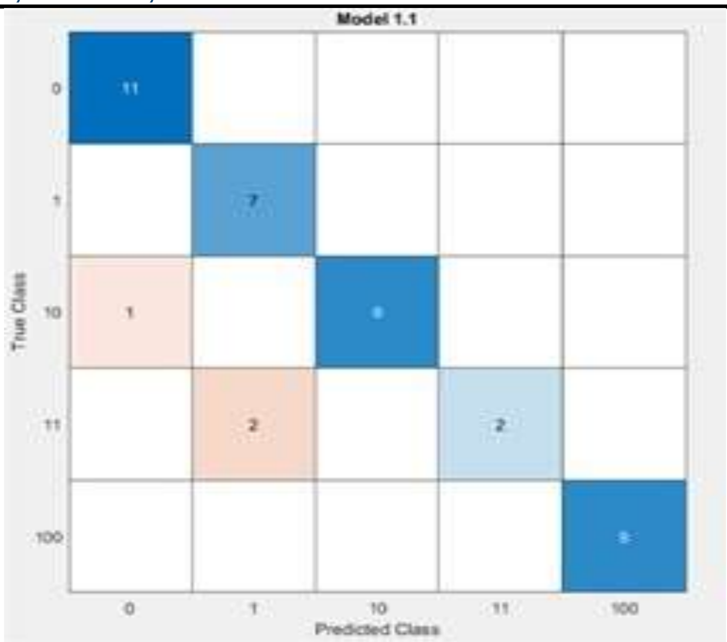


Figure 15 Confusion Matrix

This is a confusion matrix which depicts the accuracy of the trained classifier model. As we can see that the blue boxes represent the accurate predictions, and the red boxes represent the inaccurate predictions.

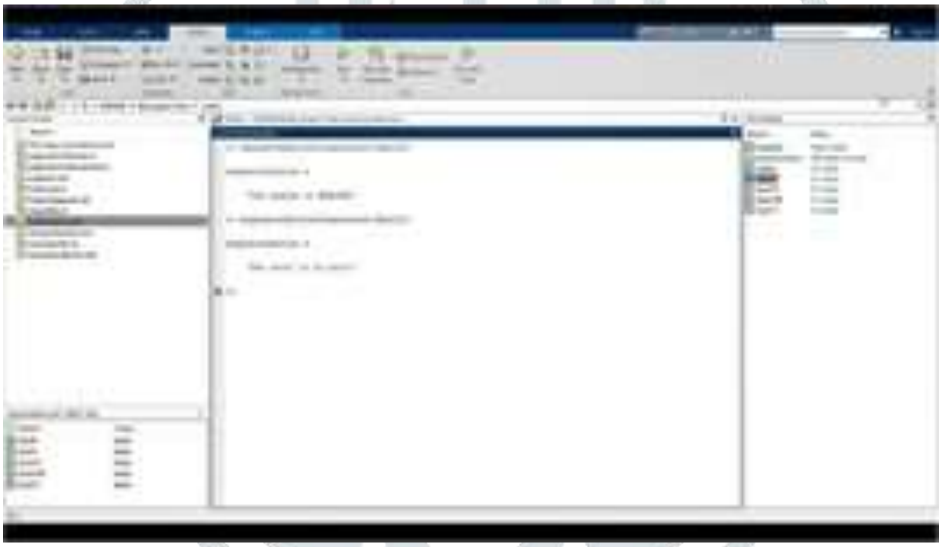


Figure 16 Results of Display Function

A total of 41 sets of buffer were tested and from the above matrix it can be inferred that

- There is no confusion in the trained model to identify the datasets of fault codes (0,1,100) i.e. the engine condition “healthy, fault in part1 and part9” can be easily identified and are distinctive.
- Out of the 10 datasets tested for fault code(10) 1 dataset was wrongly identified as the fault code(0). i.e., the engine condition “fault in part 2” is misidentified as “healthy”. The error probability for this case is low.
- For the dataset of fault code(11), 2 out of 4 were misidentified as a fault code(1). i.e., the engine condition “fault in part3” is wrongly recognized as a “fault in part1” case and the probability of this mistake is higher.

The output function final result() was tested and gave satisfactory result in predicting and outputting the health condition in the MATLAB workspace.

IV. CONCLUSION

This paper presents a model-based approach for the condition monitoring and fault diagnosis of an IC engine using the MATLAB software. The vibration signatures of normally aspirated engine contain valuable information about the health of the engine, so to extract these features a simulink model consisting of an engine and gearbox is constructed and then the vibrational data has been recorded using an accelerometer connection. Healthy and fault induced data are extracted and are further processed to match the input requirements of the MATLAB apps . 21 significant features in both the time and spectral domain have been extracted using the diagnostic feature designer. Which are then used to train the classifier which results in a function, which can be used to predict the nature of new untested data signals.

REFERENCES

[1] P. Saha, Developing Vehicle Sound Packages, Sound and Vibration, October 2011.
[2] S. Delvecchio, P. Bonfiglio, F. Pompoli Vibro-acoustic condition monitoring of Internal Combustion Engines: A critical review of existing techniques, in: Mechanical Systems and Signal Processing 99 (2018) 661–683.
[3] W. Li et al, A study of the noise from diesel engines using the independent component analysis, Mech. Syst. Signal Process. 15 (6) (2001) 1165–1184, 13.
[4] Marwan Madain , Ahmed Al-Mosaiden and Mahmood Al-khassaweneh Fault Diagnosis in Vehicle Engines Using Sound Recognition Techniques, in: IEEE (2010).
[5] S. Yadav, Automatic fault diagnosis of internal combustion engine based on spectrogram and artificial neural network, in: Proceedings of the 10th WSEAS Int. Conference on Robotics, Control and Manufacturing Technology, 2010.
[6] J.D. Wu, C.Q. Chuang, Fault diagnosis of internal combustion engines using visual dot patterns of acoustic and vibration signals, NDT&E Int. 38 (2005) 605–614.
[7] Zhivomirov, H.. (2019). On the development of STFT-analysis and ISTFT- synthesis routines and their practical implementation. TEM Journal. 8. 56-64. 10.18421/TEM81-07.