Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Fault Detection in Gears Using Machine Learning

[1]Deepraj Mulik, [2]Tanmay Biradar, [3]Ayush Khodke, [4]Mrs. Ami Barrot, [5]Dr. P.G. Kulkarni, [6]Mrs. Rashmi Pakankar

[1,2,3]BE Mechanical Engineering Students, Pune Vidyarthi Griha's College of Engineering Pune,Pune
Email: djmulik0208@gmail.com

[4,5,6]Project Guide Mechanical Engineering Department, Pune Vidyarthi Griha's College of Engineering Pune, Pune
Email :principal@pvgcoet.ac.in

--************************----------------------------------

Abstract:

Gears are essential components in various machinery, including automobiles and rotating equipment. When gears fail, it can lead to significant losses. This paper discusses a method for detecting and analysing gear faults to mitigate such risks. To achieve this, a specialized test setup is constructed to examine gear faults experimentally. This setup collects vibration data from both healthy gears and those deliberately made faulty. The collected data is then analysed using the Fast Fourier Transform (FFT) technique, implemented through LabVIEW software. This analysis provides insights into the time and frequency domains of the vibration signals, aiding in identifying fault patterns. Subsequently, an Artificial Neural Network (ANN) is trained using MATLAB with the processed vibration data. This trained ANN forms a system capable of detecting faults in gears based on the patterns learned from the experimental data. In summary, this research presents a comprehensive approach to gear fault detection, utilizing experimental studies, signal processing techniques, and machine learning algorithms.

Keywords **—Gear faults, Machinery, Vibration analysis, Fast Fourier Transform (FFT), LabVIEW, MATLAB, Artificial Neural Network (ANN), Fault detection.**

--************************----------------------------------

1. INTRODUCTION

The gearbox is the primary tool of power transmission. Gears are one of the most significant parts in mechanical transmission systems. It is critical element in a variety of industrial applications, transportation. The gearbox is the primary tool of power transmission. Gears are one of the most significant parts in mechanical transmission systems. It is critical element in a variety of industrial applications, transportation, aerospace, energy, agricultural sectors, wind generation and other fields. Smooth operation and high efficiency of gears are necessary for the normal running of machines. Therefore, gear analysis is an important activity in the field of condition monitoring and fault diagnosis. Early detection of local gear faults in industrial environments is very important to optimize the maintenance schedule and reduce the operating cost of gearbox damage. Failures of the gearbox may cause injury to human beings and important economic losses. To avoid the consequences of any harmful accidents, several techniques are developed in condition monitoring to detect faults as early as possible. Vibration signal analysis is the most common gear detection technique for gear damage detection.

Through gear vibration analysis, tons of features are acquired, and therefore the next step is optimization and classification. In the present work the authors present a review of a spread of diagnosis techniques for gearbox fault identification with reference to vibration analysis. The vibration techniques were developed for two main reasons. The first purpose is to separate the gearbox related signal from other components and to minimize the noise that may mask the gearbox signal, especially in the early stages of the fault. The second purpose is to spot the status of the gearbox, to differentiate the great and therefore the faulty gear and to point the defective components. Nowadays the strain for condition monitoring and vibration analysis aren't any longer limited trying to attenuate the results of machine failures, but to utilize existing resources more effectively.

1.2 Fault detection and diagnosis with vibration analysis

Fault detection and diagnosis through vibration analysis involves identifying a machine's condition or faults based on symptoms, akin to medical diagnosis. Vibration is often seen as a potential sign of gearbox issues. Although gearbox vibration is complex, it provides abundant data, akin to a signal for gearbox conditions.

To detect and diagnose impending failures, a thorough understanding of the evidence regarding the failure mode and methods of collecting and quantifying the evidence is necessary. While some component faults can be easily detected physically using methods like microscopy or x-ray, these methods often require component removal and can be impractical for operational machinery. Hence, non-intrusive fault detection methods, especially those based on vibration analysis, are commonly employed for routine monitoring.

Modern gear diagnostic techniques mainly rely on analysing vibration signals collected from the gearbox casing. The primary aim is to detect and classify faults early and monitor their progression to estimate residual machine life and plan maintenance effectively. The key components of geared vibration spectra typically include gear meshing frequency and its harmonics, along with sidebands indicating modulation phenomena. An increase in the number and amplitude of such sidebands can signal a fault condition, and their spacing provides additional information about the fault source.

Identifying the source and detecting faults from vibration signals associated with rotating components such as gears, rotors, bearings, and couplings depend on several factors including rotational speed, background noise level, transducer location, load sharing characteristics, and dynamic interactions with other components.

Mechanical vibration can arise from various sources including looseness, unbalance, defective bearings, gear inaccuracies, misalignments, critical speeds, drive belt issues, reciprocating forces, oil, and friction whirl, aerodynamic or hydrodynamic forces, resonances, bent rotor shafts, and defective rotor bars.

1.3 Methodology of Experiment:

1. Gear Defect Study: Examine various gear tooth defects.
2. Literature Review: Review fault diagnosis techniques.
3. Vibration Signal Collection: Gather signals from healthy and faulty gears.
4. FFT Analysis: Apply FFT to analyse signals in frequency domain.
5. ANN Implementation: Train ANN using vibration data.
6. Result Comparison: Compare FFT and ANN outcomes.

2. EXPERIMENTAL SETUP:

The setup consists of a motor, gears mounted on two shafts held together by bearings and rotated by motor's shaft itself. Accelerometer has been mounted on the bearings fixture and then further connected to the data acquisition system which is then connected to the laptop where the readings are taken using NI LabVIEW. The VFD is used to control the speed of the motor. Do vibrational analysis of healthy gear using Fast Fourier Transformer (FFT). In lab view first we get vibration graph with respect to time domain then we will convert into frequency domain. Then we will compare the graph of both gear

Fig. No1 Top view of setup

Fig. No2 Experimental Setup

Fig. No 3 VFD Box

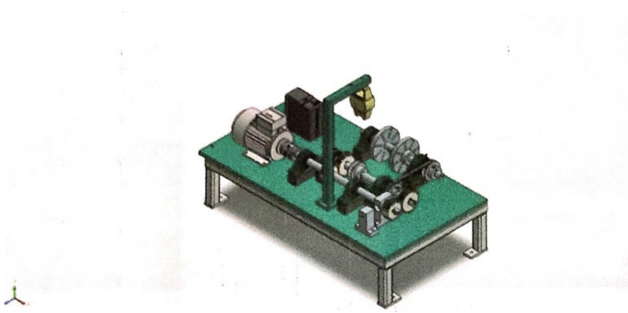


Fig. No 4 CAD Model

Fig. No 5 DAQ Box

3. ANSYSANALYSIS OF SPUR GEARMESHING

3.1Introduction:

Spur gears play a vital role in mechanical power transmission systems due to their simplicity and reliability. Analysing the meshing behaviour of spur gears is essential for optimizing system performance and longevity. Ansys software offers robust tools for conducting in-depth analyses of gear meshing phenomena.

3.2 Objective:

The main goal of this analysis is to assess contact patterns, stress distribution, and performance characteristics of spur gears under varying conditions using Ansys. Through advanced simulation techniques, engineers can refine gear designs, prevent potential failures, and improve system efficiency and reliability.

3.3 Methodology:

1. Geometry Creation: Develop 3D models of spur gears in Ansys, ensuring accurate representation of tooth profiles and gear dimensions.
2. Mesh Generation: Create high-quality meshes to accurately capture contact behaviour while optimizing computational efficiency.
3. Material Properties: Assign material properties like Young's modulus and yield strength to simulate mechanical behaviour accurately.
4. Boundary Conditions: Apply realistic operating conditions, including rotational speeds, torque loads, and environmental factors.
5. Contact Analysis: Utilize advanced contact algorithms to analyse gear tooth interaction, contact patterns, and pressure distribution.

6. Stress Analysis: Employ Finite Element Analysis to compute stress distribution, identifying critical areas prone to failure.
7. Performance Evaluation: Assess transmission efficiency, tooth load distribution, and gear deflection to evaluate system performance.

3.4 Results and Discussion:

The Ansys analysis provides insights into spur gear behaviour under different conditions. Contact analysis reveals efficient power transmission, while stress analysis identifies high-stress areas for design optimization. Key results include total deformation, maximum principle elastic strain, and maximum principal stress values.

3.5 Conclusion:

The Ansys analysis showcases the software's capability in evaluating mechanical components. By using advanced simulation techniques, engineers can enhance gear designs, improve reliability, and optimize system efficiency.

4. MACHINE LEARNING CLASSIFICATION MODELS

The use of Isolation Forest and Random Forest for gear fault detection is a promising approach in the field of mechanical fault diagnosis. Both algorithms have been successfully applied in various studies to detect faults in manufacturing equipment and predict equipment failures.

4.1 Isolation Forest:

Isolation Forest is an unsupervised anomaly detection algorithm that operates on the principle that anomalies are few and different, thus easier to isolate. It works by recursively partitioning the data space and computing an anomaly score based on the path length required to isolate a data point. Anomalies tend to have shorter paths as they are easier to isolate. This algorithm is suitable for scenarios where labelled fault data is scarce and is effective in high-dimensional spaces with minimal parameter tuning.

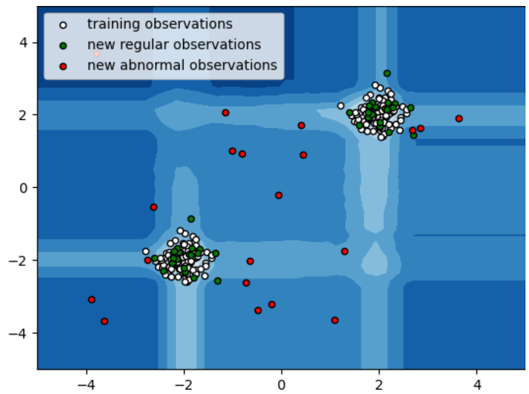


Fig. No 6 Isolation Forrest

4.2 Random Forest:

Random Forest is a supervised learning algorithm used for classification and regression tasks. It combines predictions from multiple decision trees to make a final decision, introducing diversity among the trees through bootstrap aggregating and feature randomness. This algorithm is ideal for supervised scenarios with sufficient labelled data, providing high accuracy and interpretability through feature importance metrics.

Fig. No 7 Random Forrest

4.3 Comparison and Conclusion:

The choice between Isolation Forest and Random Forest for gear fault detection depends on the data availability and the specific requirements of the fault detection task. Isolation Forest is ideal for unsupervised anomaly detection when labelled fault data is not available, while Random Forest excels in supervised classification with labelled data, providing high accuracy and interpretability. Both algorithms have their unique strengths and can be effectively used for gear fault detection, depending on the specific context and data availability.

5. MACHINE LEARNING MODEL PROGRAMMING AND RESULT

In [2]:

```
import pandas as pd

df = pd.read_excel(r'C:\Users\NVB3KOR\Documents\My Project\Tanmay_project\bfull 1000_26\0
0.xlsx')
```

In [3]:

```
df.size
```

Out[3]:

```
4000
```

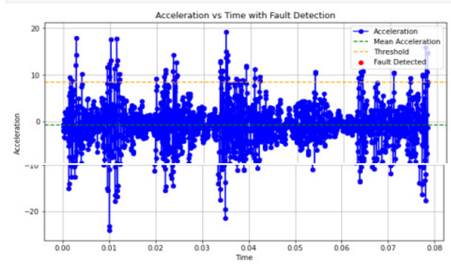
In [4]:

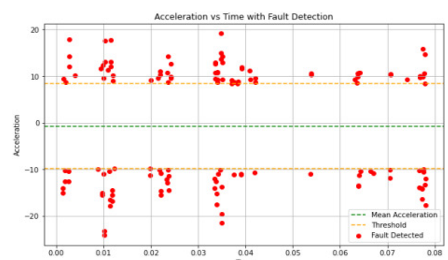
```
import pandas as pd

# Create a DataFrame from your data
# data = {
#     'Acceleration_Time*': [0.000000, 0.000039, 0.000078, 0.000117, 0.000156],
#     'Acceleration': [-2.128012, -1.951796, -1.272800, -0.584086, 0.395377]
# }
# df = pd.DataFrame(data)

# Calculate mean and standard deviation of acceleration
mean_acceleration = df['Acceleration'].mean()
std_acceleration = df['Acceleration'].std()

# Define a threshold (e.g., 2 standard deviations from the mean)
threshold = 2 * std_acceleration

# Detect faults
faults = df[abs(df['Acceleration'] - mean_acceleration) > threshold]

print("Mean Acceleration:",  mean_acceleration)
print("Standard Deviation of Acceleration:", std_acceleration)
print("Threshold for Fault Detection:", threshold)
print("Detected Faults:")
print(faults)
```

```
Mean Acceleration: -0.7442729584085699
Standard Deviation of Acceleration: 4.5552184217541365
Threshold for Fault Detection: 9.110436843508273
Detected Faults:
      Acceleration_Time*  Acceleration
34              0.001328    -15.099507
35              0.001367    -14.094204
38              0.001484      9.370030
47              0.001836    -12.647134
48              0.001875    -10.220643
...                  ...           ...
1994            0.077891     14.707655
1995            0.077930      8.436500
1997            0.078008    -13.377656
1998            0.078047    -17.755387
1999            0.078086    -11.958427

[133 rows x 2 columns]
```

In [5]:

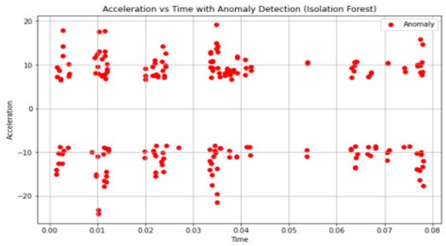
```
import pandas as pd
import matplotlib.pyplot as plt


import pandas as pd
import matplotlib.pyplot as plt
import numpy as np


# Calculate mean and standard deviation of acceleration
mean_acceleration = df['Acceleration'].mean()
std_acceleration = df['Acceleration'].std()

# Define a threshold (e.g., 2 standard deviations from the mean)
threshold = 2 * std_acceleration

# Detect faults
faults = df[abs(df['Acceleration'] - mean_acceleration) > threshold]

# Convert DataFrame columns to numpy arrays
time = np.array(df['Acceleration_Time*'])
acceleration = np.array(df['Acceleration'])

# Plot the data
plt.figure(figsize=(10, 6))
plt.plot(time, acceleration, marker='o', linestyle='-', color='b', label='Acceleration')
plt.scatter(faults['Acceleration_Time*'], faults['Acceleration'], color='r', label='Faul
t Detected')

# Plot mean line and threshold lines
plt.axhline(mean_acceleration, color='g', linestyle='--', label='Mean Acceleration')
plt.axhline(mean_acceleration + threshold, color='orange', linestyle='--', label='Thresh
old')
plt.axhline(mean_acceleration - threshold, color='orange', linestyle='--')

plt.xlabel('Time')
plt.ylabel('Acceleration')
plt.title('Acceleration vs Time with Fault Detection')
plt.legend()
plt.grid(True)
plt.show()


# Plot the data
plt.figure(figsize=(10, 6))
# plt.plot(time, acceleration, marker='o', linestyle='-', color='b', label='Acceleration'
)
plt.scatter(faults['Acceleration_Time*'], faults['Acceleration'], color='r', label='Faul
t Detected')

# Plot mean line and threshold lines
plt.axhline(mean_acceleration, color='g', linestyle='--', label='Mean Acceleration')
plt.axhline(mean_acceleration + threshold, color='orange', linestyle='--', label='Thresh
old')
plt.axhline(mean_acceleration - threshold, color='orange', linestyle='--')

plt.xlabel('Time')
plt.ylabel('Acceleration')
plt.title('Acceleration vs Time with Fault Detection')
plt.legend()
plt.grid(True)
plt.show()
```





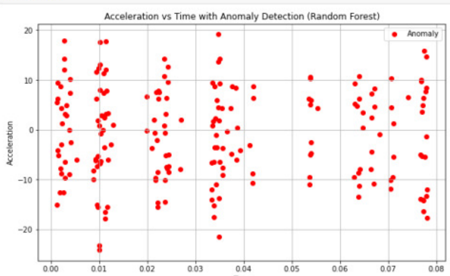
In [6]:

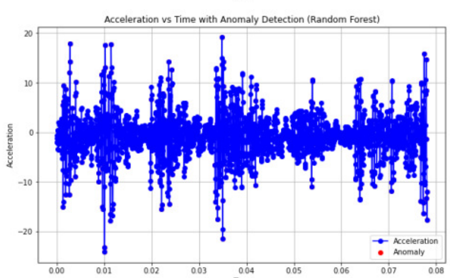
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.ensemble import IsolationForest


# Reshape the data
X = df['Acceleration'].values.reshape(-1, 1)

# Create the Isolation Forest model
model = IsolationForest(contamination=0.1)  # Adjust the contamination parameter as neede
d

# Fit the model
model.fit(X)
# Convert DataFrame columns to numpy arrays

# Predict outliers
outliers = model.predict(X)

# Convert DataFrame columns to numpy arrays
time = np.array(df['Acceleration_Time*'])
acceleration = np.array(df['Acceleration'])

# Plot the data
plt.figure(figsize=(10, 6))
# plt.plot(time, acceleration, marker='o', linestyle='-', color='b', label='Acceleration'
)
plt.scatter(df['Acceleration_Time*'][outliers == -1], df['Acceleration'][outliers == -1]
, color='r', label='Anomaly')

plt.xlabel('Time')
plt.ylabel('Acceleration')
plt.title('Acceleration vs Time with Anomaly Detection (Isolation Forest)')
plt.legend()
plt.grid(True)
plt.show()
```

In [9]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestRegressor

# Assume df is your DataFrame and it has 'Acceleration' and 'Acceleration_Time*' columns

# Prepare the data
X = df['Acceleration_Time*'].values.reshape(-1, 1)
y = df['Acceleration'].values

# Create the Random Forest model
model = RandomForestRegressor(n_estimators=100)

# Fit the model
model.fit(X, y)

# Predict values
y_pred = model.predict(X)

# Compute prediction errors
errors = np.abs(y - y_pred)

# Determine a threshold for anomalies (you can adjust this threshold)
threshold = np.percentile(errors, 90)  # E.g., anomalies are those in the top 10% of erro
rs

# Identify anomalies
anomalies = errors > threshold

# Convert DataFrame columns to numpy arrays for plotting
time = np.array(df['Acceleration_Time*'])
acceleration = np.array(df['Acceleration'])

# Plot the data
plt.figure(figsize=(10, 6))
# plt.plot(time, acceleration, marker='o', linestyle='-', color='b', label='Acceleration'
)
plt.scatter(time[anomalies], acceleration[anomalies], color='r', label='Anomaly')

plt.xlabel('Time')
plt.ylabel('Acceleration')
plt.title('Acceleration vs Time with Anomaly Detection (Random Forest) - Faults detected'
)
plt.legend()
plt.grid(True)
plt.show()


# Plot the data
plt.figure(figsize=(10, 6))
plt.plot(time, acceleration, marker='o', linestyle='-', color='b', label='Acceleration')
plt.scatter(time[anomalies], acceleration[anomalies], color='r', label='Anomaly')

plt.xlabel('Time')
plt.ylabel('Acceleration')
plt.title('Acceleration vs Time with Anomaly Detection (Random Forest)')
plt.legend()
plt.grid(True)
plt.show()
```





In [ ]:

6. CONCLUSION

In conclusion, this research highlights the potential of machine learning techniques in automating bearing fault detection. By leveraging vibration signals and mathematical features, the proposed approach offers a data-driven solution for early fault detection and predictive maintenance. Future research directions include the integration of additional sensor data and the exploration of advanced machine learning algorithms for enhanced fault diagnosis.

ACKNOWLEDGMENT

We are thankful to our beloved HOD Mr. M.M. Bhoomkar sir for his guidance and cooperation for providing us with the DAQ setup and Labview Software. We are thankful to our project guides Prof. Ami Barrot ma'am and Prof. P.G. Kulkarni sir for providing us the guidance for the completion of our project.

REFERENCES

[1] "VIBRATION ANALYSIS TECHNIQUES FOR GEARBOX DIAGNOSTIC: A REVIEW By Amit Aherwar. Md. Saifullah Khalid
[2] "Detection of Fault in Gear Box System using Vibration Analysis Method" By LS.Dhamande, A.C.Pawar and V.I Suryawanshi
[3] "A Review of Gearbox Condition Monitoring Based on vibration Analysis Techniques Diagnostics and Prognostics by Abdulrahman S. Sait, Yahya 1. Sharal Eldeen
[4] "Vibration Analysis and Control in Mechanical Structures and Wind Energy Conversion Systems" by Paolo Gardonio and Leonardo Lanari
[5] "Detection of Fault in Gearbox System Using Vibration Analysis Method" by Saurabh S. Shahapurkar and Others
[6] "Dynamic Modelling and Vibration Analysis for Gear Tooth Crack Detection" by Omar Dawood Mohammed
[7] "Spur Gear Fault Diagnosis using a Multilayer Gated Recurrent Unit Approach with Vibration Signal" by Ying Taol and Others

