

LAB: Bearing Fault Monitoring and RUL Estimation

Industrial AI & Automation 2024

Name: Jin Kwak/21900031

Date: 24.09.24

Sequence

1. Introduction
2. Procedure
3. Feature Reduction/Selection
4. Fault Monitoring and Warning
5. RUL estimation
6. Discussion
7. Conclusion
8. Reference

Introduction

In this LAB, there are two main parts

(1) Fault Monitoring

(2) RUL estimation

Download template matlab code this lab: [Download here](#)

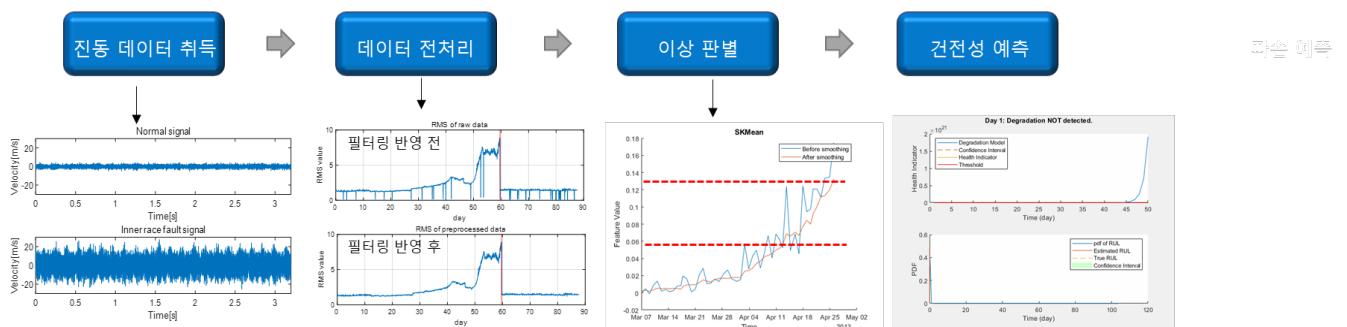
Fault Monitoring

- Detect Fault Status
- Display WARNING MESSAGE (OK, Warning, Damaged)

RUL Estimation

- Estimate RUL similarly to Tutorial codes (CaseStudy 3, 4)
- The End of Life can be assumed when the repair is conducted

An Example Process



Dataset

Raw Dataset of Bearing Velocity measured that shows a Bearing Life cycle before the repair is done. Also the velocity measurement after the repair is provided.

Download

bearing dataset for this lab: [Download here](#)

Using the given dataset, develop a simple program for

- Bearing velocity measured 24 times per day
- Each measurement is L=8192 with Fs=2560Hz
- Measurement for consecutive days until repair is done

Toolbox used in this lab

- Signal Processing Toolbox
- System Identification Toolbox
- Predictive Maintenance Toolbox

The dataset given is a time-series data and bearing velocity data is given.

Throughout this lab, the fault is monitored and most importantly, remaining useful life(RUL) is calculated.

To do this, cell datatype is used in order to extract time/frequency domain features.

Procedure

Data Preparation

Explain how datasets are prepared

```
clc; clear all; close all;
addpath("../Dataset");
addpath("../Library");
read_data = 0;
```

Enumeration

```
RAW_DATA = 1; % data{idx,RAW_DATA}
```

```

FEATURES = 2;      % data{idx,FEATURES}

N_day_data = 8192;

```

```

if ~read_data
    read_data =1;
    load("bearing-vel-fulldata-rul.mat");

    numSamples = length(date);

    time_stamp = 24      ;%24 data a day
    Fs = 2560            ;%[Hz]
    Ts = 1.0/Fs          ;%[s]
    cnt = 1;
    data{1,1} = [];
    for idx = 1: numSamples
        data{cnt,1} = [data{cnt,1}; bearingFulldata(:,idx)];

        if rem(idx,time_stamp) == 0
            cnt = cnt + 1;
            data{cnt,1} = [];
        end
    end
end
num_day = 88;

```

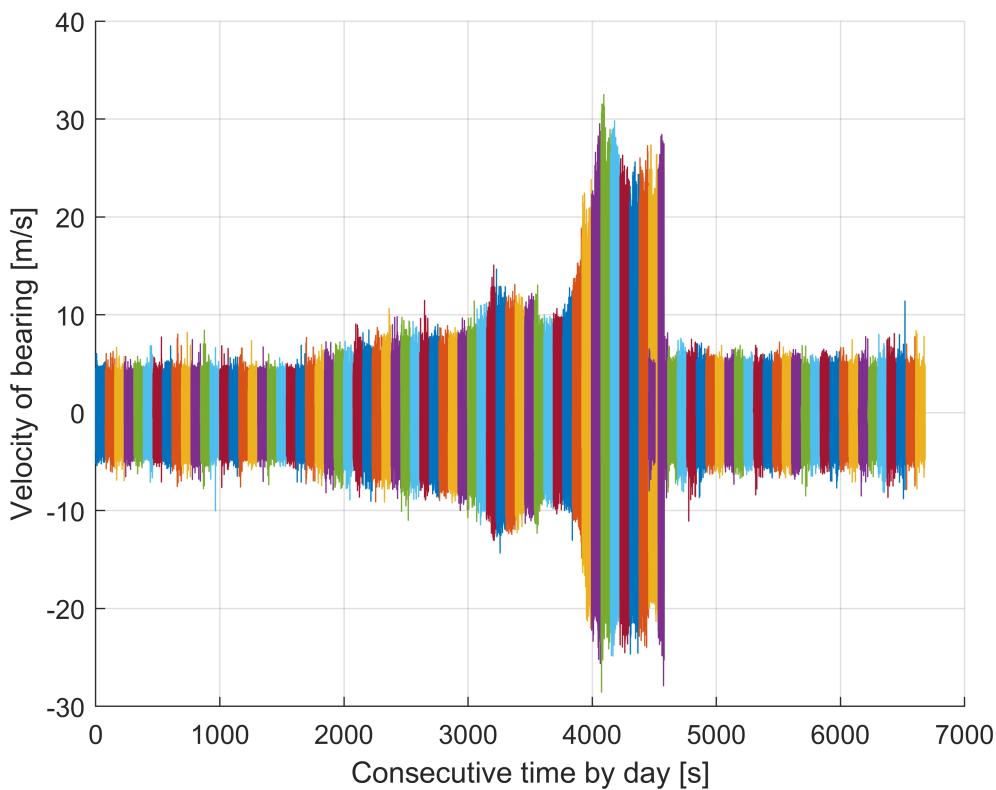
Preprocessing and Feature Extraction

Kurtosis, Standard deviation, peak to peak value extraction(day by day)

```

kurto= [];
mean_data = [];
stdv = [];
ppv  = [];
RMS  = [];
mag_STFT =[];
time_STFT=[];
freq_STFT=[];
figure; hold on;
for idx = 1:num_day
    data{idx,2} = get_features(data{idx,1})      ;
    kurto      = [kurto ;data{idx,2}.kt]         ;
    mean_data  = [mean_data ;data{idx,2}.mean];
    RMS       = [RMS   ;data{idx,2}.rms]        ;
    stdv      = [stdv  ;data{idx,2}.std]         ;
    ppv       = [ppv   ;data{idx,2}.ppv]         ;
    time_idx = (length(data{idx}(:,1))*(idx-1)+1)*Ts:Ts:(idx*length(data{idx}(:,1)))*Ts;
    plot(time_idx, data{idx,1});
end
xlabel('Consecutive time by day [s]'); ylabel('Velocity of bearing [m/s]'); hold on; grid on;

```



In these graphs, **Bearing signal data** can be compared in the same scale.

```

plot1_idx = 1;
plot2_idx = 59;
plot3_idx = 70;

repair_day= 60  ;

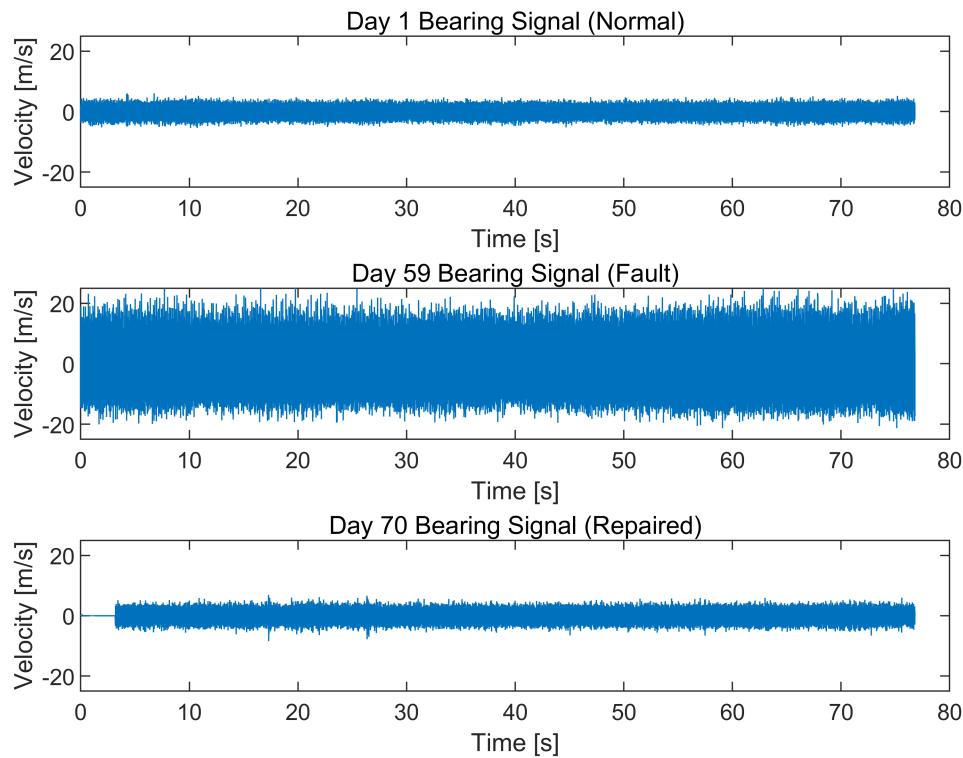
figure;
title('Vibration Data of Bearing');
subplot(3,1,1);
time = 0:Ts:(length(data{plot1_idx}))-1)*Ts;
plot(time, data{plot1_idx});
ylim([-25 25]);
xlabel('Time [s]');
ylabel('Velocity [m/s]');
title('Day 1 Bearing Signal (Normal)');
subplot(3,1,2);
time = 0:Ts:(length(data{plot2_idx}))-1)*Ts;
plot(time, data{plot2_idx});
ylim([-25 25]);
xlabel('Time [s]');
ylabel('Velocity [m/s]');
title('Day 59 Bearing Signal (Fault)');
subplot(3,1,3);
time = 0:Ts:(length(data{plot3_idx}))-1)*Ts;
plot(time, data{plot3_idx});

```

```

ylim([-25 25]);
xlabel('Time [s]');
ylabel('Velocity [m/s]');
title('Day 70 Bearing Signal (Repaired)');

```



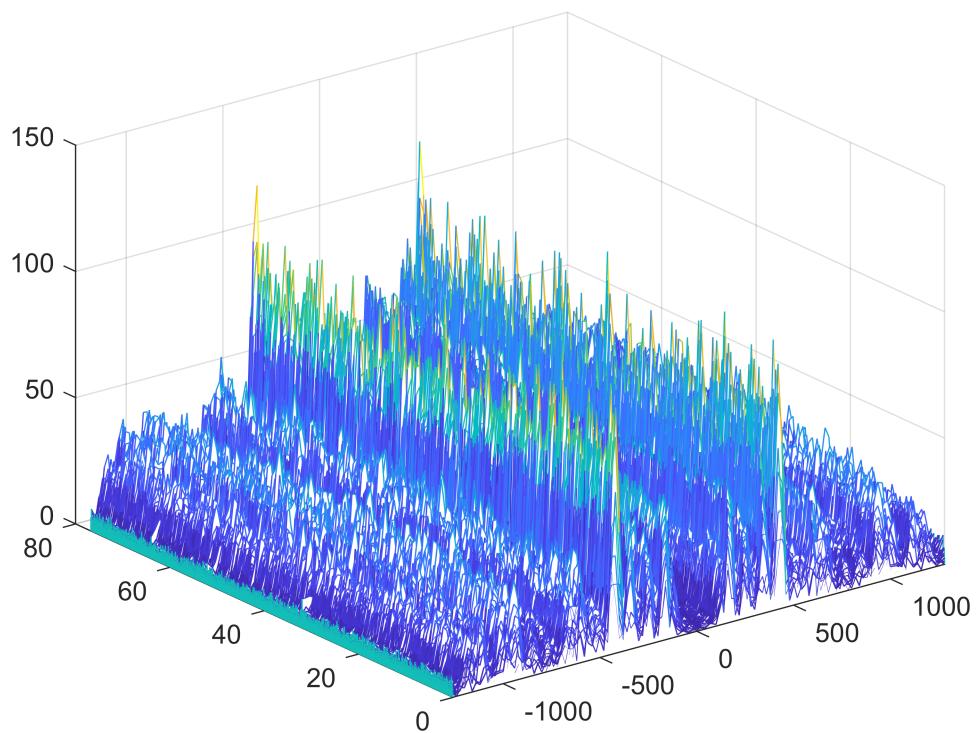
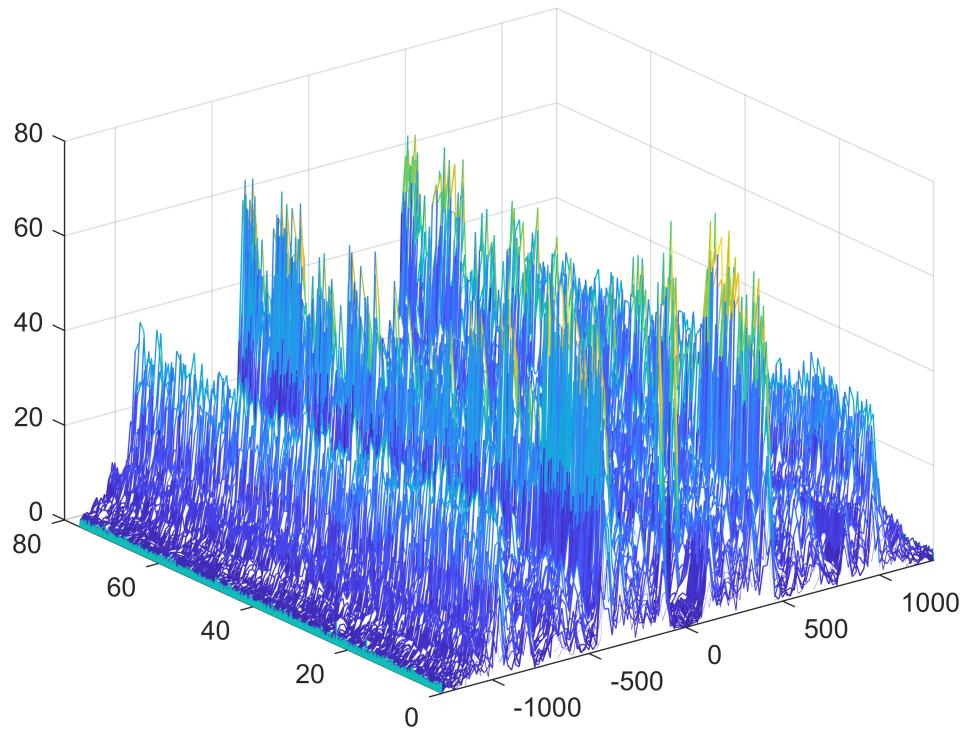
The 3D-graph shows how the spectrum of FFT varies with Normal/Damaged/Repaired data.

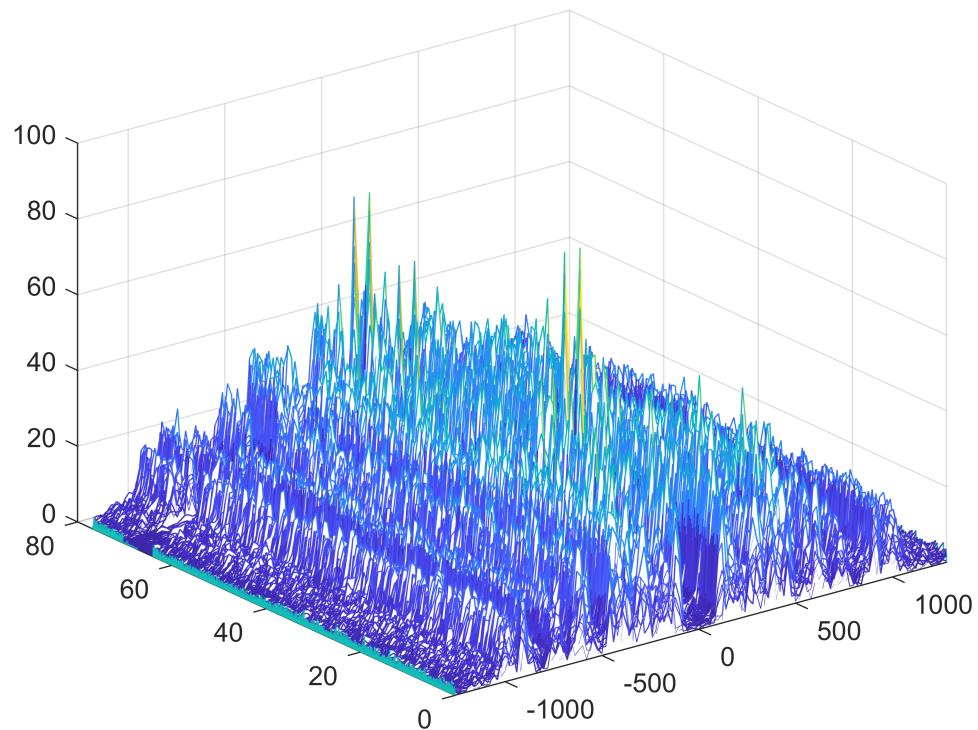
Normal data shows harmonic peaks where damaged data shows chattering of peaks all the time. The repaired dataset shows

```

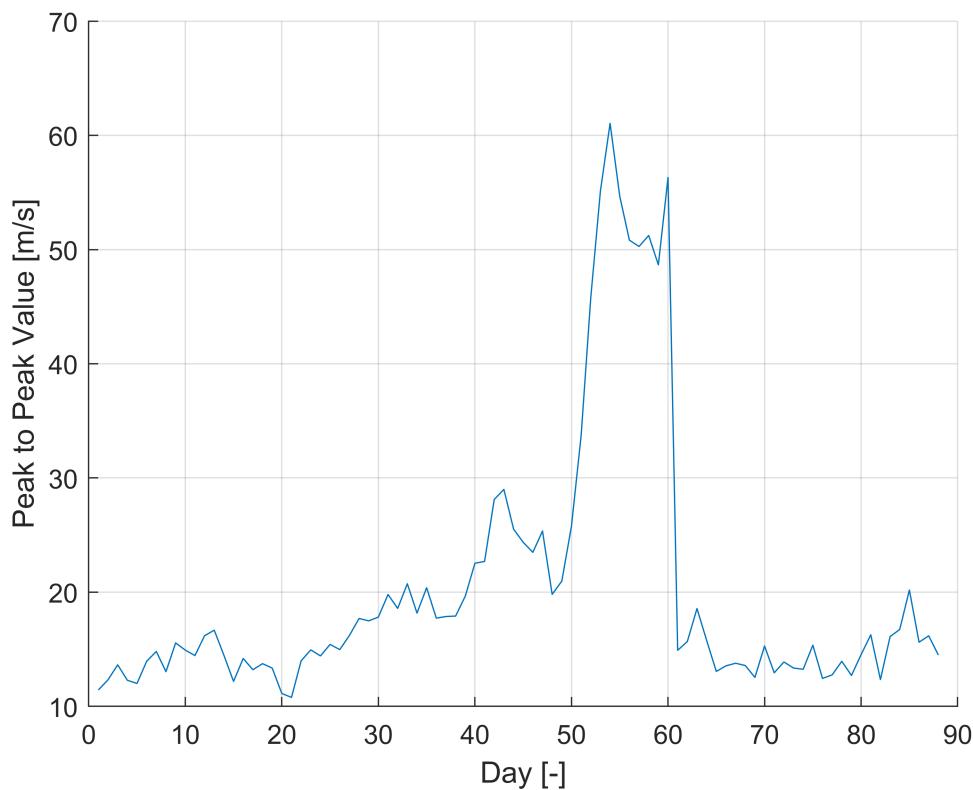
for idx = [1 50 80]
    [S,F,T] = stft(data{idx,1}, 2560);
    figure;
    title('day: ',num2str(idx));
    waterfall(F,T,abs(S))
end

```

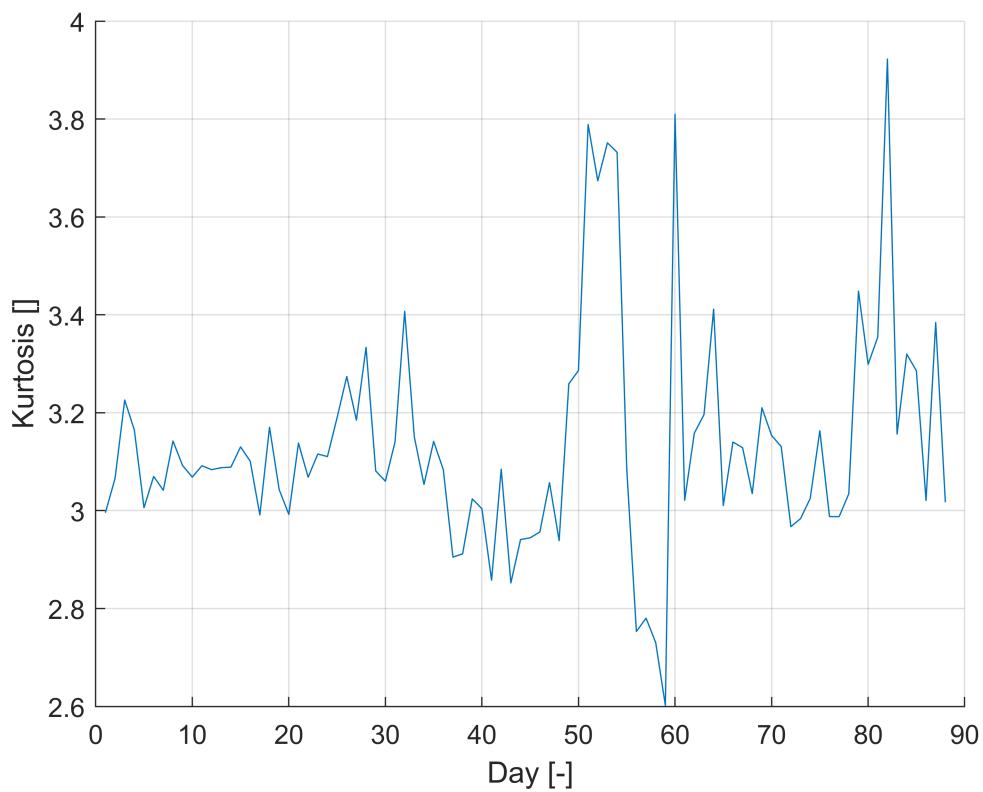




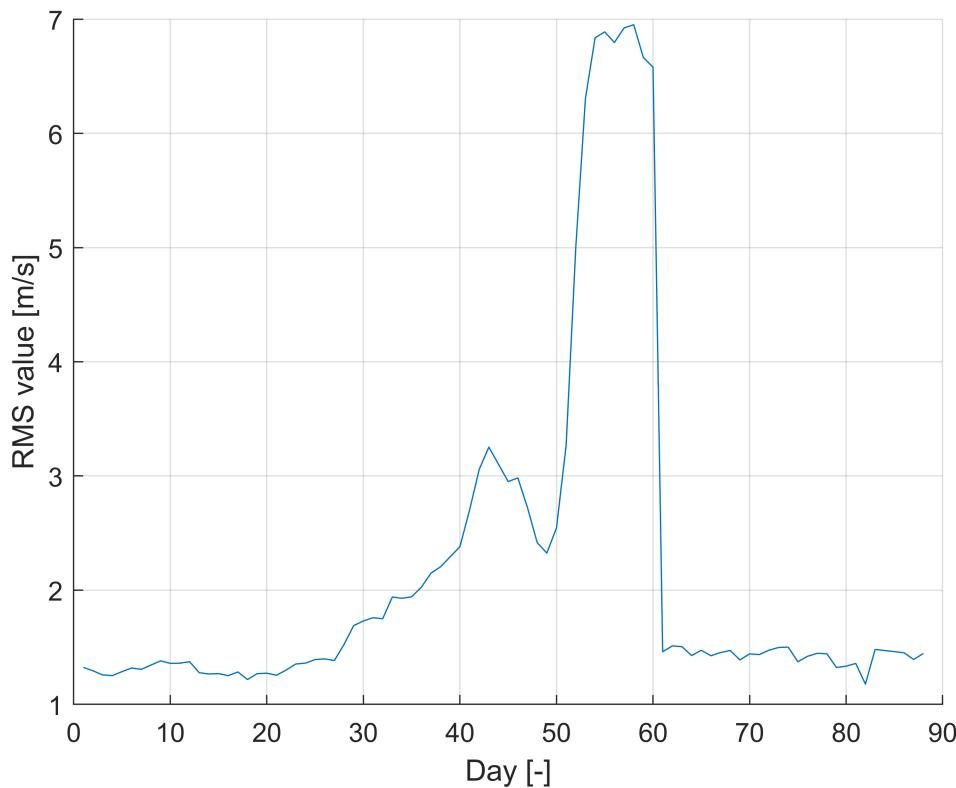
```
day = 1:num_day;
figure; hold on; grid on;
plot(day,pvv);
xlabel('Day [-]');
ylabel('Peak to Peak Value [m/s]');
```



```
figure; hold on; grid on;
plot(day,kurto);
xlabel('Day [-]');
ylabel('Kurtosis []');
```



```
figure; hold on; grid on;
plot(day,RMS);
xlabel('Day [-]');
ylabel('RMS value [m/s]');
```



```
disp('Average Kurtosis value of normal bearing dataset');
```

Average Kurtosis value of normal bearing dataset

```
average_kurtosis_value = mean(kurto(1:25))
```

average_kurtosis_value = 3.0909

```
rms_kurtosis_value = rms(kurto(1:25))
```

rms_kurtosis_value = 3.0915

```
disp('Average Peak to Peak value of normal bearing dataset');
```

Average Peak to Peak value of normal bearing dataset

```
average_ppv_value = mean(ppv(1:25))
```

average_ppv_value = 13.7281

```
rms_ppv_value = rms(ppv(1:25))
```

rms_ppv_value = 13.8121

% Mean value of rms in normal data and abnormal data

```
average_rms_value_normal = mean(RMS(1:25))
```

average_rms_value_normal = 1.3054

```
average_rms_value_fault = mean(RMS(35:59))
```

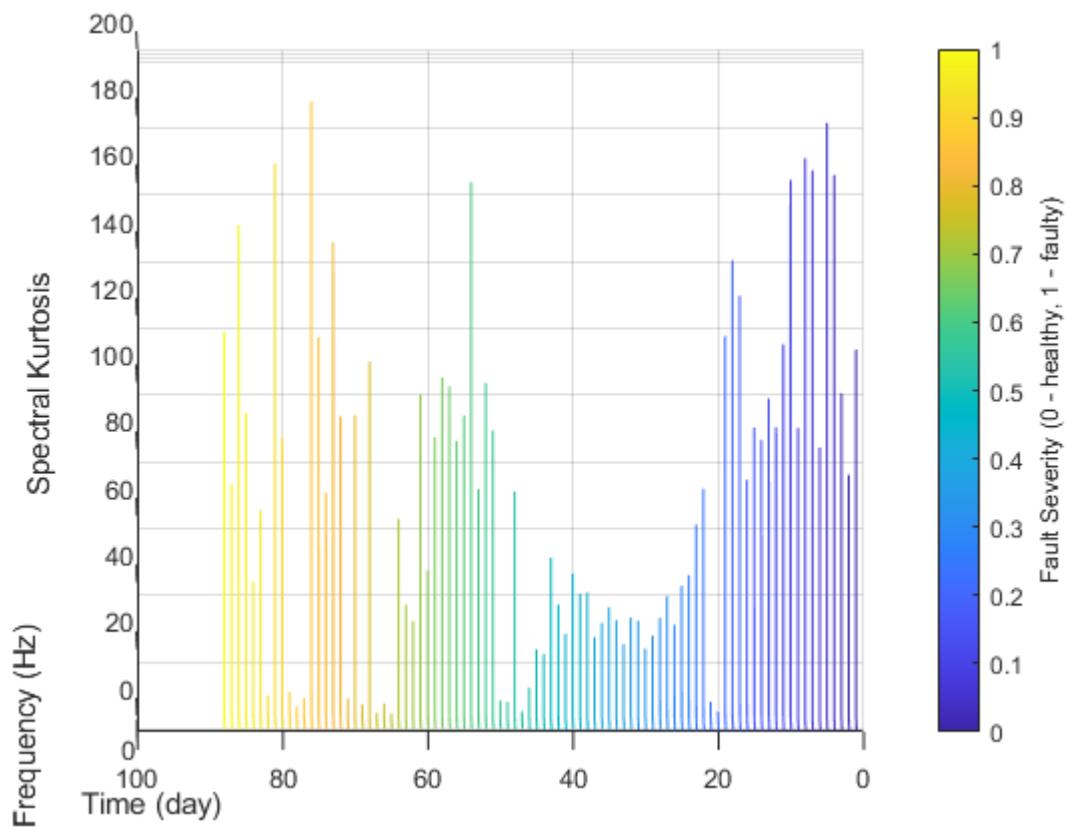
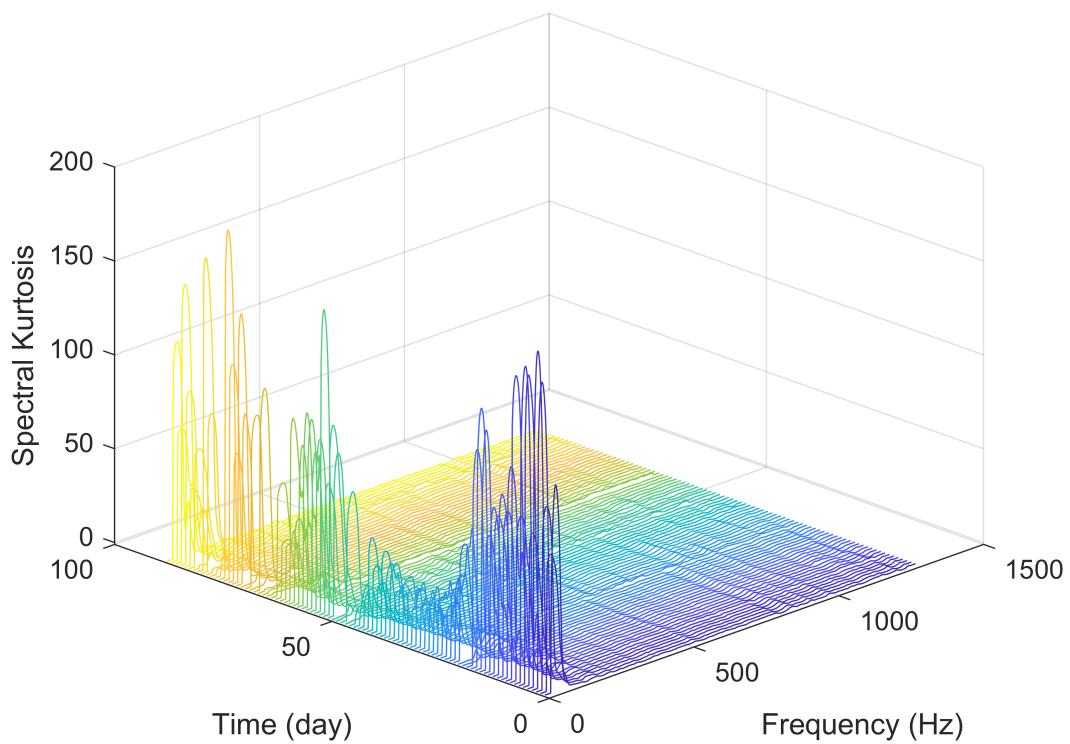
```
average_rms_value_fault = 3.8673
```

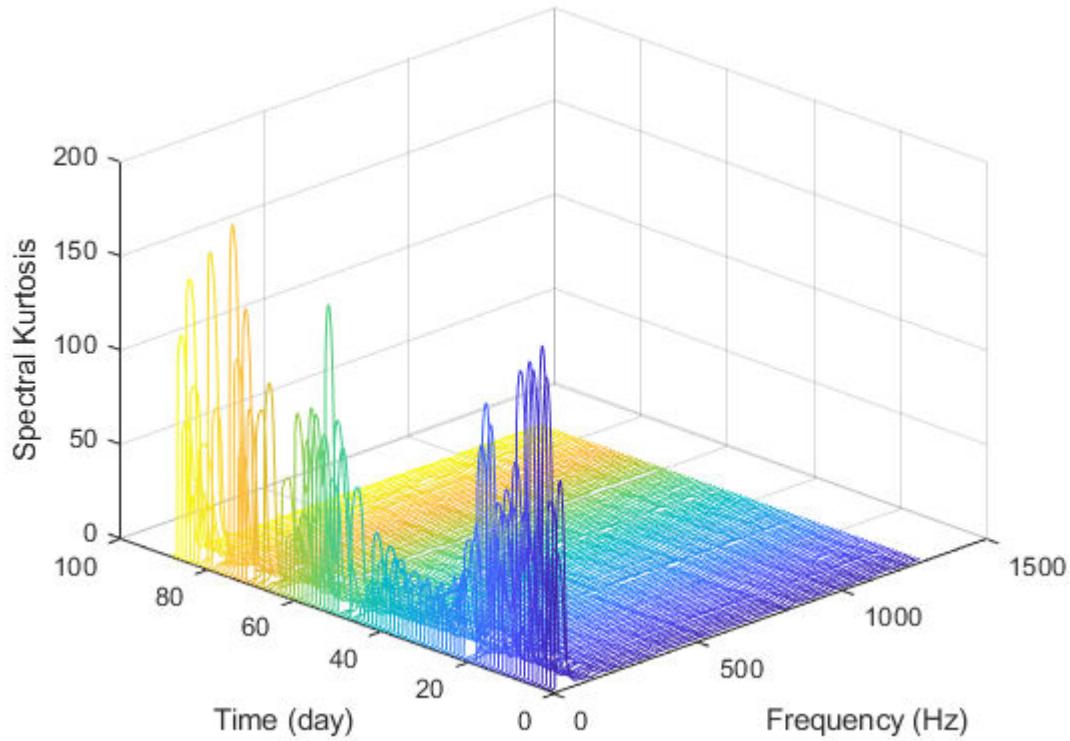
- The root-mean squared of Kurtosis, mean value of kurtosis are roughly 0.006 different in other words, in which means that kurtosis of normal data are very likely to be around 3.1.
- Mean values of Root-mean-squared value of normal data and abnormal data differs roughly about 3 times.
- Through this section, Peak to peak value, kurtosis, rms value are observed.
- The most distinguishable data is rms value
- distinguishable data

Spectral Kurtosis

```
SpectralKurtosis = [];
normal = [];
bad = [];
% Compute spectral kurtosis with window size = 128
wc = 128;
colors = parula(num_day);

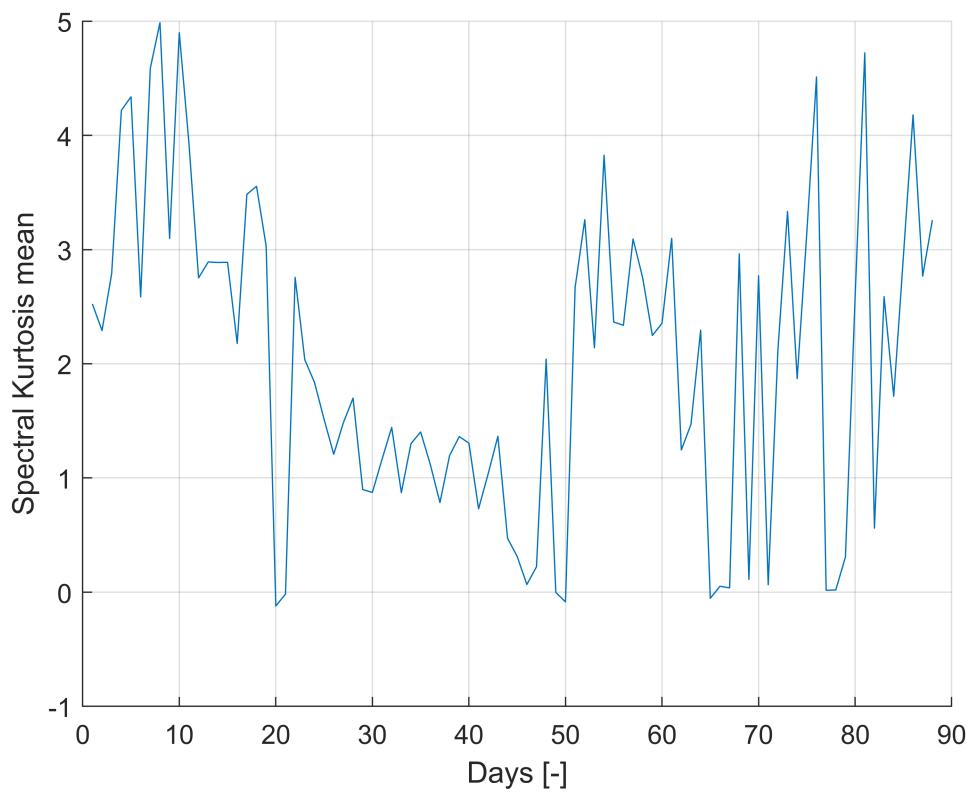
figure; hold on;
for idx = 1:num_day
    % Get vibration signal and measurement date
    vib_data = data{idx,RAW_DATA};
    [SK, F] = pkurtosis(vib_data, Fs, wc);
    data{idx,FEATURES}.SKmean = mean(SK);
    data{idx,FEATURES}.SKstd = std(SK);
    data{idx,FEATURES}.SKskew = skewness(SK);
    data{idx,FEATURES}.SKkurt = kurtosis(SK);
    SpectralKurtosis = [SpectralKurtosis;data{idx,FEATURES}.SKmean data{idx,FEATURES}.SKstd data{idx,FEATURES}.SKskew data{idx,FEATURES}.SKkurt];
    plot3(F, day(idx)*ones(size(F)), SK, 'Color', colors(idx, :));
    if idx == 88
        break;
    elseif idx< 25 || (idx>60 && idx <88)
        normal = [normal data{idx,RAW_DATA}];
    else
        bad = [bad data{idx,RAW_DATA}];
    end
end
normal=[normal [data{88,RAW_DATA}; zeros(length(data{87,RAW_DATA}))-length(data{88,RAW_DATA}),1];
hold off
xlabel('Frequency (Hz)')
ylabel('Time (day)')
zlabel('Spectral Kurtosis')
grid on; view(-45, 30);
```



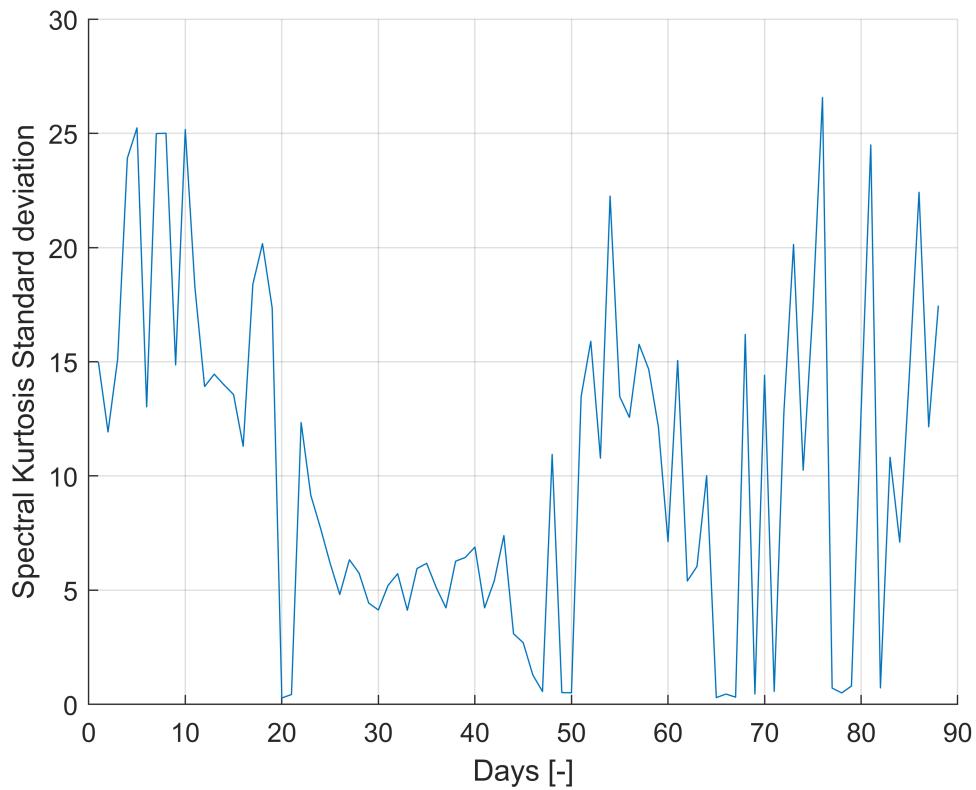


Plot SKmean

```
%Plot
SK_MEAN = 1;
SK_STD  = 2;
SK_SKEW  = 3;
SK_KURT  = 4;
figure; hold on; grid on;
plot(day, SpectralKurtosis(:,SK_MEAN));
xlabel('Days [-]'); ylabel("Spectral Kurtosis mean");
```



```
figure; hold on; grid on;
plot(day, SpectralKurtosis(:,SK_STD));
xlabel('Days [-]'); ylabel("Spectral Kurtosis Standard deviation");
```



Spectral Kurtosis mean/ skewness does not show very distinct variation with normal/fault data

The features are tabulated.

Rows : Days

Columns: Features

```
data_in_table = cell2table(data);
```

The columns are updated until the repair day (Which is assumed 60th day).

```
data_in_table = data_in_table(1:repair_day,:)
```

```
data_in_table = 60x16 table
```

| | fft_max_idx | IF | SF | CF | MF | RMS | SRA | AAV |
|---|-------------|--------|--------|--------|--------|--------|--------|--------|
| 1 | 31596 | 5.7178 | 1.2536 | 4.5612 | 6.7514 | 1.3242 | 0.8946 | 1.0563 |
| 2 | 32055 | 6.0796 | 1.2569 | 4.8372 | 7.1870 | 1.2941 | 0.8710 | 1.0297 |
| 3 | 32049 | 6.8109 | 1.2804 | 5.3191 | 8.2396 | 1.2577 | 0.8119 | 0.9823 |
| 4 | 164554 | 5.8755 | 1.2796 | 4.5916 | 7.1143 | 1.2523 | 0.8082 | 0.9786 |
| 5 | 164573 | 5.6290 | 1.2555 | 4.4836 | 6.6545 | 1.2869 | 0.8671 | 1.0250 |
| 6 | 164562 | 6.5162 | 1.2565 | 5.1860 | 7.7027 | 1.3189 | 0.8880 | 1.0497 |

| | fft_max_idx | IF | SF | CF | MF | RMS | SRA | AAV |
|----|-------------|--------|--------|--------|--------|--------|--------|--------|
| 7 | 164577 | 7.4314 | 1.2564 | 5.9146 | 8.7888 | 1.3069 | 0.8795 | 1.0402 |
| 8 | 163966 | 6.4950 | 1.2786 | 5.0797 | 7.8661 | 1.3446 | 0.8683 | 1.0516 |
| 9 | 164400 | 7.2977 | 1.2568 | 5.8064 | 8.6280 | 1.3812 | 0.9295 | 1.0990 |
| 10 | 32203 | 7.5895 | 1.2567 | 6.0390 | 8.9776 | 1.3595 | 0.9145 | 1.0818 |
| 11 | 32210 | 6.9016 | 1.2572 | 5.4895 | 8.1621 | 1.3608 | 0.9152 | 1.0824 |
| 12 | 164406 | 7.7107 | 1.2568 | 6.1351 | 9.1173 | 1.3741 | 0.9246 | 1.0933 |
| 13 | 31893 | 6.5372 | 1.2561 | 5.2045 | 7.7255 | 1.2777 | 0.8608 | 1.0173 |
| 14 | 170628 | 6.7843 | 1.2559 | 5.4018 | 8.0165 | 1.2663 | 0.8532 | 1.0082 |
| 15 | 31887 | 5.7297 | 1.2762 | 4.4897 | 6.9172 | 1.2696 | 0.8240 | 0.9948 |
| 16 | 31891 | 6.6456 | 1.2587 | 5.2799 | 7.8674 | 1.2519 | 0.8402 | 0.9946 |
| 17 | 31885 | 7.1840 | 1.2532 | 5.7326 | 8.4825 | 1.2839 | 0.8677 | 1.0245 |
| 18 | 31892 | 7.0540 | 1.2792 | 5.5144 | 8.5543 | 1.2181 | 0.7852 | 0.9522 |
| 19 | 164718 | 5.9403 | 1.2568 | 4.7267 | 7.0293 | 1.2694 | 0.8536 | 1.0101 |
| 20 | 164732 | 5.5342 | 1.2536 | 4.4145 | 6.5353 | 1.2731 | 0.8599 | 1.0155 |
| 21 | 32054 | 5.1462 | 1.2794 | 4.0224 | 6.2382 | 1.2555 | 0.8095 | 0.9813 |
| 22 | 73713 | 6.6706 | 1.2575 | 5.3046 | 7.8927 | 1.3008 | 0.8742 | 1.0344 |
| 23 | 32059 | 7.1398 | 1.2584 | 5.6736 | 8.4473 | 1.3536 | 0.9092 | 1.0757 |
| 24 | 32053 | 6.4489 | 1.2577 | 5.1276 | 7.6259 | 1.3612 | 0.9153 | 1.0824 |
| 25 | 164555 | 6.5626 | 1.2608 | 5.2050 | 7.7695 | 1.3929 | 0.9331 | 1.1047 |
| 26 | 122855 | 6.5257 | 1.2641 | 5.1624 | 7.7339 | 1.3986 | 0.9336 | 1.1064 |
| 27 | 164551 | 6.6413 | 1.2610 | 5.2667 | 7.8650 | 1.3842 | 0.9269 | 1.0977 |
| 28 | 47995 | 7.6375 | 1.2849 | 5.9441 | 9.2558 | 1.5259 | 0.9799 | 1.1876 |
| 29 | 34190 | 6.2808 | 1.2565 | 4.9985 | 7.4216 | 1.6905 | 1.1386 | 1.3454 |
| 30 | 48009 | 6.3249 | 1.2566 | 5.0333 | 7.4771 | 1.7305 | 1.1649 | 1.3771 |
| 31 | 162879 | 7.6378 | 1.2592 | 6.0655 | 9.0372 | 1.7591 | 1.1807 | 1.3970 |
| 32 | 33719 | 7.3406 | 1.3081 | 5.6116 | 9.1162 | 1.7499 | 1.0772 | 1.3378 |
| 33 | 162414 | 6.3251 | 1.2587 | 5.0252 | 7.4777 | 1.9402 | 1.3039 | 1.5415 |
| 34 | 33713 | 6.0667 | 1.2557 | 4.8314 | 7.1683 | 1.9290 | 1.3002 | 1.5363 |
| 35 | 162883 | 7.5346 | 1.2756 | 5.9066 | 9.0905 | 1.9419 | 1.2618 | 1.5223 |
| 36 | 47776 | 5.6732 | 1.2730 | 4.4564 | 6.8365 | 2.0263 | 1.3209 | 1.5917 |
| 37 | 148833 | 5.1649 | 1.2469 | 4.1422 | 6.0772 | 2.1486 | 1.4645 | 1.7232 |
| 38 | 47743 | 4.9279 | 1.2490 | 3.9455 | 5.8077 | 2.2051 | 1.4980 | 1.7655 |
| 39 | 28160 | 5.3552 | 1.2714 | 4.2119 | 6.4554 | 2.2935 | 1.4964 | 1.8038 |

| | fft_max_idx | IF | SF | CF | MF | RMS | SRA | AAV |
|----|-------------|--------|--------|--------|--------|--------|--------|--------|
| 40 | 168469 | 6.0897 | 1.2694 | 4.7972 | 7.3309 | 2.3806 | 1.5578 | 1.8753 |
| 41 | 28143 | 5.1773 | 1.2445 | 4.1600 | 6.0896 | 2.7019 | 1.8458 | 2.1710 |
| 42 | 183380 | 6.2786 | 1.2732 | 4.9313 | 7.5715 | 3.0580 | 1.9917 | 2.4018 |
| 43 | 28131 | 5.6094 | 1.2448 | 4.5064 | 6.5981 | 3.2532 | 2.2219 | 2.6135 |
| 44 | 13212 | 5.2848 | 1.2510 | 4.2243 | 6.2339 | 3.1011 | 2.1014 | 2.4788 |
| 45 | 168547 | 5.1407 | 1.2517 | 4.1070 | 6.0658 | 2.9510 | 1.9981 | 2.3576 |
| 46 | 183429 | 5.1149 | 1.2517 | 4.0864 | 6.0314 | 2.9831 | 2.0211 | 2.3833 |
| 47 | 183433 | 6.0050 | 1.2539 | 4.7893 | 7.0862 | 2.7227 | 1.8402 | 2.1715 |
| 48 | 32076 | 5.1583 | 1.2493 | 4.1289 | 6.0794 | 2.4171 | 1.6416 | 1.9347 |
| 49 | 26584 | 5.8773 | 1.2842 | 4.5768 | 7.1246 | 2.3248 | 1.4934 | 1.8103 |
| 50 | 184109 | 6.3524 | 1.2667 | 5.0150 | 7.5366 | 2.5444 | 1.6931 | 2.0087 |
| 51 | 177610 | 7.4352 | 1.2892 | 5.7673 | 8.8960 | 3.2667 | 2.1178 | 2.5339 |
| 52 | 184059 | 6.1710 | 1.2954 | 4.7638 | 7.4123 | 5.0002 | 3.2136 | 3.8600 |
| 53 | 12542 | 6.1767 | 1.3193 | 4.6817 | 7.6111 | 6.3054 | 3.8785 | 4.7792 |
| 54 | 177562 | 6.2771 | 1.3203 | 4.7542 | 7.7448 | 6.8361 | 4.1964 | 5.1775 |
| 55 | 19059 | 5.4604 | 1.2614 | 4.3290 | 6.4764 | 6.8885 | 4.6044 | 5.4612 |
| 56 | 19046 | 4.7857 | 1.2364 | 3.8705 | 5.6069 | 6.7949 | 4.6906 | 5.4955 |
| 57 | 12518 | 4.5966 | 1.2420 | 3.7010 | 5.4007 | 6.9224 | 4.7438 | 5.5736 |
| 58 | 184114 | 4.8601 | 1.2388 | 3.9231 | 5.7066 | 6.9511 | 4.7787 | 5.6110 |
| 59 | 12508 | 5.0119 | 1.2220 | 4.1013 | 5.8239 | 6.6637 | 4.6927 | 5.4531 |
| 60 | 12515 | 6.1010 | 1.4124 | 4.3196 | 8.1634 | 6.5792 | 3.4814 | 4.6582 |

```
window_size = 5;
data_in_table_MAF5 = varfun(@(x) movmean(x, [window_size 0]), data_in_table); %smoothing
data_in_table_MAF5.Properties.VariableNames = data_in_table.Properties.VariableNames;
```

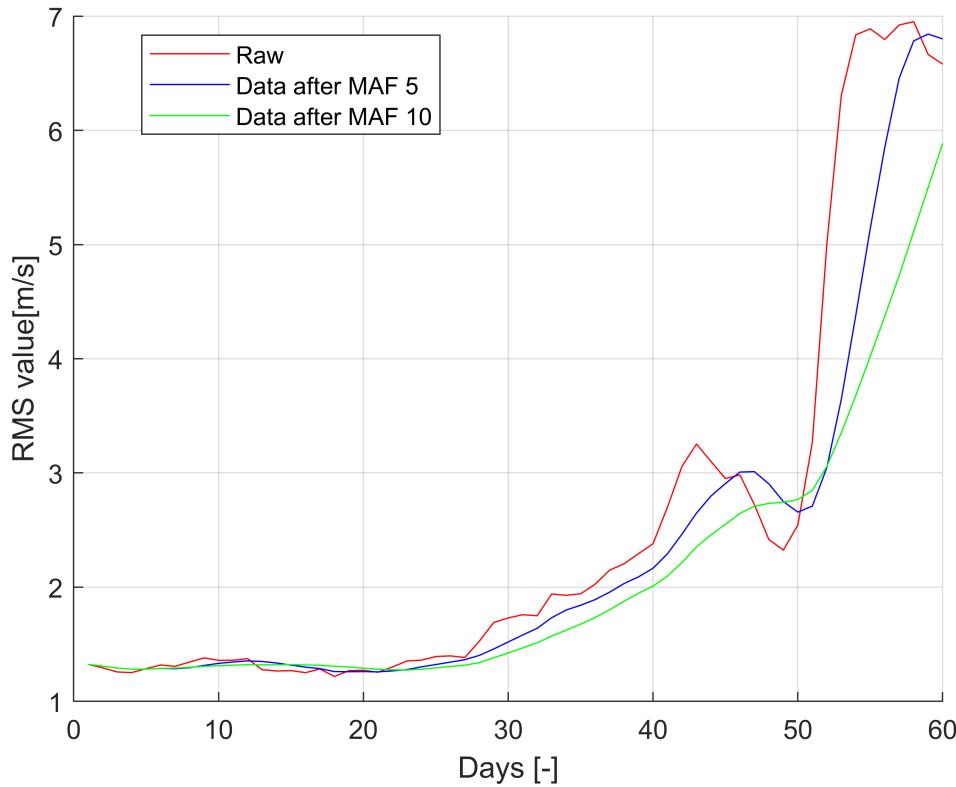
Smoothing the dataset may give soft curve shape and may be more fit to the degradation curve.

Yet the moving average filter acts as a phase delay and may affect the trendability of the dataset. Therefore, the window size 5 is selected. Maximum time(phase) delay of the system is 5 days because warning status may be up to 20~30 days until the next status(Fault).

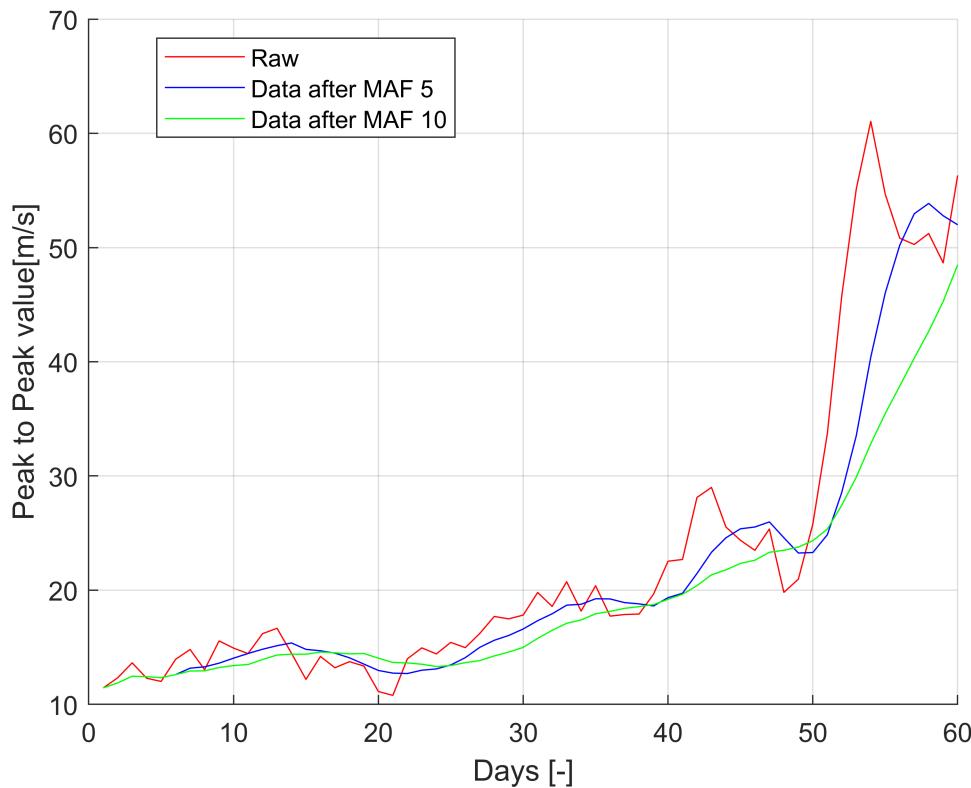
```
data_in_table_MAF10 = varfun(@(x) movmean(x, [10 0]), data_in_table); %smoothing
data_in_table_MAF10.Properties.VariableNames = data_in_table.Properties.VariableNames;

figure; hold on; grid on;
xlabel('Days [-]'); ylabel('RMS value[m/s]');
plot(data_in_table.RMS, 'r-');
plot(data_in_table_MAF5.RMS, 'b-');
plot(data_in_table_MAF10.RMS, 'g-');
```

```
legend('Raw','Data after MAF 5','Data after MAF 10'); legend('Location','best');
```



```
figure; hold on; grid on;
xlabel('Days [-]'); ylabel('Peak to Peak value[m/s]');
plot(data_in_table.PEAK2PEAK,'r-');
plot(data_in_table_MAF5.PEAK2PEAK,'b-');
plot(data_in_table_MAF10.PEAK2PEAK,'g-');
legend('Raw','Data after MAF 5','Data after MAF 10'); legend('Location','best');
```



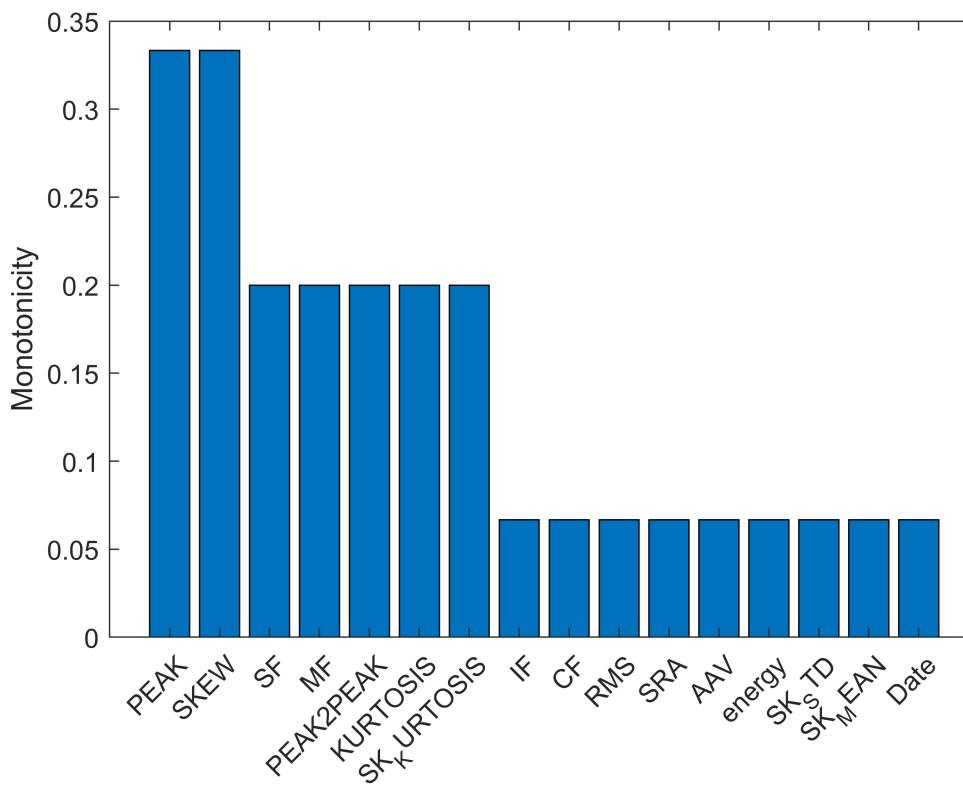
Feature Reduction/Selection

Energy is the most relevant monotonicity feature.

Energy → AAV(Average Absolute Value) → SRA(Square Root Average) → Peak and Peak to peak features

This shows that the phenomena caused by fault of a bearing may affect the amplitude the most as the important features are all related to the amplitude(e.g. $Energy = \sum x^2(t)$).

```
break_point = 45;
feature_monotonicity = monotonicity(data_in_table_MAF5(break_point:end,:), 'WindowSize', 0);
names = string(feature_monotonicity.Properties.VariableNames);
important_feature = table2array(feature_monotonicity);
helperSortedBarPlot(feature_monotonicity, 'Monotonicity');
```



```
confidence_level = 0.15;
train_selected = data_in_table_MAF5(:, feature_monotonicity{:, :}) >= confidence_level)
```

train_selected = 60x7 table

| | IF | CF | energy | PEAK | PEAK2PEAK | SKEW | SK_MEAN |
|----|--------|--------|------------|--------|-----------|---------|---------|
| 1 | 5.7178 | 4.5612 | 3.4477e+05 | 6.0400 | 11.4400 | -0.0519 | 2.5236 |
| 2 | 5.8987 | 4.6992 | 3.3702e+05 | 6.1500 | 11.8850 | -0.0518 | 2.4072 |
| 3 | 6.2028 | 4.9058 | 3.2835e+05 | 6.3300 | 12.4700 | -0.0559 | 2.5350 |
| 4 | 6.1210 | 4.8273 | 3.2334e+05 | 6.1850 | 12.4250 | -0.0571 | 2.9558 |
| 5 | 6.0226 | 4.7585 | 3.2380e+05 | 6.1020 | 12.3420 | -0.0562 | 3.2321 |
| 6 | 6.1049 | 4.8298 | 3.2683e+05 | 6.2250 | 12.6100 | -0.0562 | 3.1242 |
| 7 | 6.3904 | 5.0554 | 3.2534e+05 | 6.5067 | 13.1717 | -0.0569 | 3.4683 |
| 8 | 6.4597 | 5.0958 | 3.2970e+05 | 6.6017 | 13.2900 | -0.0563 | 3.9176 |
| 9 | 6.5408 | 5.1770 | 3.4038e+05 | 6.8233 | 13.6100 | -0.0533 | 3.9685 |
| 10 | 6.8265 | 5.4182 | 3.4956e+05 | 7.2333 | 14.0500 | -0.0493 | 4.0818 |
| 11 | 7.0386 | 5.5859 | 3.5597e+05 | 7.5167 | 14.4567 | -0.0493 | 4.0143 |
| 12 | 7.2376 | 5.7441 | 3.6083e+05 | 7.7817 | 14.8300 | -0.0471 | 4.0422 |
| 13 | 7.0886 | 5.6257 | 3.5836e+05 | 7.6017 | 15.1400 | -0.0463 | 3.7595 |
| 14 | 7.1368 | 5.6794 | 3.5166e+05 | 7.6033 | 15.3800 | -0.0462 | 3.4095 |

| | IF | CF | energy | PEAK | PEAK2PEAK | SKEW | SK_MEAN |
|----|--------|--------|------------|---------|-----------|---------|---------|
| 15 | 6.8755 | 5.4599 | 3.4196e+05 | 7.2167 | 14.8183 | -0.0465 | 3.3749 |
| 16 | 6.7182 | 5.3334 | 3.3275e+05 | 6.9500 | 14.6967 | -0.0489 | 2.9213 |
| 17 | 6.7652 | 5.3739 | 3.2609e+05 | 6.9317 | 14.4900 | -0.0483 | 2.8465 |
| 18 | 6.6558 | 5.2705 | 3.1284e+05 | 6.6462 | 14.0821 | -0.0515 | 2.9801 |
| 19 | 6.5563 | 5.1908 | 3.1215e+05 | 6.5378 | 13.5304 | -0.0538 | 3.0032 |
| 20 | 6.3480 | 5.0263 | 3.1271e+05 | 6.3345 | 12.9721 | -0.0555 | 2.5019 |
| 21 | 6.2507 | 4.9484 | 3.1155e+05 | 6.2262 | 12.7387 | -0.0580 | 2.0177 |
| 22 | 6.2549 | 4.9525 | 3.1563e+05 | 6.2745 | 12.7037 | -0.0608 | 2.1143 |
| 23 | 6.2475 | 4.9427 | 3.2166e+05 | 6.3278 | 12.9937 | -0.0636 | 1.8729 |
| 24 | 6.1467 | 4.8782 | 3.3376e+05 | 6.3717 | 13.1067 | -0.0618 | 1.5869 |
| 25 | 6.2504 | 4.9579 | 3.4454e+05 | 6.5800 | 13.4517 | -0.0616 | 1.3341 |
| 26 | 6.4156 | 5.0826 | 3.5552e+05 | 6.8467 | 14.0917 | -0.0617 | 1.5554 |
| 27 | 6.6648 | 5.2900 | 3.6665e+05 | 7.2200 | 14.9950 | -0.0609 | 1.8058 |
| 28 | 6.8260 | 5.3966 | 3.8751e+05 | 7.5817 | 15.6133 | -0.0498 | 1.6296 |
| 29 | 6.6828 | 5.2841 | 4.2111e+05 | 7.7100 | 16.0367 | -0.0340 | 1.4402 |
| 30 | 6.6621 | 5.2683 | 4.5851e+05 | 7.9983 | 16.6033 | -0.0205 | 1.2795 |
| 31 | 6.8413 | 5.4118 | 4.9634e+05 | 8.5683 | 17.3317 | -0.0038 | 1.2212 |
| 32 | 6.9771 | 5.4866 | 5.3259e+05 | 9.0017 | 17.9333 | 0.0110 | 1.2604 |
| 33 | 6.9244 | 5.4464 | 5.9316e+05 | 9.4117 | 18.6883 | 0.0221 | 1.1582 |
| 34 | 6.6626 | 5.2609 | 6.3881e+05 | 9.4533 | 18.7667 | 0.0227 | 1.0918 |
| 35 | 6.8716 | 5.4123 | 6.6873e+05 | 9.9567 | 19.2500 | 0.0188 | 1.1758 |
| 36 | 6.7630 | 5.3161 | 7.0515e+05 | 10.0100 | 19.2350 | 0.0113 | 1.2166 |
| 37 | 6.3508 | 4.9956 | 7.5502e+05 | 9.7150 | 18.9133 | 0.0010 | 1.1534 |
| 38 | 5.9487 | 4.7179 | 8.1400e+05 | 9.5283 | 18.8017 | -0.0062 | 1.1126 |
| 39 | 5.7871 | 4.5823 | 8.6301e+05 | 9.5133 | 18.6200 | -0.0101 | 1.1943 |
| 40 | 5.7909 | 4.5766 | 9.2678e+05 | 9.8633 | 19.3475 | -0.0130 | 1.1951 |
| 41 | 5.3980 | 4.2855 | 1.0424e+06 | 9.8250 | 19.7308 | -0.0155 | 1.0829 |
| 42 | 5.4989 | 4.3647 | 1.2143e+06 | 10.8333 | 21.4625 | -0.0178 | 1.0697 |
| 43 | 5.5730 | 4.4254 | 1.4098e+06 | 11.7933 | 23.3175 | -0.0207 | 1.1665 |
| 44 | 5.6325 | 4.4718 | 1.5656e+06 | 12.5267 | 24.5875 | -0.0274 | 1.0456 |
| 45 | 5.5968 | 4.4544 | 1.6786e+06 | 12.9367 | 25.3742 | -0.0131 | 0.8708 |
| 46 | 5.4343 | 4.3359 | 1.7845e+06 | 13.0650 | 25.5333 | 0.0073 | 0.6644 |
| 47 | 5.5722 | 4.4408 | 1.7882e+06 | 13.3650 | 25.9783 | 0.0086 | 0.5799 |

| | IF | CF | energy | PEAK | PEAK2PEAK | SKEW | SK_MEAN |
|----|--------|--------|------------|---------|-----------|--------|---------|
| 48 | 5.3855 | 4.3070 | 1.6732e+06 | 12.5150 | 24.5933 | 0.0141 | 0.7469 |
| 49 | 5.4302 | 4.3188 | 1.5036e+06 | 11.8450 | 23.2550 | 0.0248 | 0.5189 |
| 50 | 5.6081 | 4.4506 | 1.4006e+06 | 11.7883 | 23.2983 | 0.0446 | 0.4261 |
| 51 | 5.9905 | 4.7273 | 1.4649e+06 | 12.9083 | 24.8533 | 0.0583 | 0.8189 |
| 52 | 6.1666 | 4.8402 | 1.9925e+06 | 14.8467 | 28.5717 | 0.0705 | 1.3513 |
| 53 | 6.1952 | 4.8222 | 3.0524e+06 | 17.5933 | 33.5350 | 0.1171 | 1.6707 |
| 54 | 6.3816 | 4.9265 | 4.3923e+06 | 21.3467 | 40.4083 | 0.1503 | 1.9682 |
| 55 | 6.3122 | 4.8852 | 5.7701e+06 | 24.5433 | 46.0250 | 0.1561 | 2.3629 |
| 56 | 6.0510 | 4.6944 | 7.0709e+06 | 26.8000 | 50.1967 | 0.1539 | 2.7665 |
| 57 | 5.5779 | 4.3500 | 8.2915e+06 | 27.9300 | 52.9583 | 0.1303 | 2.8370 |
| 58 | 5.3595 | 4.2099 | 9.0554e+06 | 28.5050 | 53.8633 | 0.1167 | 2.7520 |
| 59 | 5.1653 | 4.1132 | 9.2077e+06 | 28.1400 | 52.7833 | 0.0862 | 2.7700 |
| 60 | 5.1360 | 4.0408 | 9.0948e+06 | 27.4600 | 51.9933 | 0.0684 | 2.5247 |

```
feature_selected = data_in_table_MAF5(:, feature_monotonicity{:, :}>=confidence_level);
```

To achieve PCA, **Normalization** is applied

```
mean_train = mean(train_selected{:, :})
```

```
mean_train = 1x7
10^6 ×
    0.0000    0.0000    1.5293    0.0000    0.0000   -0.0000    0.0000
```

```
std_train = std(train_selected{:, :})
```

```
std_train = 1x7
10^6 ×
    0.0000    0.0000    2.3571    0.0000    0.0000    0.0000    0.0000
```

```
train_norm = (train_selected{:, :} - mean_train)./std_train;
coeff_pca = pca(train_norm)
```

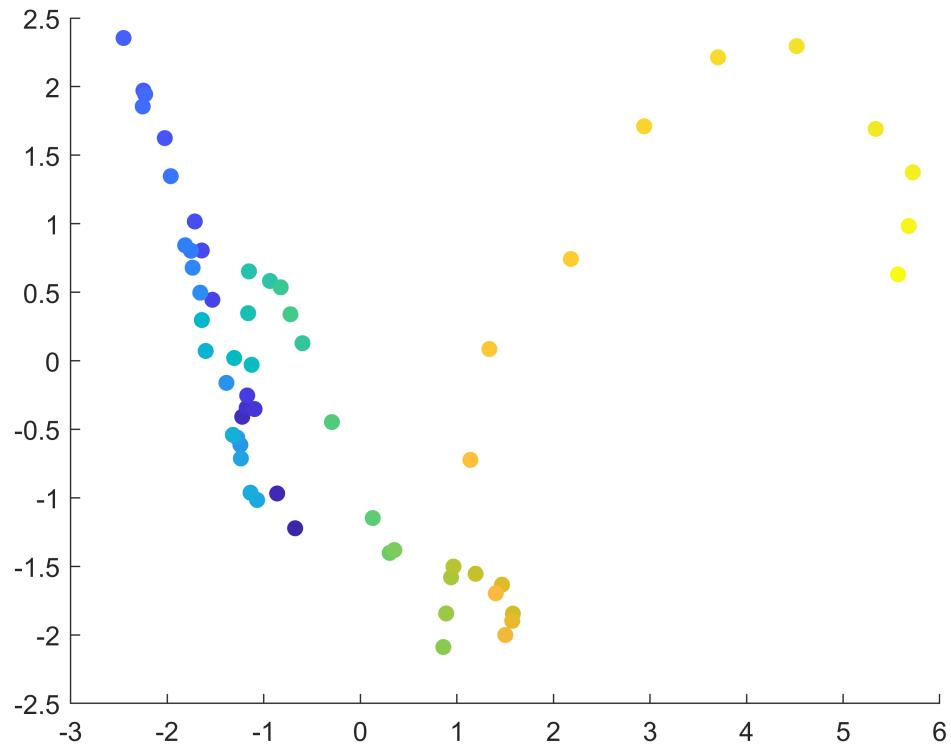
```
coeff_pca = 7x7
-0.3379    0.4891   -0.3755   -0.1178    0.0685   -0.6974    0.0252
-0.3555    0.4610   -0.3431   -0.1982   -0.0595    0.7073   -0.0176
 0.4326    0.2527    0.1360   -0.4779    0.7071    0.0440   -0.0146
 0.4457    0.2272   -0.0565   -0.1613   -0.4361   -0.0154    0.7281
 0.4461    0.2211   -0.0639   -0.2011   -0.4856   -0.0651   -0.6838
 0.4008    0.2213   -0.4156    0.7398    0.2500    0.0831   -0.0312
 -0.1187    0.5787    0.7367    0.3234   -0.0580    0.0059   -0.0137
```

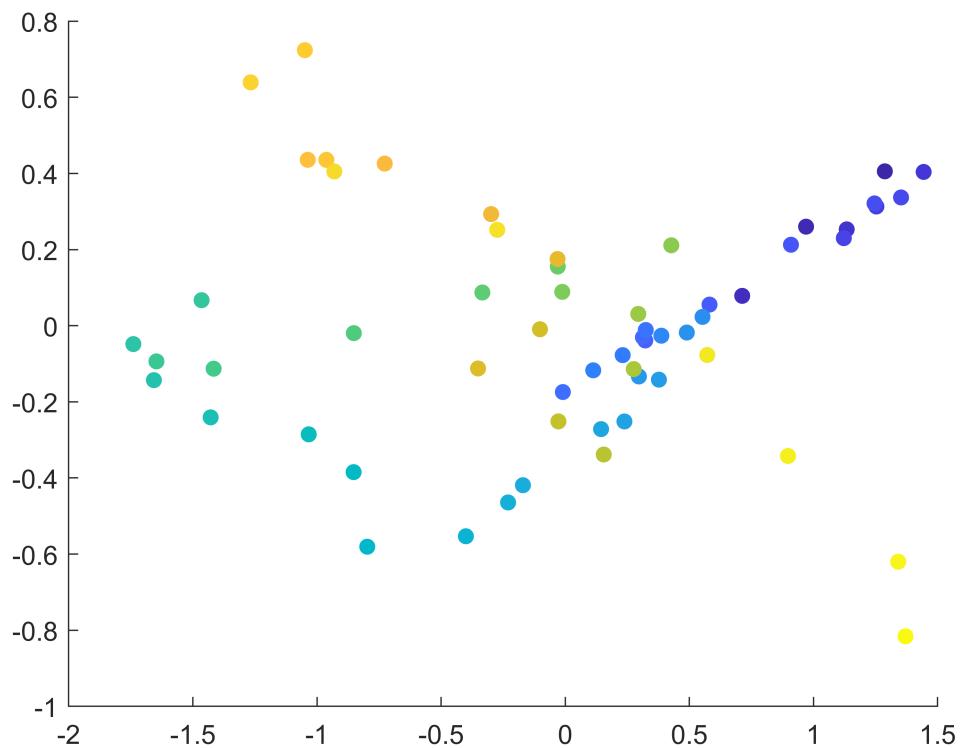
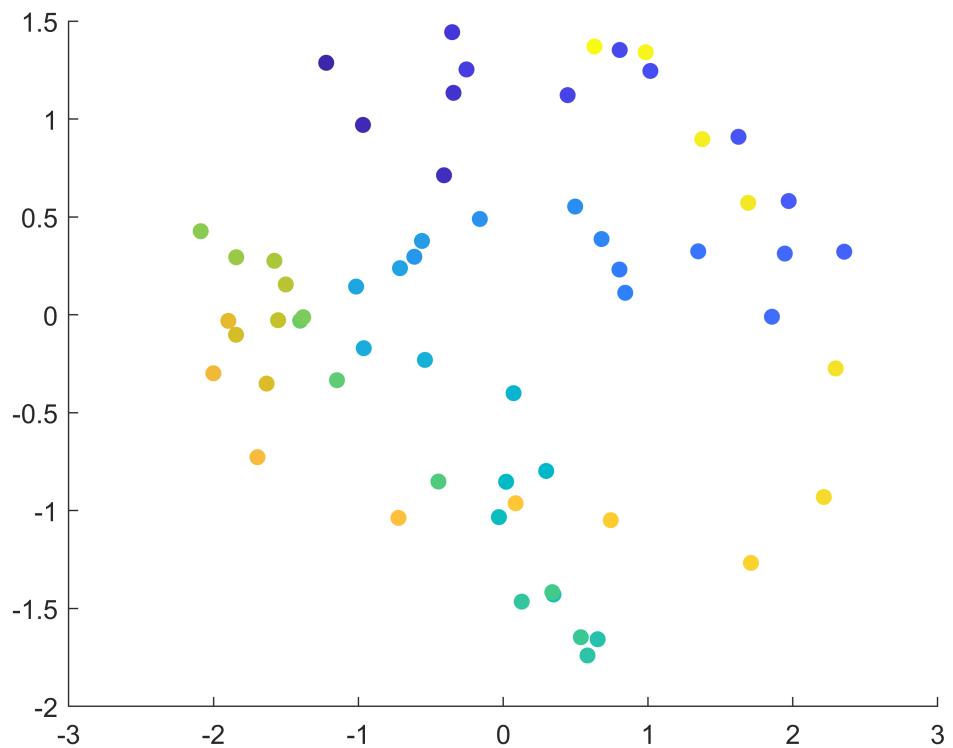
```
num_pca = length(coeff_pca(:, 1));
for idx = 1:num_pca
    PCA(:, idx) = (feature_selected{:, :} - mean_train) ./ std_train * coeff_pca(:, idx);
end
for idx = 1:num_pca-1
    figure;
```

```

scatter(PCA(:,idx), PCA(:,idx+1), [], 1:repair_day, 'filled');
if idx > 2
    break;
end
end

```





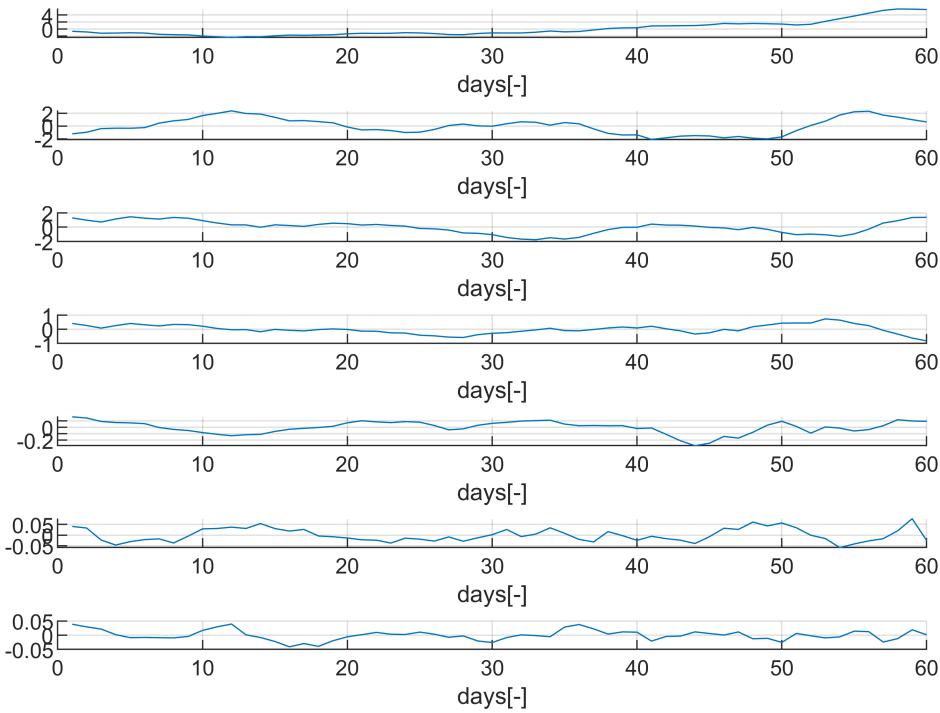
After the trends of PCA data, PCA 1 is chosen.

figure;

```

for idx =1:num_pca
    subplot(num_pca,1,idx); hold on; grid on ;
    plot(PCA(:,idx)); xlabel('days[-]');
end

```



Fault Monitoring and Warning

After the PCA graph above, PCA1 is chosen as the health indicator. PCA 1 data has the most large scale of value.

Using this graph, warning threshold, damaged threshold are chosen.

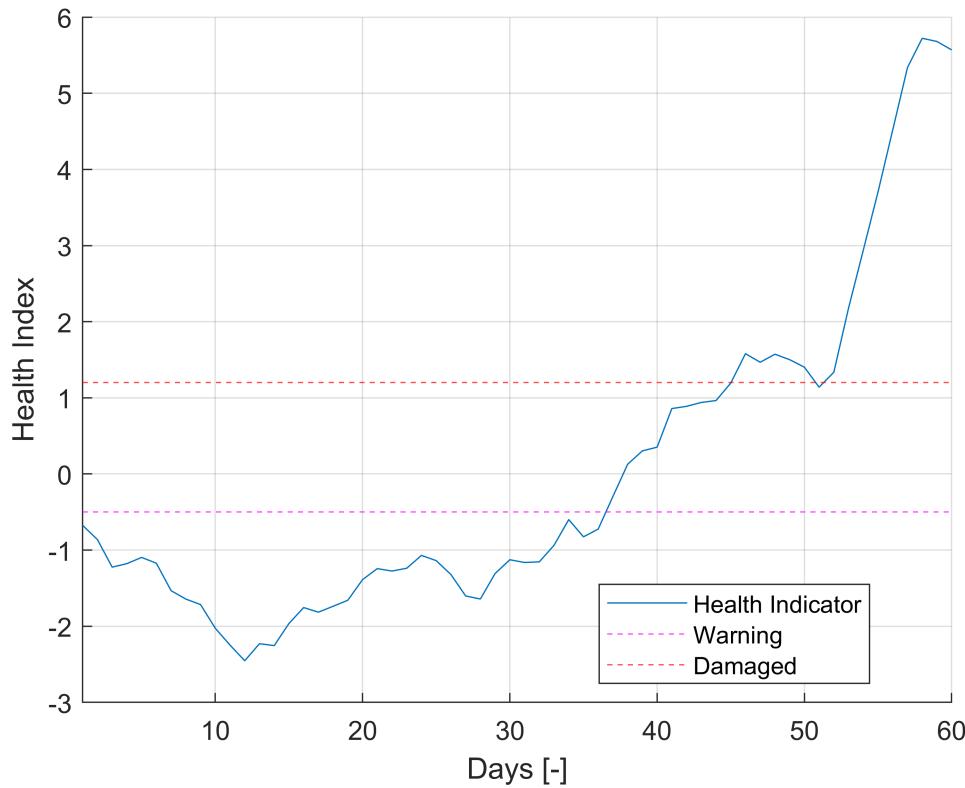
From day 30, the value reached -1 and this implies that the system is now in aware.

Around day 53, the health indicator is rapidly increasing.

```

health_indicator = PCA(:,1);
days = 1:repair_day;
figure; hold on; grid on;
plot(days, health_indicator,'DisplayName','Health Indicator'); xlabel('Days [-]'); ylabel('Heal
yline(-0.5,'m--','DisplayName','Warning'); yline(1.2,'r--','DisplayName','Damaged'); legend('Lo

```



After monotonicity, trendability and prognosability is tested.

They do not show distinct features and therefore, **withdrawed**.

```
% feature_trendability = trendability(data_in_table_MAF5);
% names    = string(feature_trendability.Properties.VariableNames);
% important_feature_trend = table2array(feature_trendability);
% figure; bar(categorical(names),important_feature_trend);
% xlabel('Features');ylabel('Trendability level');
```

RUL Estimation

```
healthIndicator = health_indicator - health_indicator(1);
% healthIndicator = healthIndicator(1:60);
threshold_HI    = max(healthIndicator);

mdl = exponentialDegradationModel(...  

    'Theta', 1, ...  

    'ThetaVariance', 1e6, ...  

    'Beta', 1, ...  

    'BetaVariance', 1e6, ...  

    'Phi', -1, ...  

    'NoiseVariance', (0.1*threshold_HI/(threshold_HI + 1))^2, ...  

    'SlopeDetectionLevel', 0.05);
```

```

total_day = 88;
estRULs = zeros(repair_day,1);
trueRULs = zeros(repair_day,1);
CI_RULs = zeros(repair_day,2);
pdfRULs = cell (repair_day,1);

% Create figures and axes for plot updating
figure
ax1 = subplot(2, 1, 1);
ax2 = subplot(2, 1, 2);
timeUnit = 'day';
for currentDay = 1:repair_day

    % Update model parameter posterior distribution
    update(mdl, [currentDay healthIndicator(currentDay)]);

    % Predict Remaining Useful Life
    [estRUL, CIRUL, pdfRUL] = predictRUL(mdl, ...
        [currentDay healthIndicator(currentDay)], ...
        threshold_HI);
    trueRUL = repair_day - currentDay + 1;

    % Updating RUL distribution plot
    helperPlotTrend(ax1, currentDay, healthIndicator, mdl, threshold_HI, timeUnit);
    helperPlotRUL(ax2, trueRUL, estRUL, CIRUL, pdfRUL, timeUnit)

    % Keep prediction results
    estRULs(currentDay) = estRUL;
    trueRULs(currentDay) = trueRUL;
    CIRULs(currentDay, :) = CIRUL;
    pdfRULs{currentDay} = pdfRUL;
    pause(0.1);
    legend('Location','best')
end

```

경고: 데이터가 지수 모델 가정을 충족하지 않습니다. 모델의 사전 파라미터를 확인하거나 다른 성능 저하 모델을 사용해 보십시오.

경고: 데이터가 지수 모델 가정을 충족하지 않습니다. 모델의 사전 파라미터를 확인하거나 다른 성능 저하 모델을 사용해 보십시오.

경고: 데이터가 지수 모델 가정을 충족하지 않습니다. 모델의 사전 파라미터를 확인하거나 다른 성능 저하 모델을 사용해 보십시오.

경고: 데이터가 지수 모델 가정을 충족하지 않습니다. 모델의 사전 파라미터를 확인하거나 다른 성능 저하 모델을 사용해 보십시오.

경고: 데이터가 지수 모델 가정을 충족하지 않습니다. 모델의 사전 파라미터를 확인하거나 다른 성능 저하 모델을 사용해 보십시오.

경고: 데이터가 지수 모델 가정을 충족하지 않습니다. 모델의 사전 파라미터를 확인하거나 다른 성능 저하 모델을 사용해 보십시오.

경고: 데이터가 지수 모델 가정을 충족하지 않습니다. 모델의 사전 파라미터를 확인하거나 다른 성능 저하 모델을 사용해 보십시오.

경고: 데이터가 지수 모델 가정을 충족하지 않습니다. 모델의 사전 파라미터를 확인하거나 다른 성능 저하 모델을 사용해 보십시오.

경고: 데이터가 지수 모델 가정을 충족하지 않습니다. 모델의 사전 파라미터를 확인하거나 다른 성능 저하 모델을 사용해 보십시오.

경고: 데이터가 지수 모델 가정을 충족하지 않습니다. 모델의 사전 파라미터를 확인하거나 다른 성능 저하 모델을 사용해 보십시오.

경고: 데이터가 지수 모델 가정을 충족하지 않습니다. 모델의 사전 파라미터를 확인하거나 다른 성능 저하 모델을 사용해 보십시오.

경고: 데이터가 지수 모델 가정을 충족하지 않습니다. 모델의 사전 파라미터를 확인하거나 다른 성능 저하 모델을 사용해 보십시오.

경고: 데이터가 지수 모델 가정을 충족하지 않습니다. 모델의 사전 파라미터를 확인하거나 다른 성능 저하 모델을 사용해 보십시오.

경고: 데이터가 지수 모델 가정을 충족하지 않습니다. 모델의 사전 파라미터를 확인하거나 다른 성능 저하 모델을 사용해 보십시오.

경고: 데이터가 지수 모델 가정을 충족하지 않습니다. 모델의 사전 파라미터를 확인하거나 다른 성능 저하 모델을 사용해 보십시오.

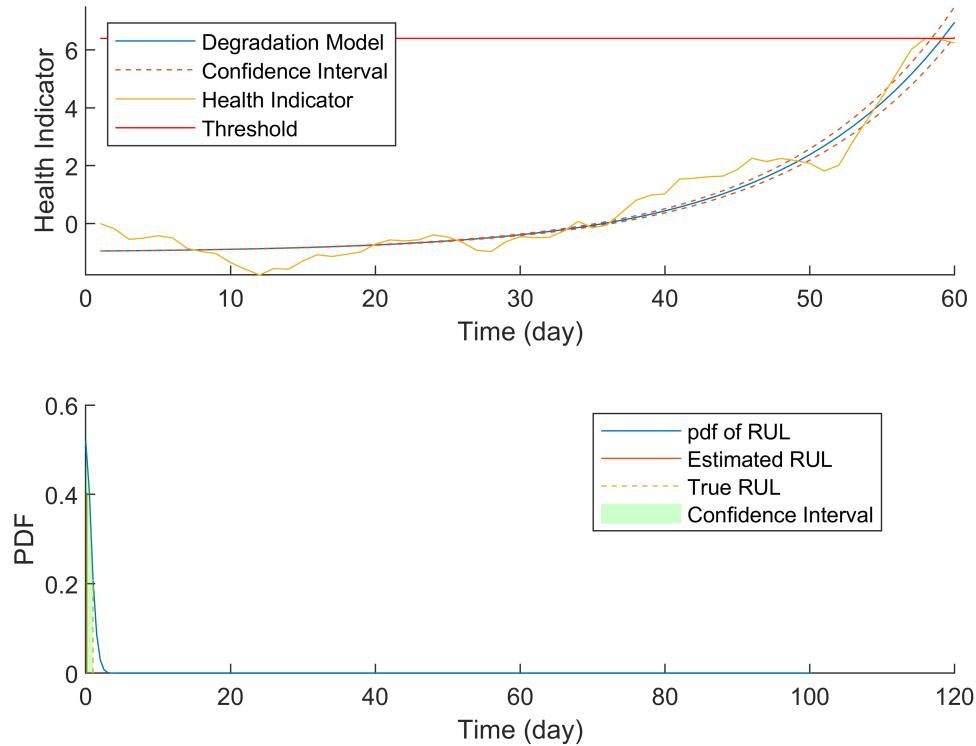
경고: 데이터가 지수 모델 가정을 충족하지 않습니다. 모델의 사전 파라미터를 확인하거나 다른 성능 저하 모델을 사용해 보십시오.

경고: 데이터가 지수 모델 가정을 충족하지 않습니다. 모델의 사전 파라미터를 확인하거나 다른 성능 저하 모델을 사용해 보십시오.

경고: 데이터가 지수 모델 가정을 충족하지 않습니다. 모델의 사전 파라미터를 확인하거나 다른 성능 저하 모델을 사용해 보십시오.

경고: 데이터가 지수 모델 가정을 충족하지 않습니다. 모델의 사전 파라미터를 확인하거나 다른 성능 저하 모델을 사용해 보십시오.

Day 60: Degradation detected!



Before the break point, the health indicator decreases and this might not satisfy the exponential degradation model, if this happens, this implies the fact that the data is healthy and clean.

Discussion

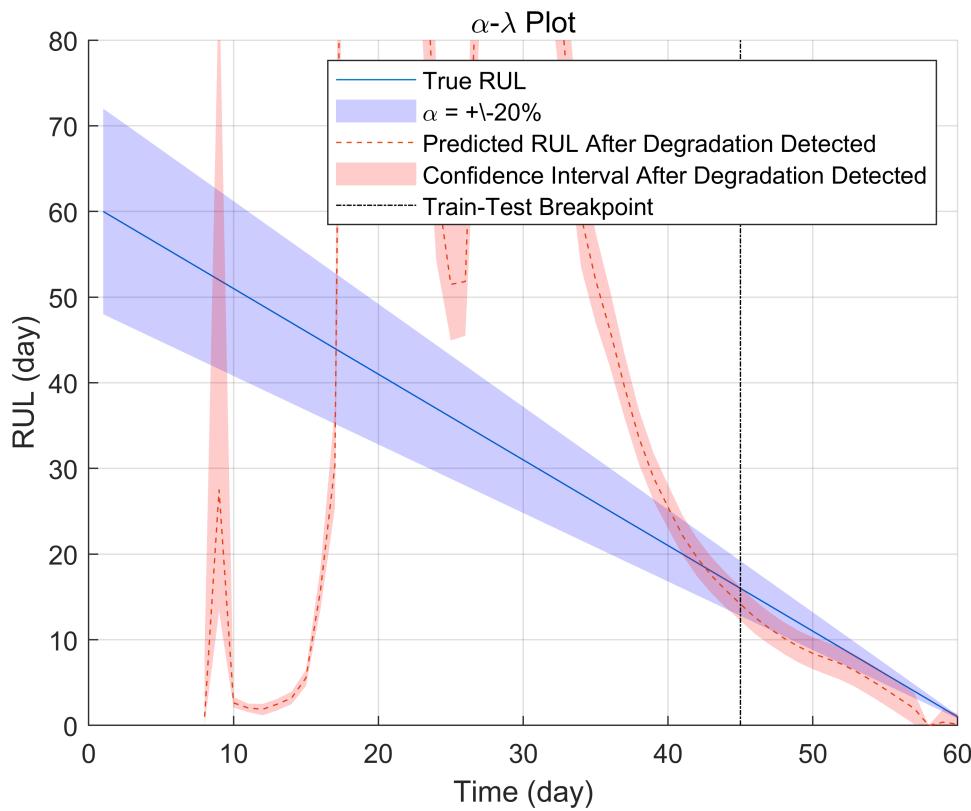
The red graph is the increase in prediction accuracy through learning. The data is compared as if the prediction matches with true RUL.

```
% Performance Analysis
data_in_table_MAF5.Date = (1:repair_day)';
```

```

alpha = 0.2;
detectTime = mdl.SlopeDetectionInstant;
prob = helperAlphaLambdaPlot(alpha, trueRULs, estrULs, CIRULs, ...
    pdfRULs, detectTime, break_point, timeUnit);
title('alpha-lambda Plot')

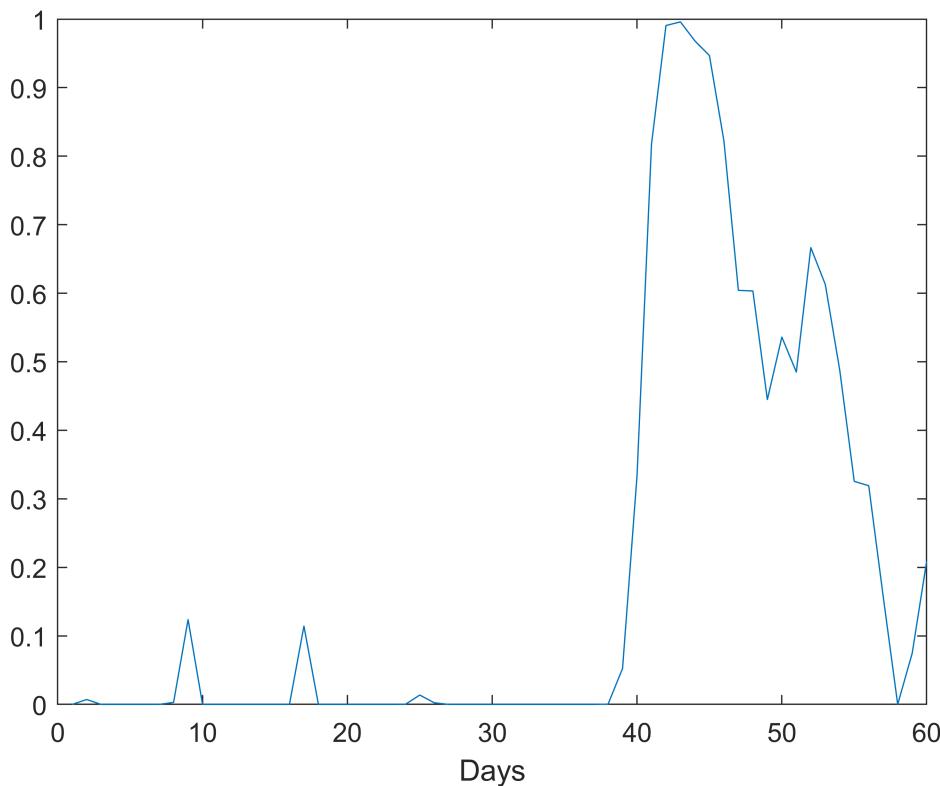
```



```

figure; plot(prob); xlabel('Days');

```



Conclusion

This Lab deals with Health Monitoring approach for bearing data. Basic time features of the time-series dataset are extracted at the **Procedure**. Important features are selected. PCA is used to replace the whole features for health indicator. The health indicator is selected as it shows dramatic degradation from day 30 upto replacement. For conservative approach to estimate the degradation, break point is assumed to be day 45.

The threshold levels of **warning** and **damaged** are -0.5 and 1.2 respectively for the selected health indicator. -0.5 is as the health indicator tends to increase at the level of 0.5. 1.2 value is selected as the degradation accelerates to the maximum value.

As the $\alpha\text{-}\lambda$ does not match with the true RUL until about day 35, the RUL shows that the data is absolutely normal at the first half. The predicted RUL starts to match from day 40.

Because of the wear of industrial equipment, the degradation model may look exponential or linear. At this point, the degradation model is exponential where the bearing starts to crack more rapidly as it cracks.

Caution: The data of each day is extracted hour by hour, but this data is **concatenated** as a day, this might **affect the frequency/time features**. The author assumed that can be ignored and treated as random noise. To prevent this happening, try averaging the total day dataset as there are 24 dataset within a day which is enough and using **Central Limit Theorem**, the random noises can be removed. This may bring better effect to the result.

Functions

```

function xfeature = get_features(x)
% This is based on the function timeFeatures()
% Input: Raw Data (Time-series)
% Output: Struct type
%
% |---Time features(e.g. RMS, peak2peak, kurtosis)
% |---Frequency features(e.g. FFT, rmsf)
N = length(x);
xfeature.mean=mean(x);
xfeature.std=std(x);

xfeature.rms      =sqrt(sum(x.^2)/N);
xfeature.sra      = (sum(sqrt(abs(x)))/N)^2;
xfeature.aav      =sum(abs(x))/N;
xfeature.energy   =sum(x.^2);
xfeature.peak     =max(x);
xfeature.ppv      =peak2peak(x);
xfeature.if       =xfeature.peak/xfeature.aav;
xfeature.sf       =xfeature.rms/xfeature.aav;
xfeature.cf       =xfeature.peak/xfeature.rms;
xfeature.mf       =xfeature.peak/xfeature.sra;
xfeature.sk       =skewness(x);
xfeature.kt       =kurtosis(x);
p1 = fft(x);
p1(2:end-1)= 2.0 * p1(2:end-1);
xfeature.fft = p1;

xfeature.rmsf= sqrt(1/N *sum(xfeature.fft.^2));
end

function data_table = cell2table(data)
% Every data is grouped by a cell, but now they are collected together in
% terms of Time-Series
rms_data    = [];    sra_data    = [];    aav_data    = [];    energy_data= [];
peak_data   = [];    ppv_data   = [];    if_data     = [];    sf_data     = [];
cf_data    = [];    mf_data    = [];    sk_data    = [];    kurto_data = [];
SKmean_data= [];    SK_std_data= [];    SK_kt_data = [];    fft_max_idx= [];

days = 1:88;

data_table = table;
for idx = days
    rms_data    = [rms_data    ;data{idx,2}.rms    ];
    sra_data   = [sra_data   ;data{idx,2}.sra   ];
    aav_data   = [aav_data   ;data{idx,2}.aav   ];
    energy_data= [energy_data;data{idx,2}.energy];
    peak_data  = [peak_data  ;data{idx,2}.peak  ];
    ppv_data   = [ppv_data   ;data{idx,2}.ppv   ];
    if_data    = [if_data    ;data{idx,2}.if    ];
    sf_data    = [sf_data    ;data{idx,2}.sf    ];
    cf_data   = [cf_data   ;data{idx,2}.cf    ];

```

```

mf_data      = [mf_data      ;data{idx,2}.mf      ];
sk_data      = [sk_data      ;data{idx,2}.sk      ];
kurto_data   = [kurto_data   ;data{idx,2}.kt      ];
SK_std_data= [SK_std_data;data{idx,2}.SKstd ];
SK_kt_data  = [SK_kt_data  ;data{idx,2}.SKkurt];
SKmean_data= [SKmean_data;data{idx,2}.SKmean];
fft_max_idx= [fft_max_idx;find(data{idx,2}.fft==max(data{idx,2}.fft))];

end
data_table.fft_max_idx = fft_max_idx;
data_table.IF          = if_data      ;
data_table.SF          = sf_data      ;
data_table.CF          = cf_data      ;
data_table.MF          = mf_data      ;
data_table.RMS         = rms_data    ;
data_table.SRA         = sra_data    ;
data_table.AAV         = aav_data    ;
data_table.energy      = energy_data;
data_table.PEAK         = peak_data   ;
data_table.PEAK2PEAK   = ppv_data   ;
data_table.SKEW         = sk_data     ;
data_table.KURTOSIS    = kurto_data  ;
data_table.SK_STD       = SK_std_data;
data_table.SK_MEAN      = SKmean_data;
data_table.SK_KURTOSIS = SK_kt_data ;

```

end

Reference

[1] [Github](#)

[2] Rauber, T. W., de Assis Boldt, F., & Varejao, F. M. (2014). Heterogeneous feature models and feature selection applied to bearing fault diagnosis. *IEEE Transactions on Industrial Electronics*, 62(1), 637-646.