All-Files Analysis

Import All Data

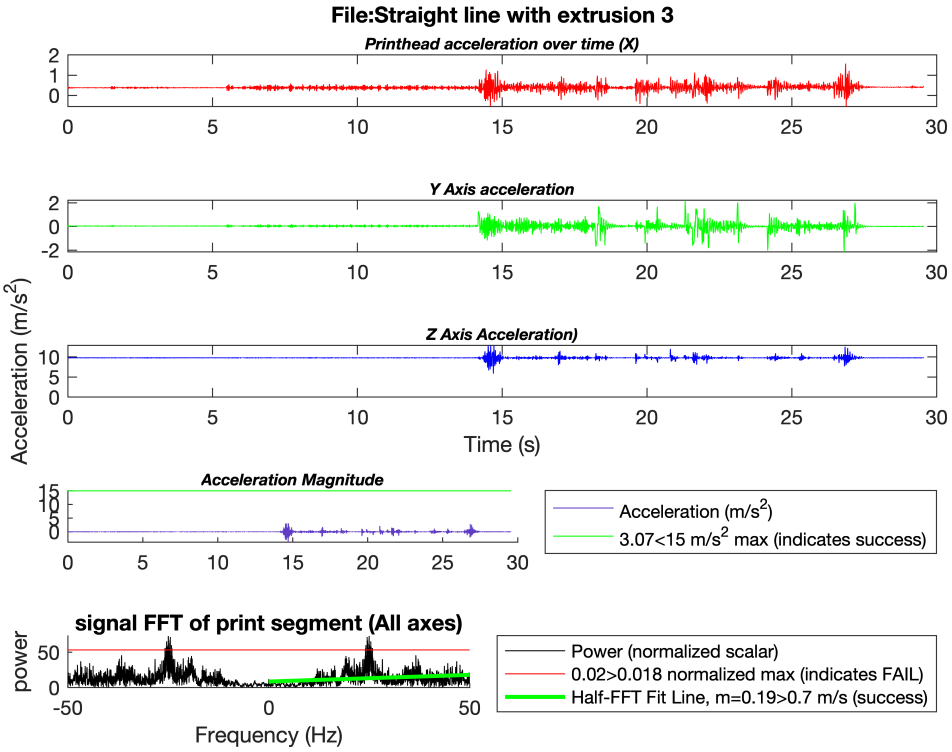
```
clear;

datasets = dir('data/*.mat');
numfiles = length(datasets);
% numfiles = 1;
mydata = cell(1, numfiles);

% Import Data
for k = 1:numfiles
  mydata{k} = importdata(strcat('data/',datasets(k).name));
  mydata{k}.Source = datasets(k).name;
end

% Run simple analysis on all
% for k = 1:length(mydata)
%     sample = mydata(k);
%     Acceleration = sample{1}.Acceleration;
%     figure()
%     stackedplot(Acceleration);
%     title({'Time table visualization of printhead acceleration';strcat('\fontsize{8}\itFile:
% end
```

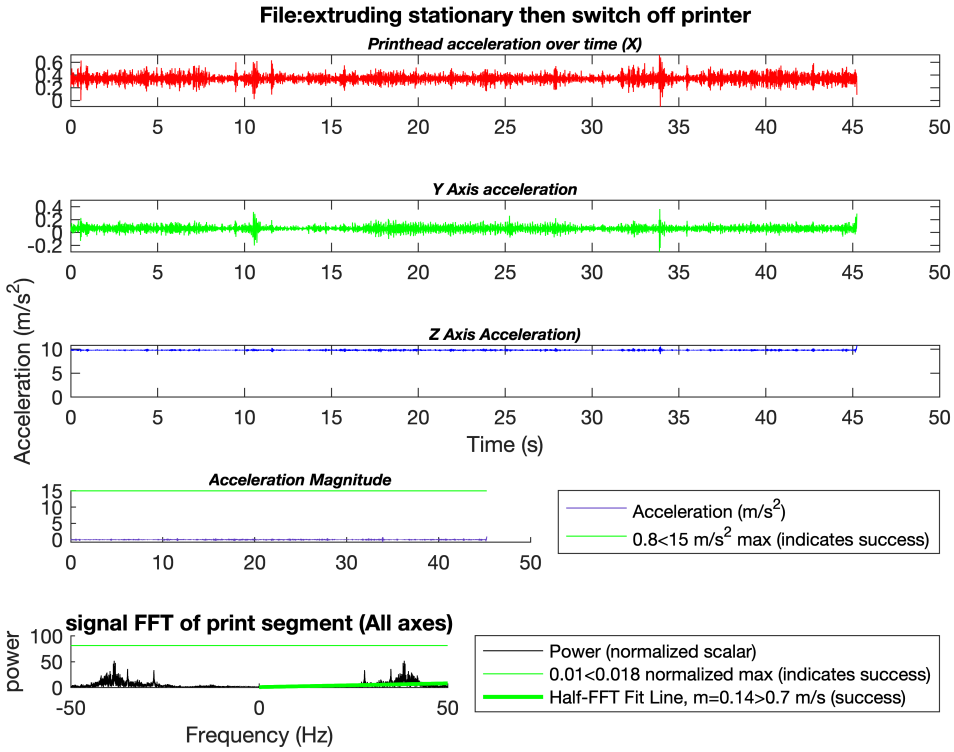
Plot All Data

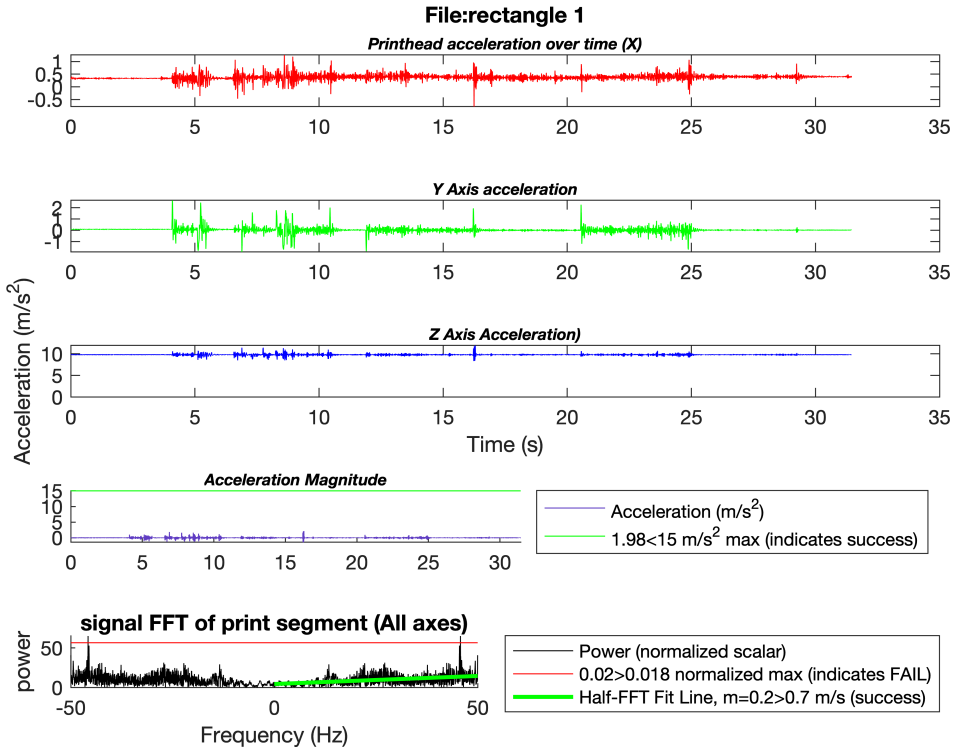
```
tic
for k = 1:length(mydata)
    sample = mydata(k);
    Acceleration = sample{1}.Acceleration;
    t = linspace(0,height(Acceleration)/100,height(Acceleration));
    figure()
    a = timetable2table(Acceleration);
    accelData = [a.X a.Y a.Z]';

    subplot(5,1,1);
    plot(t,accelData(1,:), 'r')
    title({strcat('File: ',sample{1}.Source(1:end-4));"\fontsize{8}\itPrinthead acceleration ov

    subplot(5,1,2);
    plot(t,accelData(2,:), 'g')
    title("\fontsize{8}\itY Axis acceleration")

    subplot(5,1,3);
    plot(t,accelData(3,:), 'b')
    title("\fontsize{8}\itZ Axis Acceleration)")
    xlabel("Time (s)")
    ylabel("Acceleration (m/s^2)")
    hold on;
    plot([0,0],[0,0])
```

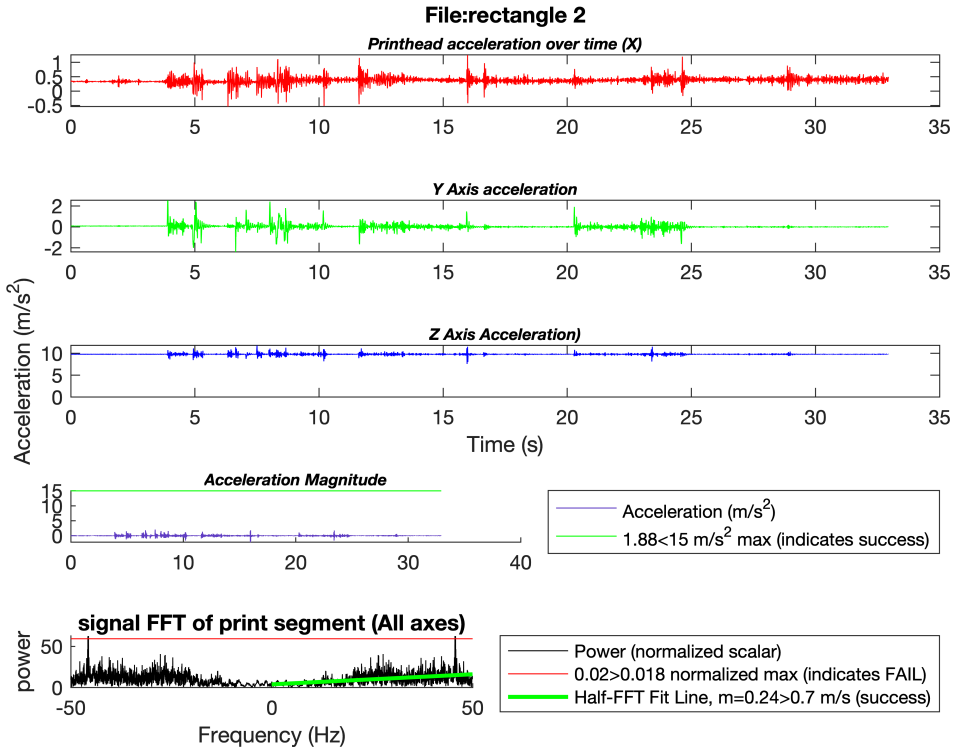
```
subplot(5,1,4);
accelData = [a.X a.Y a.Z]';
x = vecnorm(accelData);
x = x - mean(x);
hold on;
plot(t,x, 'Color', [99, 64, 199]./255)
title("\fontsize{8}\itAcceleration Magnitude")

[success, level] = classify_accel_threshold(x);
if success
    plot([0 length(x)/100], [15 15], 'g')
    legend('Acceleration (m/s^2)', strcat(string(round(level,2)),'<15 m/s^2 max (indicates
else
    plot([0 length(x)/100], [15 15], 'r')
    legend('Acceleration (m/s^2)', strcat(string(round(level,2)),'>15 m/s^2 max (indicates
    set(gcf, 'color', [1 0.4 0.4])
end
hold off;

subplot(5, 1, 5);
N = length(x);
Fs = 100; % sampling rate of 100Hz
f = linspace(-Fs/2, Fs/2 - Fs/N, N) + Fs/(2*N)*mod(N, 2);
FA = fft(x);
fftOutput = fftshift(abs(FA));
hold on;
plot(f,fftOutput,'Color',[0 0 0]./255);
title("signal FFT of print segment (All axes)")
xlabel("Frequency (Hz)")
ylabel("power")
sampleLength = length(fftOutput);
normalizedSample = fftOutput./sampleLength;
[success, level] = classify_fft_threshold(normalizedSample, 0.018);
if success
    plot([-50 50], ones(2)*0.018*sampleLength, 'g')
    legend('Power (normalized scalar)', strcat(string(round(level,2)),'<0.018 normalized ma
else
    plot([-50 50], ones(2)*0.018*sampleLength, 'r')
    legend('Power (normalized scalar)', strcat(string(round(level,2)),'>0.018 normalized ma
    % set(gcf, 'color', [1 0.4 0.4])
end

d = round(length(fftOutput)/2);
halfF = f(d:end);
halfFFToutput = fftOutput(d:end);
% plot(halfF,halfFFToutput);
p = polyfit(halfF,halfFFToutput,1);

if p(1) > -0.7
    plot(halfF,polyval(p,halfF),'g','LineWidth',2.0,'DisplayName',strcat('Half-FFT Fit Line
else
    plot(halfF,polyval(p,halfF),'r','LineWidth',2.0,'DisplayName',strcat('Half-FFT Fit Line
end
```

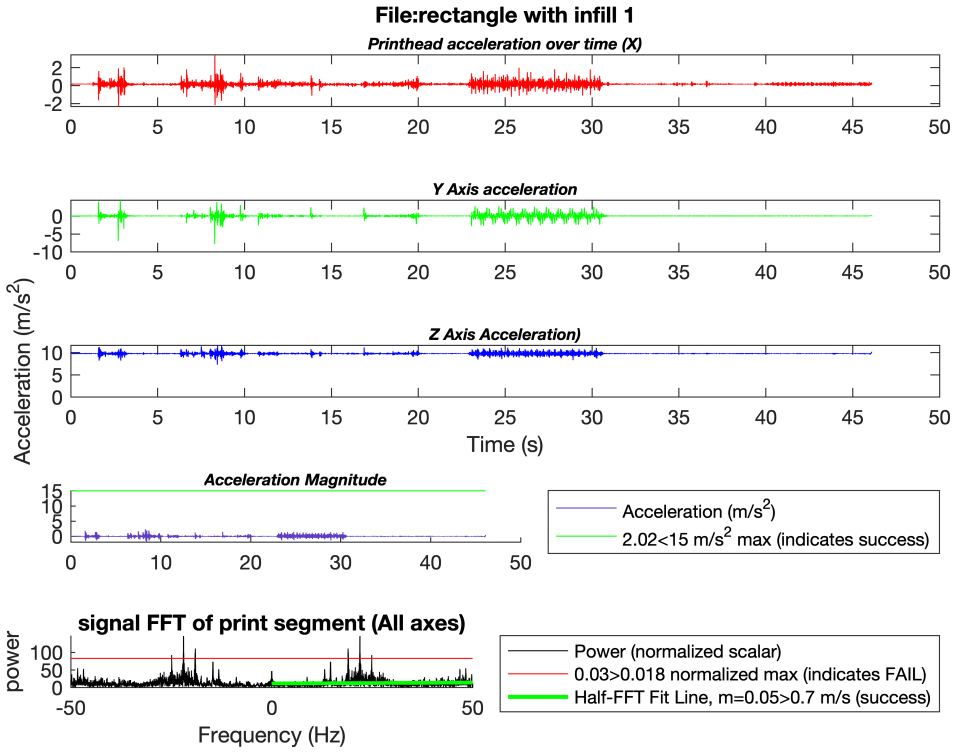
```
    saveas(gcf, strcat('./imgs/',sample{1}.Source(1:end-4),'.png'))
end
```

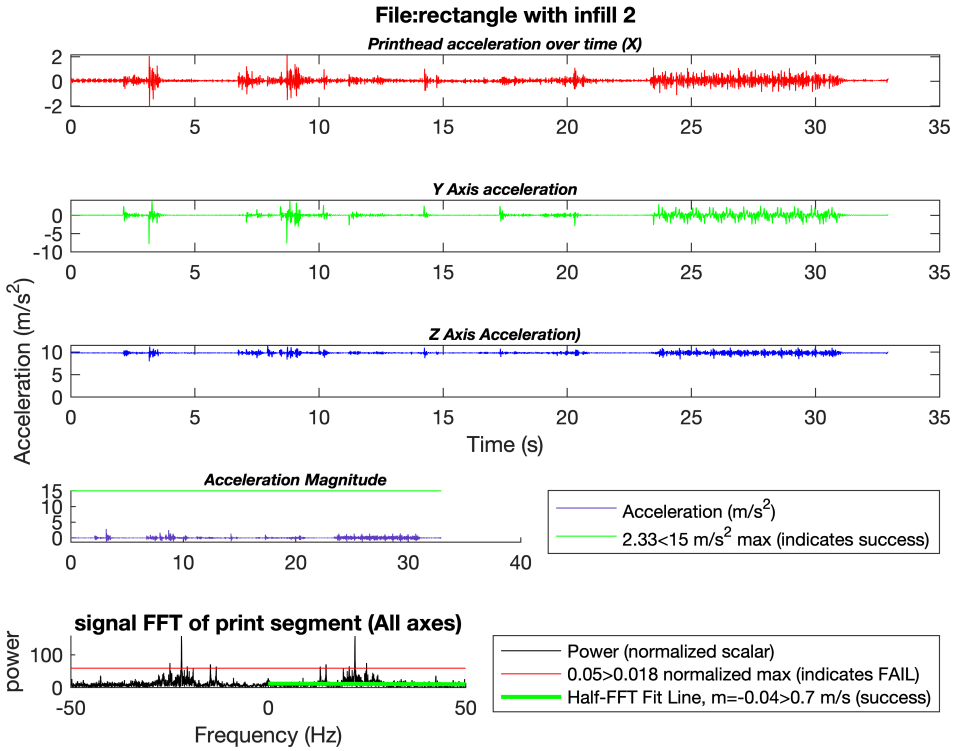


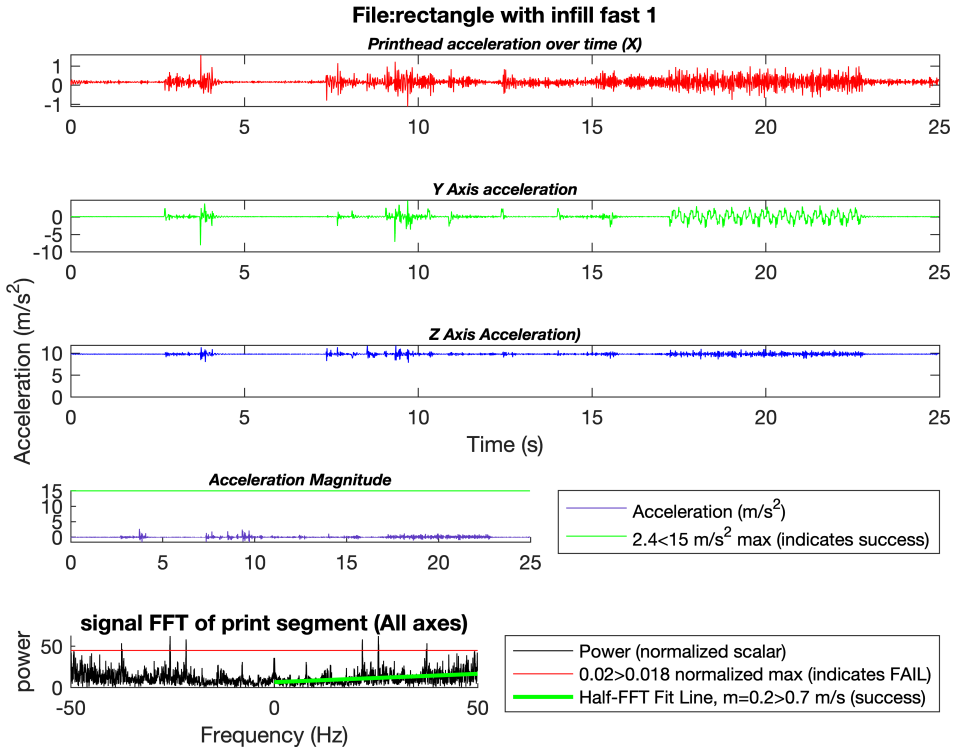

File:extruding stationary then switch off printer
Printhead acceleration over time (X)
Y Axis acceleration
Z Axis Acceleration)
Acceleration (m/s2)
Time (s)
Acceleration Magnitude
Acceleration (m/s2)
0.8<15 m/s2 max (indicates success)
signal FFT of print segment (All axes)
power
Frequency (Hz)
Power (normalized scalar)
0.01<0.018 normalized max (indicates success)
Half-FFT Fit Line, m=0.14>0.7 m/s (success)

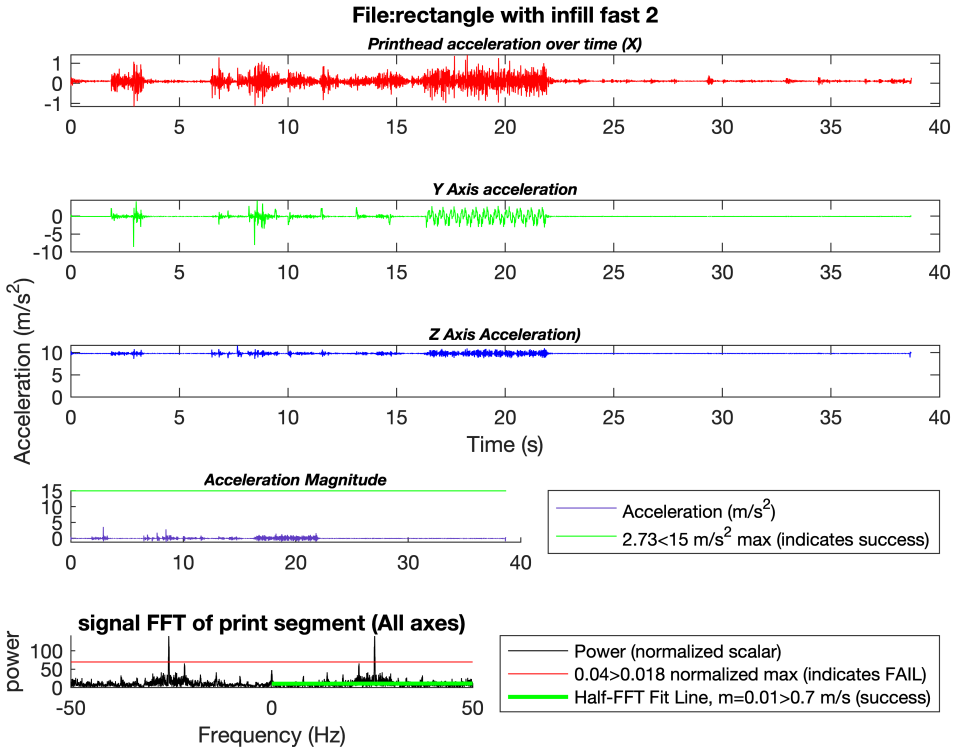

File:rectangle 1
Printhead acceleration over time (X)
Y Axis acceleration
Z Axis Acceleration)
Acceleration (m/s2)
Time (s)
Acceleration Magnitude
Acceleration (m/s2)
1.98<15 m/s2 max (indicates success)
signal FFT of print segment (All axes)
power
Frequency (Hz)
Power (normalized scalar)
0.02>0.018 normalized max (indicates FAIL)
Half-FFT Fit Line, m=0.2>0.7 m/s (success)

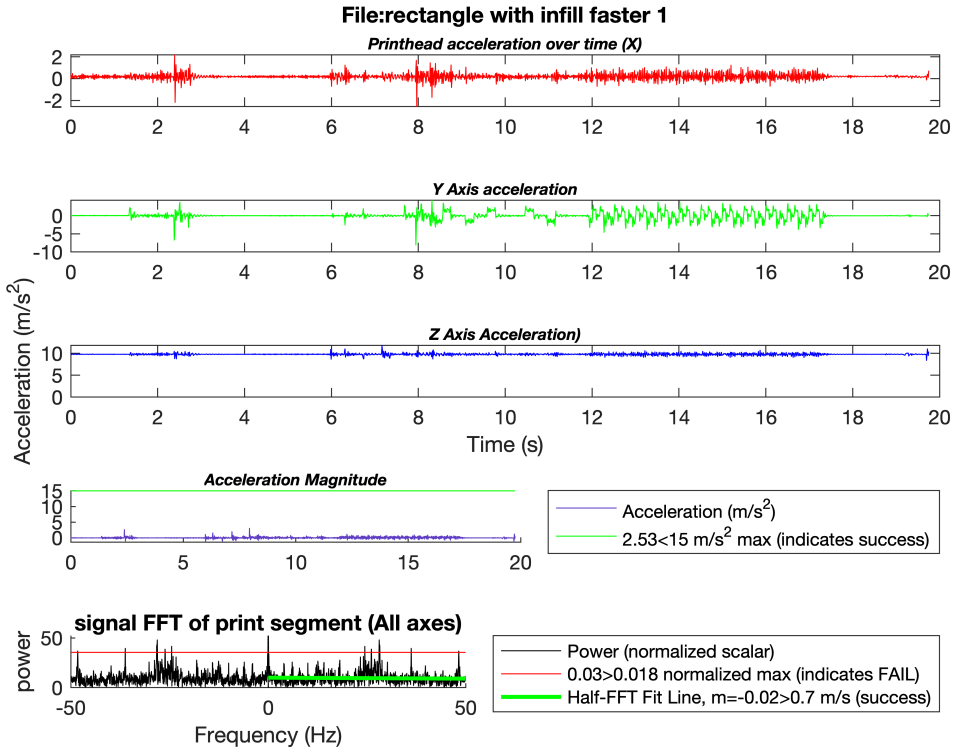

File:rectangle 2
Printhead acceleration over time (X)
Y Axis acceleration
Z Axis Acceleration)
Acceleration (m/s2)
Time (s)
Acceleration Magnitude
Acceleration (m/s2)
1.88<15 m/s2 max (indicates success)
signal FFT of print segment (All axes)
power
Frequency (Hz)
Power (normalized scalar)
0.02>0.018 normalized max (indicates FAIL)
Half-FFT Fit Line, m=0.24>0.7 m/s (success)

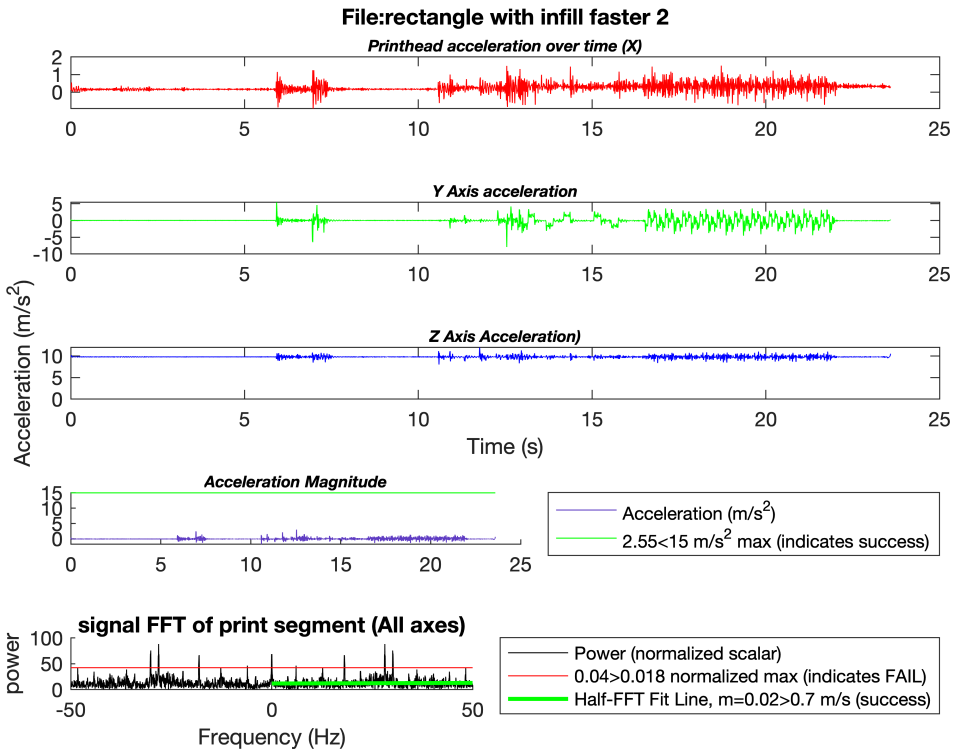

File:rectangle with infill 1
Printhead acceleration over time (X)
Y Axis acceleration
Z Axis Acceleration)
Acceleration (m/s2)
Time (s)
Acceleration Magnitude
Acceleration (m/s2)
2.02<15 m/s2 max (indicates success)
signal FFT of print segment (All axes)
power
Frequency (Hz)
Power (normalized scalar)
0.03>0.018 normalized max (indicates FAIL)
Half-FFT Fit Line, m=0.05>0.7 m/s (success)

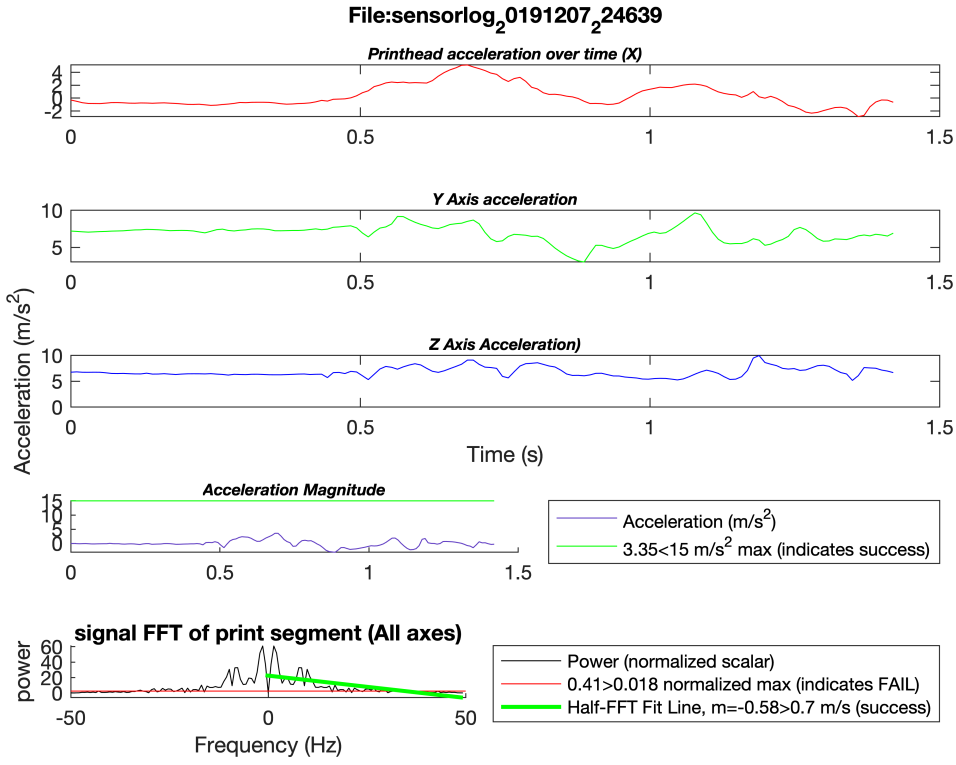

File:rectangle with infill 2
Printhead acceleration over time (X)
Y Axis acceleration
Z Axis Acceleration)
Acceleration (m/s2)
Time (s)
Acceleration Magnitude
Acceleration (m/s2)
2.33<15 m/s2 max (indicates success)
signal FFT of print segment (All axes)
power
Frequency (Hz)
Power (normalized scalar)
0.05>0.018 normalized max (indicates FAIL)
Half-FFT Fit Line, m=-0.04>0.7 m/s (success)


File:rectangle with infill fast 1
Printhead acceleration over time (X)
Y Axis acceleration
Z Axis Acceleration)
Acceleration (m/s2)
Time (s)
Acceleration Magnitude
Acceleration (m/s2)
2.4<15 m/s2 max (indicates success)
signal FFT of print segment (All axes)
power
Frequency (Hz)
Power (normalized scalar)
0.02>0.018 normalized max (indicates FAIL)
Half-FFT Fit Line, m=0.2>0.7 m/s (success)


File:rectangle with infill fast 2
Printhead acceleration over time (X)
Y Axis acceleration
Z Axis Acceleration)
Acceleration (m/s2)
Time (s)
Acceleration Magnitude
Acceleration (m/s2)
2.73<15 m/s2 max (indicates success)
signal FFT of print segment (All axes)
power
Frequency (Hz)
Power (normalized scalar)
0.04>0.018 normalized max (indicates FAIL)
Half-FFT Fit Line, m=0.01>0.7 m/s (success)


File:rectangle with infill faster 1
Printhead acceleration over time (X)
Y Axis acceleration
Z Axis Acceleration)
Acceleration (m/s2)
Time (s)
Acceleration Magnitude
Acceleration (m/s2)
2.53<15 m/s2 max (indicates success)
signal FFT of print segment (All axes)
power
Frequency (Hz)
Power (normalized scalar)
0.03>0.018 normalized max (indicates FAIL)
Half-FFT Fit Line, m=-0.02>0.7 m/s (success)






```
toc
```

```
Elapsed time is 26.361364 seconds.
```

Big Plots

```
tic
for k = 1:length(mydata)
    sample = mydata(k);
    Acceleration = sample{1}.Acceleration;
    t = linspace(0,height(Acceleration)/100,height(Acceleration));
    figure()
    a = timetable2table(Acceleration);
    accelData = [a.X a.Y a.Z]';

%     subplot(5,1,1);
%     plot(t,accelData(1,:), 'r')


%     subplot(5,1,2);
%     plot(t,accelData(2,:), 'g')
%     title("\fontsize{8}\itY Axis acceleration")

%     subplot(5,1,3);
%     plot(t,accelData(3,:), 'b')
%     title("\fontsize{8}\itZ Axis Acceleration)")
%     xlabel("Time (s)")
%     ylabel("Acceleration (m/s^2)")
%     hold on;
%     plot([0,0],[0,0])

%     subplot(5,1,4);
    accelData = [a.X a.Y a.Z]';
    x = vecnorm(accelData);
    x = x - mean(x);
%     hold on;
%     plot(t,x, 'Color', [99, 64, 199]./255)
%     title("\fontsize{8}\itAcceleration Magnitude")

    [success, level] = classify_accel_threshold(x);
    if success
%         plot([0 length(x)/100], [15 15], 'g')
%         legend('Acceleration (m/s^2)', strcat(string(round(level,2)),'/15 m/s^2 max (indicate
    else
        plot([0 length(x)/100], [15 15], 'r')
%         legend('Acceleration (m/s^2)', strcat(string(round(level,2)),'/15 m/s^2 max (indicate
%         set(gcf, 'color', [1 0.4 0.4])
    end

    N = length(x);
    Fs = 100; % sampling rate of 100Hz
    f = linspace(-Fs/2, Fs/2 - Fs/N, N) + Fs/(2*N)*mod(N, 2);
    FA = fft(x);
    fftOutput = fftshift(abs(FA));
    hold on;
    plot(f,fftOutput,'Color',[99 64 199]./255);
    title({strcat('File: ',sample{1}.Source(1:end-4));"\fontsize{8}\itSignal FFT of print segme
```

```
    xlabel("Frequency (Hz)")
    ylabel("power")
    sampleLength = length(fftOutput);
    normalizedSample = fftOutput./sampleLength;
    [success, level] = classify_fft_threshold(normalizedSample, 0.018);
    if success
        plot([-50 50], ones(2)*0.018*sampleLength, 'g')
        legend('Power (normalized scalar)', strcat(string(round(level,2)),'<0.018 normalized ma
    else
        plot([-50 50], ones(2)*0.018*sampleLength, 'r')
        legend('Power (normalized scalar)', strcat(string(round(level,2)),'>0.018 normalized ma
        % set(gcf, 'color', [1 0.4 0.4])
    end

    d = round(length(fftOutput)/2);
    halfF = f(d:end);
    halfFFToutput = fftOutput(d:end);
    % plot(halfF,halfFFToutput);
    p = polyfit(halfF,halfFFToutput,1);

    if p(1) > -0.7
        plot(halfF,polyval(p,halfF),'g','LineWidth',2.0,'DisplayName',strcat('Half-FFT Fit Line
    else
        plot(halfF,polyval(p,halfF),'r','LineWidth',2.0,'DisplayName',strcat('Half-FFT Fit Line
    end



    hold off;

    saveas(gcf, strcat('./imgs/fourier/',sample{1}.Source(1:end-4),'.png'))
end
```