

**Homework 3 - ENME 691**

Report - Group 3

March 4th, 2024

Brian O’Malley

Ashwin Sudarshan

Sai Dinesh Gelam

# 

**Table of Contents**

1. Introduction
2. Solution Methodology
3. Results
4. Discussion
5. Conclusions
6. Appendix
7. **Introduction**
8. **Background information and description about the problem**

**Rotor-Bearing System:** This system typically consists of a rotating shaft (the rotor) supported by bearings. Bearings are critical components that enable smooth rotation while minimizing friction and wear. Monitoring the health of these bearings is essential for ensuring the reliability and efficiency of rotating machinery.

**Unbalance Defects:** Unbalance occurs when the center of mass of a rotating component (such as the rotor in this case) is not aligned with the axis of rotation. This can lead to vibration and potential damage to the bearings and other components of the system. In the testbed, unbalanced defects were induced by adding screws to the disk of the shaft.

In this case, we have 2 levels of unbalance. Unbalance 1 being the case with 1 screw and Unbalance 2 representing the case with 2 screws added to the disk.

**Vibration Analysis:** Vibration signals provide valuable insights into the condition of rotating machinery. Changes in vibration patterns can indicate various faults or anomalies, including unbalance, misalignment, bearing defects, and more. Accelerometers are commonly used to measure vibration levels in rotating machinery.

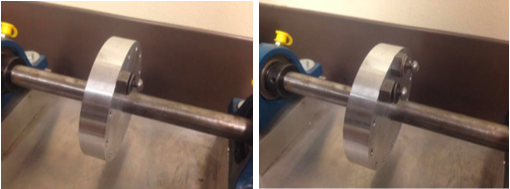
**Data Collection:** Vibration data were collected using an accelerometer mounted on the bearing block. The data were sampled at a high rate of 2560 Hz to capture detailed vibration characteristics. This high sampling rate is necessary to capture the fast-changing vibration signals accurately

1. **Raw data description**

A rotor-bearing testbed (Figure 1) was built to analyze the health condition of the shaft with unbalance defects.The rotating speed of the test-bed is 20 Hz, and screws were added on the disk of the shaft to induce unbalance defects as shown in Figure 2. An accelerometer was mounted on the bearing block and to measure the bearing's vibrations. The vibration data were collected under balanced and unbalanced conditions at a rate of 2560 Hz



**Figure 1**: Healthy rotor-bearing test beds



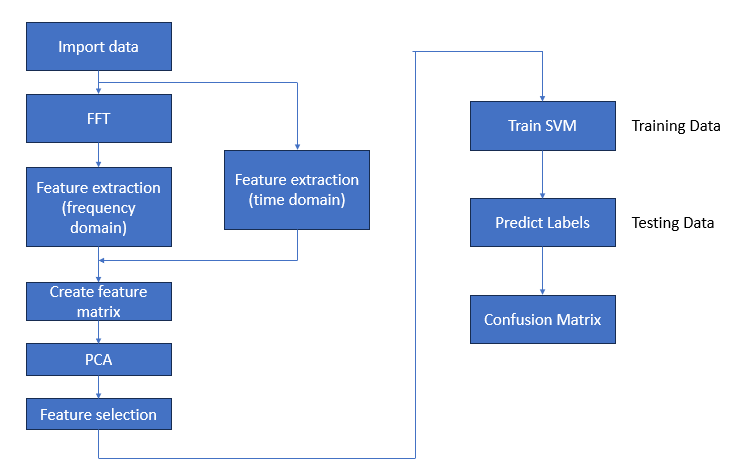
**Figure 2:** Unbalanced level 1(left) and level 2 (right) rotor-bearing test beds.

The resultant data was split into training and testing sets for the machine learning model being developed as a part of this work. Sixty tests were allocated for the training set and thirty for the testing set. The training set was composed of 40 faulty bearing tests (20 each for level 1 and 2) and 20 healthy bearing tests while the proportion of balanced to unbalanced tests in the testing set was unknown.

1. **Solution Methodology**

**a. Flowchart**

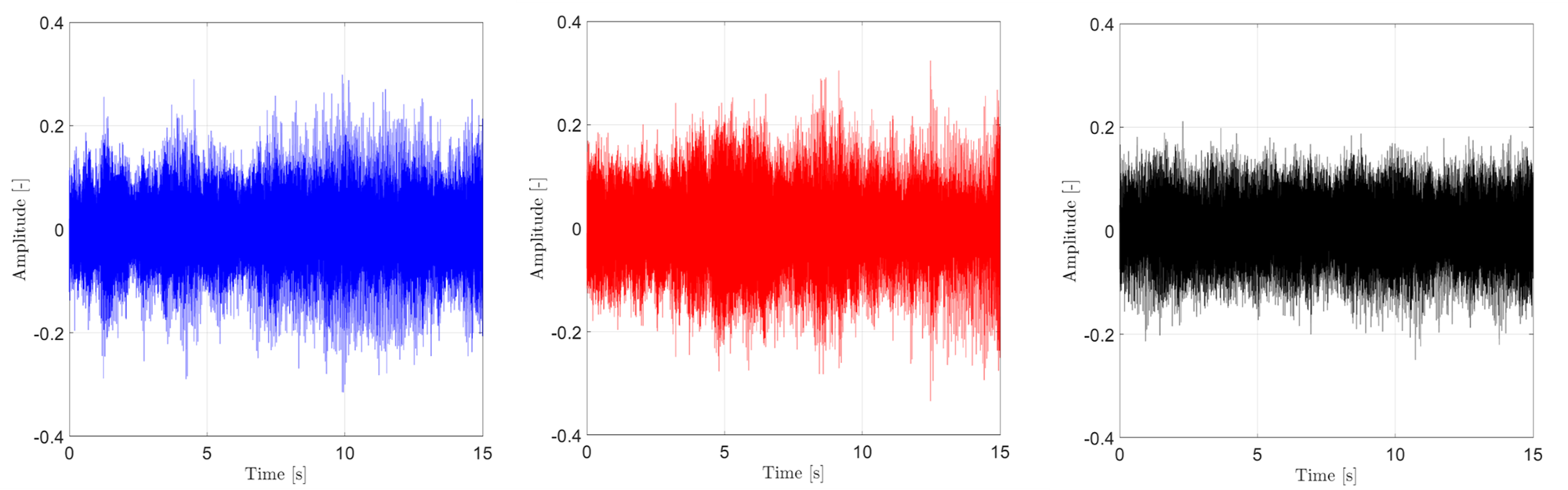
The workflow of the entire process including the data analysis, PCA, feature extraction, and model development can be seen in Figure 3.



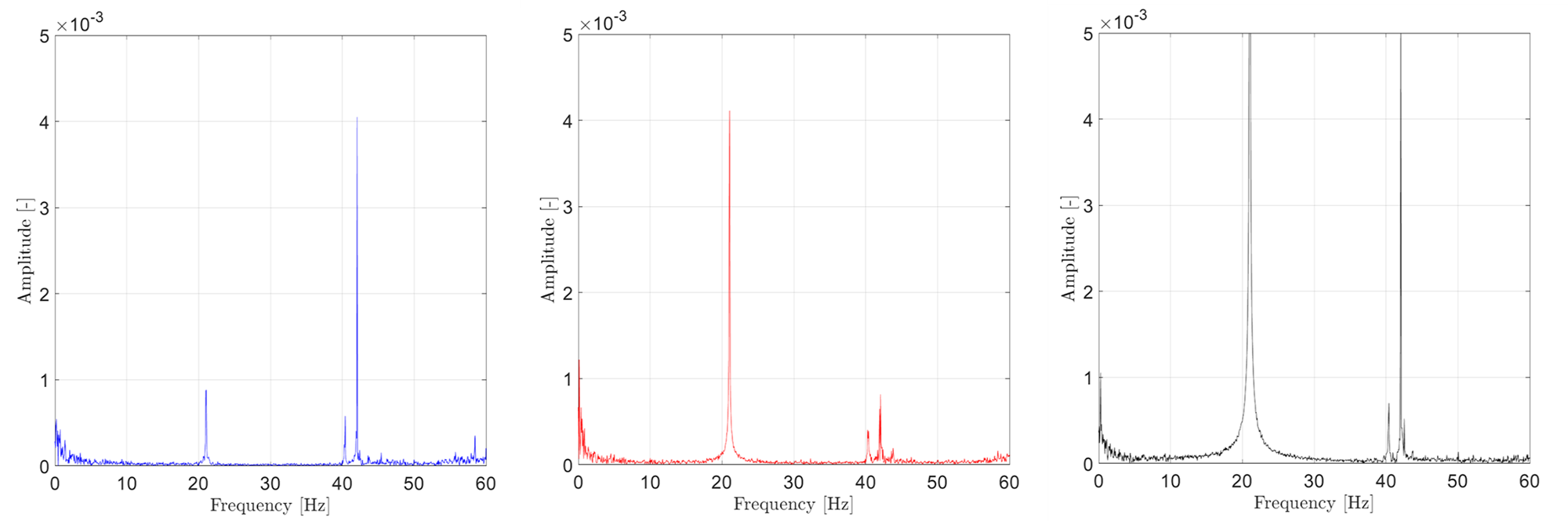
**Figure 3:** Workflow.

**b. Data Analysis**

The solution process begins with loading the vibration data from the healthy, faulty, and testing datasets. Time domain analysis is conducted by plotting healthy and both levels of faulty data sets across the data collection period, but the difference in magnitude is hard to discern as shown in Figures 4. The data is transformed using the fast fourier transform as shown in Figures 5, where the peak amplitudes are visible. The amplitude of the signal is scaled down by ½ the number of points in each signal and results are only shown up to 60 Hz. The difference in amplitude between the healthy and the two unbalanced samples is now clear. The same data analysis process is applied to the test data.



**Figure 4:** Time domain of the healthy (left), unbalanced level 1 (middle), and unbalanced level 2 (right) samples.

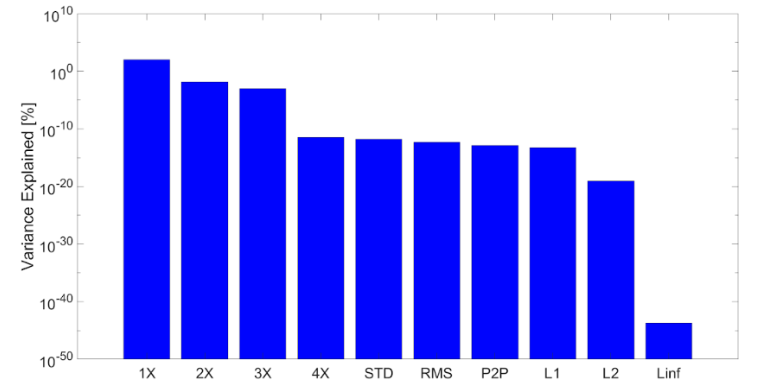


**Figure 5**: Frequency domain of the healthy (left), unbalanced level 1 (middle), and unbalanced level 2 (right) samples.

* 1. **Principal component analysis**

The most relevant sensor measurements were selected via Principal

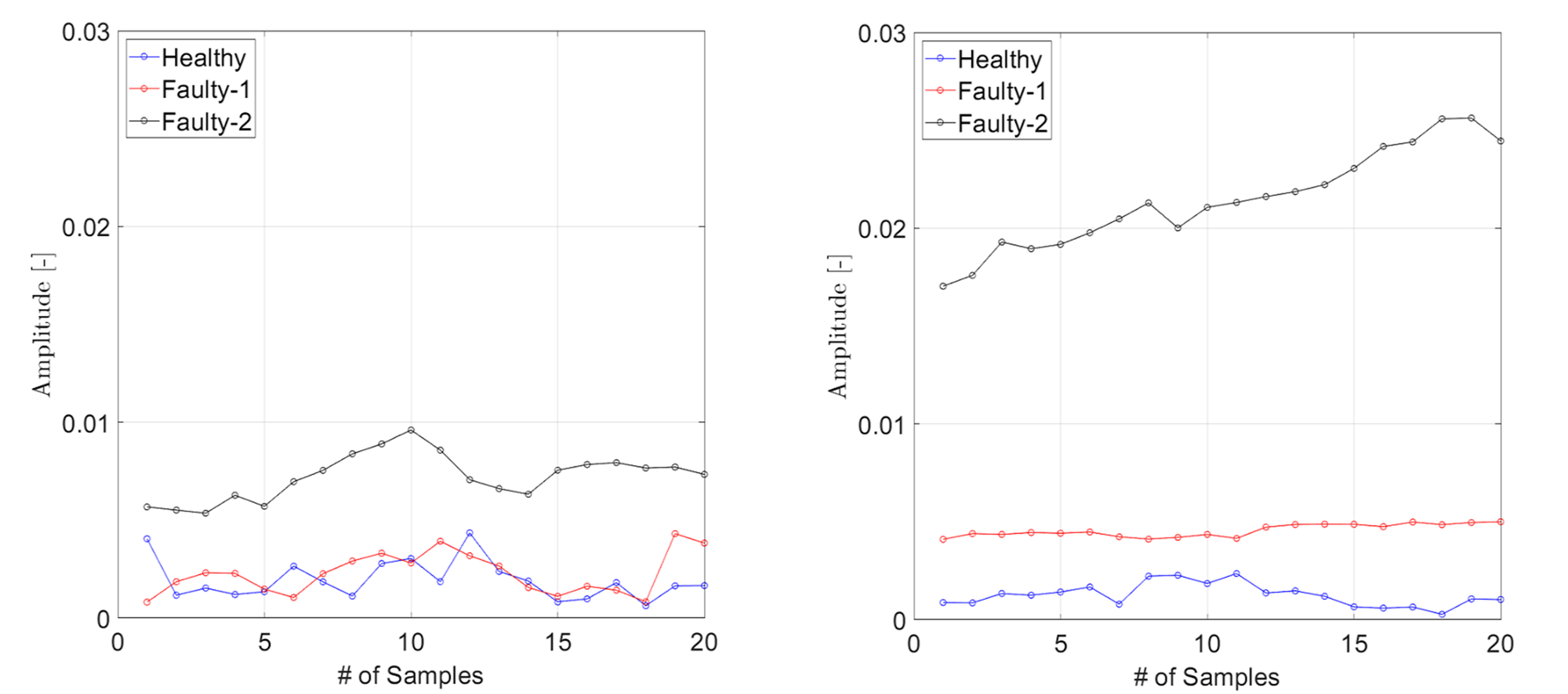
Components Analysis (PCA). From Figure 6, it can be observed that the variables which are of most significance are the 1X Harmonic and 2X Harmonic, closely followed by 3X Harmonic.We find from the figure that Rms and Std hardly have much significance. Hence, the two features we selected were 1X harmonic and 2X harmonic.

****

**Figure 6 :** Principal component analysis

* 1. **Feature Extraction**

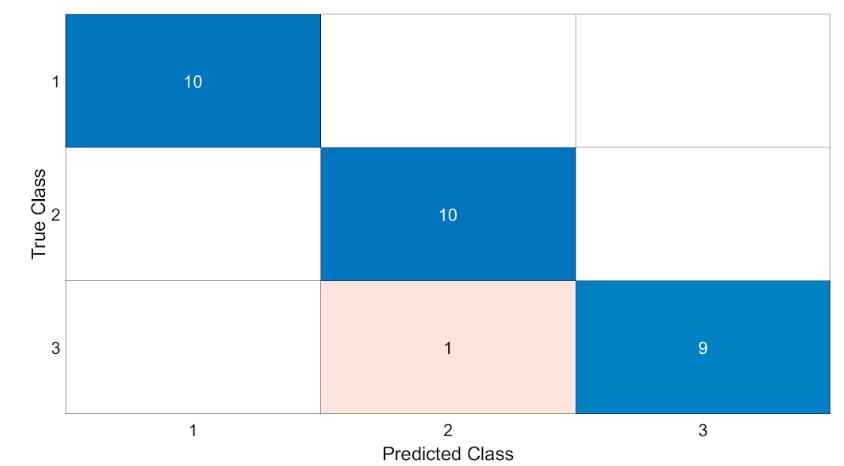
After carrying out PCA, the amplitude of the signal at the 1X harmonic (20 Hz) is selected by looping through the data between 15 and 25 Hz and returning the maximum value. All of the peaks (like those shown in Figures 5) can be grouped together and trends can be established. As shown in Figure 7, the unbalanced level 2 data displays an amplitude between 0.017 and 0.026 as compared to the unbalanced level 1 data which shows an amplitude around 0.004. Meanwhile, the healthy data has magnitudes in the range of 2.8e-4 to 2.4e-3. The same process is applied to the test data.Another feature that has been extracted is the amplitude of the signal at the 2X Harmonic. The unbalanced level 2 displays an amplitude ranging from 0.004 to 0.013 compared to the amplitude of the unbalanced level 1 which ranges from 0.006 to 0.011. Meanwhile, the healthy data has amplitudes ranging from 0.002 to 0.015.



**Figure 7** : Feature of amplitude of the 1X (left) and 2X (right) harmonic

1. **Results and Discussion**

The confusion matrix reveals the SVM model's strong ability to distinguish between healthy and faulty conditions, as well as its capability to differentiate between unbalanced levels. The minimal misclassification (1/30 cases) between Faulty Level 1 and Level 2 indicates the model's sensitivity to changes in harmonic amplitudes, underscoring the effectiveness of our feature selection.



**Figure 8** : Confusion matrix

1. **Conclusions**

The implementation of SVM utilizing 1X and 2X harmonic amplitudes as features has demonstrated a significant potential for accurately assessing the health condition of rotor-bearing systems. Our analysis not only confirms the relevance of these harmonics in fault diagnosis but also highlights SVM's capability to effectively classify system states with high precision. The comparative analysis with logistic regression underscores the advanced modeling power of SVM for complex classification tasks in industrial applications. Going forward, expanding the feature set and employing a more diverse dataset will be crucial for refining the model's predictive performance and extending its applicability to a wider range of fault conditions.

1. **Appendix**

%ENME 691 - Industrial AI

%HW3

% Spring 2024

clc;clear;close;

%% Top Matter

format long; format compact;

set(0,'defaultTextInterpreter','latex'); %trying to set the default

sz = 60; %Marker Size

szz = sz/35;

lw = 1;

ms=8;

fs=25;

txtsz = 30;

txtFactor = 0.8;

ax = [0.9,1.4,0.0,2.0];

loc = 'southwest';

pos = [218,114,1478,796];

plotFlag = false;

% txtsz = 24;

%% 1: Set file path

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Specify the location of the libsvm/matlab folder %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

dir\_lib = 'F:\NOTES\Classes\Industrial AI\HW3';

cd(dir\_lib)

load("FeatMat\_train.mat");

load("FeatMat\_test.mat");

load("cMatTest.mat");

dir\_lib = 'F:\NOTES\Classes\Industrial AI\HW3\libsvm\matlab';

cd(dir\_lib)

%% 2: Import data

%%%%%%%%%%%%%%%%%%%

% Write your code %

%%%%%%%%%%%%%%%%%%%

dirTrainH = "F:\NOTES\Classes\Industrial AI\HW2\HW2\Homework 2\Training\Training\Healthy";

dirTrainF1 = "F:\NOTES\Classes\Industrial AI\HW3\Training\Faulty\Unbalance 1";

dirTrainF2 = "F:\NOTES\Classes\Industrial AI\HW3\Training\Faulty\Unbalance 2";

dirTest = "F:\NOTES\Classes\Industrial AI\HW2\HW2\Homework 2\Testing\Testing";

filesH = dir(dirTrainH +'\\*.txt');

filesF1 = dir(dirTrainF1 +'\\*.txt');

filesF2 = dir(dirTrainF2 +'\\*.txt');

filesT = dir(dirTest +'\\*.txt');

%set general use values

haftspeed = 20;% Hz

Fs = 2560; % Sampling frequency

T = 1/Fs; % Sampling period

L = 38400; % Length of signal

time = (0:L-1)\*T; % Time vector

FsRange = Fs/L\*(0:L-1);

%set the range of interest (1x, 2x, 3x, 4x, harmonics)

low1X = find(FsRange==15);

up1X = find(FsRange==25);

low2X = find(FsRange==35);

up2X = find(FsRange==45);

low3X = find(FsRange==55);

up3X = find(FsRange==65);

low4X = find(FsRange==75);

up4X = find(FsRange==85);

%% 3: Feature extraction / FFT

%%%%%%%%%%%%%%%%%%%

% Write your code %

%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%

%%%healthy data%%%%

%%%%%%%%%%%%%%%%%%%

for i=1:length(filesH)

tempStr = dirTrainH +"\"+filesH(i).name;

trainingDataH{i} =readtable(tempStr);

%get additional data

t=abs(fft(trainingDataH{i}.Date3\_26\_2014));

peakToPeakH(i) = peak2peak(t);

L1H(i) = norm(t,1);

L2H(i) = norm(t,2);

LinfH(i) = norm(t,inf);

% plot(FsRange,fft(trainingDataH{i}.Date3\_26\_2014),'o')

transFFT\_H{i} = t; clear t;

end

for i=1:20

temp = transFFT\_H{i}/L\*2;

healthy1X(i) = max(temp(low1X:up1X));

healthy2X(i) = max(temp(low2X:up2X));

healthy3X(i) = max(temp(low3X:up3X));

healthy4X(i) = max(temp(low4X:up4X));

rmsH(i) = rms(temp);

stdH(i) = std(temp);

clear temp;

end

%%%%%%%%%%%%%%%%%%%

%%%faulty 1 data%%%

%%%%%%%%%%%%%%%%%%%

for i=1:length(filesF1)

tempStr = dirTrainF1 +"\"+filesF1(i).name;

trainingDataF1{i} =readtable(tempStr);

t=abs(fft(trainingDataF1{i}.Date3\_26\_2014));

%get additional data

peakToPeakF1(i) = peak2peak(t);

L1F1(i) = norm(t,1);

L2F1(i) = norm(t,2);

LinfF1(i) = norm(t,inf);

transFFT\_F1{i} = t; clear t;

end

for i=1:20

temp = transFFT\_F1{i}/L\*2;

faulty11X(i) = max(temp(low1X:up1X));

faulty12X(i) = max(temp(low2X:up2X));

faulty13X(i) = max(temp(low3X:up3X));

faulty14X(i) = max(temp(low4X:up4X));

rmsF1(i) = rms(temp);

stdF1(i) = std(temp);

clear temp;

end

%%%%%%%%%%%%%%%%%%%

%%%faulty 2 data%%%

%%%%%%%%%%%%%%%%%%%

for i=1:length(filesF2)

tempStr = dirTrainF2 +"\"+filesF2(i).name;

trainingDataF2{i} =readtable(tempStr);

t=abs(fft(trainingDataF2{i}.Date3\_26\_2014));

%get additional data

peakToPeakF2(i) = peak2peak(t);

L1F2(i) = norm(t,1);

L2F2(i) = norm(t,2);

LinfF2(i) = norm(t,inf);

transFFT\_F2{i} = t; clear t;

end

for i=1:20

temp = transFFT\_F2{i}/L\*2;

faulty21X(i) = max(temp(low1X:up1X));

faulty22X(i) = max(temp(low2X:up2X));

faulty23X(i) = max(temp(low3X:up3X));

faulty24X(i) = max(temp(low4X:up4X));

stdF2(i) = std(temp);

rmsF2(i) = rms(temp);

clear temp;

end

%%%%%%%%%%%%%%%%%%%

%%%testing data%%%

%%%%%%%%%%%%%%%%%%%

for i=1:length(filesT)

tempStr = dirTest +"\"+filesT(i).name;

testData{i} =readtable(tempStr);

t=abs(fft(testData{i}.Date3\_26\_2014));

%get additional data

peakToPeakT(i) = peak2peak(t);

L1T(i) = norm(t,1);

L2T(i) = norm(t,2);

LinfT(i) = norm(t,inf);

transFFT\_T{i} = t; clear t;

end

%pick out the peak for the Testing data

for i=1:30

temp = transFFT\_T{i}/L\*2;

testing1X(i) = max(temp(low1X:up1X));

testing2X(i) = max(temp(low2X:up2X));

testing3X(i) = max(temp(low3X:up3X));

testing4X(i) = max(temp(low4X:up4X));

stdT(i) = std(temp);

rmsT(i) = rms(temp);

clear temp;

end

%%%%%%%%%%%%%%%%%%%

%%%group data%%%

%%%%%%%%%%%%%%%%%%%

%all components

tempH = [healthy1X',healthy2X',healthy3X',healthy4X',stdH',rmsH',peakToPeakH',L1H',L2H',LinfH'];

tempF1 = [faulty11X',faulty12X',faulty13X',faulty14X',stdF1',rmsF1',peakToPeakF1',L1F1',L2F1',LinfF1'];

tempF2 = [faulty21X',faulty22X',faulty23X',faulty24X',stdF2',rmsF2',peakToPeakF2',L1F2',L2F2',LinfF2'];

tempT = [testing1X',testing2X',testing3X',testing4X',stdT',rmsT',peakToPeakT',L1T',L2T',LinfT'];

%load components into feature matrix

FeatMat\_train = [tempH;tempF1;tempF2];

FeatMat\_test = tempT;

cMatTest = [1\*ones(10,1);2\*ones(10,1);3\*ones(10,1)];

[coeff,score,latent,tsquared,explained,mu] = pca(FeatMat\_train);

%clear low impact components (only need 1st and 2nd harmonic)

FeatMat\_train(:,3:end) =[];

FeatMat\_test(:,3:end) = [];

save("FeatMat\_train.mat","FeatMat\_train");

save("FeatMat\_test.mat","FeatMat\_test");

save("cMatTest.mat","cMatTest");

%% 4: Plot signals, features

%%%%%%%%%%%%%%%%%%%

% Write your code %

%%%%%%%%%%%%%%%%%%%

%plot time

figure(1)

grid on; hold on; box on;

axis square;

ax=gca;

ax.FontSize = fs;

% pbaspect([2 1 1])

xlim([0 15]);

ylim([-0.4 0.4]);

yticks([-0.4 -0.2 0 0.2 0.4]);

temp = table2array(trainingDataH{1});

plot(time,temp,'-b');

xlabel('Time [s]','FontSize',fs);

ylabel('Amplitude [-]','FontSize',fs);

clear temp;

figure(2)

grid on; hold on; box on;

axis square;

% pbaspect([2 1 1])

ax=gca;

ax.FontSize = fs;

xlim([0 15]);

ylim([-0.4 0.4]);

yticks([-0.4 -0.2 0 0.2 0.4]);

temp = table2array(trainingDataF1{1});

plot(time,temp,'-r');

xlabel('Time [s]','FontSize',fs);

ylabel('Amplitude [-]','FontSize',fs);

clear temp;

figure(3)

grid on; hold on; box on;

axis square;

% pbaspect([2 1 1])

ax=gca;

ax.FontSize = fs;

xlim([0 15]);

ylim([-0.4 0.4]);

yticks([-0.4 -0.2 0 0.2 0.4]);

temp = table2array(trainingDataF2{1});

plot(time,temp,'-k');

xlabel('Time [s]','FontSize',fs);

ylabel('Amplitude [-]','FontSize',fs);

clear temp;

%plot frequency

figure(4)

grid on; hold on; box on;

axis square;

% pbaspect([2 1 1])

ax=gca;

ax.FontSize = fs;

xlim([0 60]);

ylim([0 0.005]);

yticks([ 0 0.001 0.002 0.003 0.004 0.005]);

temp = transFFT\_H{1}/L\*2;

plot(FsRange, temp,'-b');

xlabel('Frequency [Hz]','FontSize',fs);

ylabel('Amplitude [-]','FontSize',fs);

clear temp;

figure(5)

grid on; hold on; box on;

axis square;

% pbaspect([2 1 1])

ax=gca;

ax.FontSize = fs;

xlim([0 60]);

ylim([0 0.005]);

yticks([ 0 0.001 0.002 0.003 0.004 0.005]);

temp = transFFT\_F1{1}/L\*2;

plot(FsRange, temp,'-r');

xlabel('Frequency [Hz]','FontSize',fs);

ylabel('Amplitude [-]','FontSize',fs);

clear temp;

% legend('Healthy','Faulty','Location','Northwest','FontSize',fs);

% end

figure(6)

grid on; hold on; box on;

axis square;

% pbaspect([2 1 1])

ax=gca;

ax.FontSize = fs;

xlim([0 60]);

ylim([0 0.005]);

yticks([ 0 0.001 0.002 0.003 0.004 0.005]);

temp = transFFT\_F2{1}/L\*2;

plot(FsRange, temp,'-k');

xlabel('Frequency [Hz]','FontSize',fs);

ylabel('Amplitude [-]','FontSize',fs);

clear temp;

% legend('Healthy','Faulty','Location','Northwest','FontSize',fs);

%plot features

figure(7)

grid on; hold on; box on;

axis square;

% pbaspect([2 1 1])

ax=gca;

ax.FontSize = fs;

yticks([ 0 0.01 0.02 0.03]);

xlim([0 20]);

ylim([0 0.03]);

plot(healthy1X,'-ob');

plot(faulty11X,'-or');

plot(faulty21X,'-ok');

legend('Healthy','Faulty-1','Faulty-2','Location','Northwest','FontSize',fs);

xlabel('# of Samples','FontSize',fs);

ylabel('Amplitude [-]','FontSize',fs);

figure(8)

grid on; hold on; box on;

axis square;

ax=gca;

ax.FontSize = fs;

yticks([ 0 0.01 0.02 0.03]);

% pbaspect([2 1 1])

xlim([0 20]);

ylim([0 0.03]);

plot(healthy2X,'-ob');

plot(faulty12X,'-or');

plot(faulty22X,'-ok');

legend('Healthy','Faulty-1','Faulty-2','Location','Northwest','FontSize',fs);

xlabel('# of Samples','FontSize',fs);

ylabel('Amplitude [-]','FontSize',fs);

figure(9)

grid on; hold on; box on;

axis square;

ax=gca;

ax.FontSize = fs;

yticks([ 0 0.01 0.02 0.03]);

% pbaspect([2 1 1])

xlim([0 20]);

ylim([0 0.03]);

plot(healthy3X,'-ob');

plot(faulty13X,'-or');

plot(faulty23X,'-ok');

legend('Healthy','Faulty-1','Faulty-2','Location','Northwest','FontSize',fs);

xlabel('# of Samples','FontSize',fs);

ylabel('Amplitude [-]','FontSize',fs);

figure(10)

grid on; hold on; box on;

% axis square;

f11 = figure(11)

grid on; hold on; box on;

% axis square;

ax=gca;

ax.FontSize = fs;

yticks([ 0 0.01 0.02 0.03]);

pbaspect([2 1 1])

xlim([0 20]);

ylim([0 0.03]);

plot(stdH,'-ob');

plot(stdF1,'-or');

plot(stdF2,'-ok');

legend('Healthy','Faulty-1','Faulty-2','Location','Northwest','FontSize',fs);

xlabel('# of Samples','FontSize',fs);

ylabel('STD [-]','FontSize',fs);

%PCA results

figure(13)

hBar=bar(explained,'b','EdgeColor','k');

set(gca,'xticklabel',{'1X','2X','3X','4X','STD','RMS','P2P','L1','L2','Linf'})

set(gca,'YScale','log')

set(gca,'FontSize',fs)

ylabel('Variance Explained [%]')

%% 5. SVM

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Before this section, you need prepare

% - FeatMat\_train: Feature matrix for training data

% - FeatMat\_test : Feature matrix for testing data

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

N\_train = 60;

Nh = 20;

Nf1 = 20;

% Training data

Train\_X = FeatMat\_train;

Train\_Y = zeros(N\_train,1);

Train\_Y(1:Nh,1) = 1;

Train\_Y(Nh+1:Nh+Nf1,1) = 2;

Train\_Y(Nh+Nf1+1:N\_train,1) = 3;

% Test Data

Test\_X = FeatMat\_test;

Test\_Y = zeros(30,1);

Test\_Y( 1:10,1) = 1;

Test\_Y(11:20,1) = 2;

Test\_Y(21:30,1) = 3;

% train SVM with different kernel

Mehtod\_list = {'rbf','linear','polynomial','Sigmoid'}; % kernel function selection

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Here you can select kernel function

% Try different kernel and check the results

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Method = Mehtod\_list{1}; % 1: rbf, 2: linear, 3: polynomial, 4: softmargin

switch Method

case 'rbf'

svmStruct = libsvmtrain(Train\_Y,Train\_X,'-s 0 -t 2 -g 0.333 -c 1');

% refer to README file in libsvm for more infomation

case 'linear'

svmStruct = libsvmtrain(Train\_Y,Train\_X,'-s 0 -t 0 -g 0.333 ');

case 'polynomial'

svmStruct = libsvmtrain(Train\_Y,Train\_X,'-s 0 -t 1 -g 0.333 ');

case 'Sigmoid'

svmStruct = libsvmtrain(Train\_Y,Train\_X,'-s 0 -t 3 -g 0.333 ');

end

% Test and predict label

% use trained SVM model for classification

[predicted\_result, accuracy,~] = libsvmpredict(Test\_Y,Test\_X,svmStruct);

%% 6. Confusion Matrix

%%%%%%%%%%%%%%%%%%%

% Write your code %

%%%%%%%%%%%%%%%%%%%

figure(12)

cm = confusionchart(cMatTest,predicted\_result)

cm.FontSize = fs;