MECH 880: Numerical Methods

Professor: Dr. Florin Bobaru

Author: Ricardo Jacome

Wavelet Transforms – Introduction and Application in Signal Analysis

Abstract: Signal analysis is used to decompose a signal into elements such as noise or other elementary functions in order to filter or categorized the sampled information. Many methods for decomposition and filtering exists such as Fourier Series and Fast Fourier Transforms (FFT). This paper takes a more general method known as Wavelets with its corresponding Wavelet Transforms. A basic review of the method is provided along with some examples and applications of Wavelets in engineering applications such as acceleration analysis. The Wavelets are compared with the FFT method for filtering a noisy signal. The Wavelet method is performed with the MATLAB wavelet tool that permits decomposition and analysis of the accelerations experimented by a vehicle under harsh braking scenarios such as different surface roads.

Background

Waves are an oscillating function defined in time and space, such as sinusoids. These sinusoids are used as basis functions to construct any periodic signal. Such construction is known as a Fourier Series representation. This is done to filter signals by finding the frequency content that represents the desired signal and removing all other frequencies that are categorized as noise. This method has limitations in terms of locating the time event of the frequencies captured. Similarly, this method is limited to periodic functions only. For this reason, wavelets were introduced to compensate for the limitations on representing signals with Fourier Series. Wavelets can be interpreted as a small wave with its energy concentrated in a position in time. These wavelets serve as the new basis functions that can decompose signals that are non-periodic while maintaining information about both frequency and time contents.

To exemplify a signal decomposition in wavelets, a Fourier series decomposition is shown in Figure for a direct comparison. Instead of sines and cosines, the wavelet decomposition is composed of two functions: The Scaling Function $\varphi_{j_o,k}(t)$ and the Wavelet Function $\psi_{j,k}(t)$. Similar to sines and cosines, both the scaling and wavelet functions are orthogonal functions that are linearly independent of each other.

$$Fourier\ Series:\ f(t) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n cos(n\pi t) + \sum_{n=1}^{\infty} \mathrm{b_n} \sin(n\pi t) \qquad n \in \mathbb{Z}$$

$$Wavelet\ Decomposition: g(t) = \sum_{k=-\infty}^{\infty} c_{jo}(k)\, \varphi_{j_o,k}(t) + \sum_{k=-\infty}^{\infty} \sum_{j=j_o}^{\infty} d_j(k)\, \psi_{j,k}(t)$$

The coefficients c and d can be found through the principle of inner product for orthogonal functions. These coefficients receive the name of Discrete Wavelet Transform (DWT) Coefficients, which is analogous to the FFT Coefficients for signal decomposition. In general, the "d" coefficients serve to make a level of decomposition [1]. Instead of a basis function in the form of a sine or a cosine, wavelets have the advantage of an infinite range of Wavelet functions available for signal construction. In general, the scaling function has the following form below, where $h(n)$ consists of scaling coefficients.

$$\varphi(t) = \sum_n h(n)\sqrt{2}\,\varphi(2t - n), \qquad n \in \mathbb{Z}$$

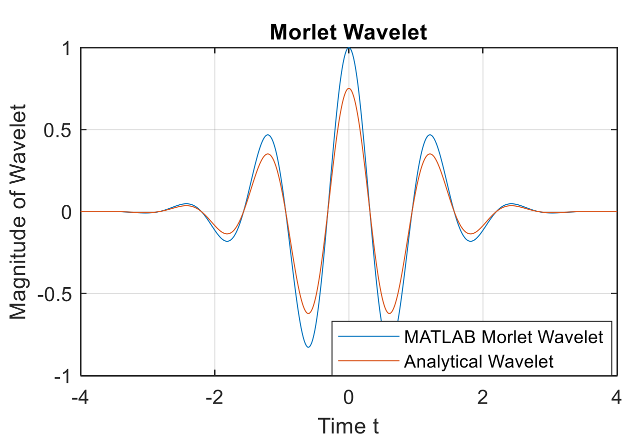
To define a Wavelet Function, the scaling function from before is used with a shift, along with some Wavelet Coefficients $h_1(n)$ as shown below.

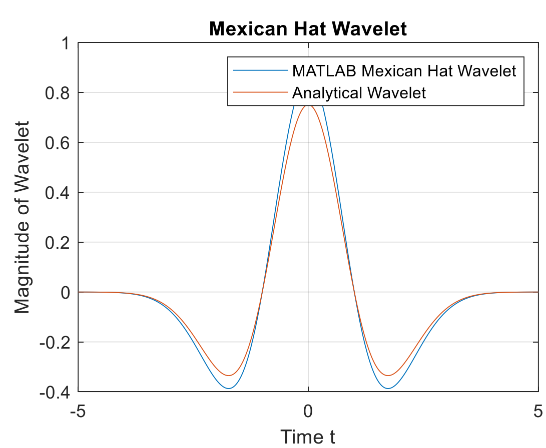
$$\psi(t) = \sum_n h_1(n)\sqrt{2}\,\varphi(2t - n), \qquad n \in \mathbb{Z}$$

Depending on the signal to be analyzed, there are many wavelet functions that can be used. Similarly, there are families of wavelets within each type [2]. These families each are classified by how coarse the approximation is. The most popular, available in MATLAB library are:

Wavelet Family Name	
Haar Wavelet	Biorthogonal Wavelets
Daubechies Wavelets	Reverse Biorthogonal Wavelets
Symlets	Meyer Wavelet
Coiflets	Gaussian Wavelets
Mexican Hat Wavelet	Morlet Wavelet
Complex Gaussian Wavelet	Shannon Wavelets
Frequency B-Spline Wavelets	Complex Morlet Wavelets

To make a comparison in between an analytical formulation and the embedded functions, the Morlet and Mexican Hat Wavelets are shown below.





$$\psi_{morlet} = \pi^{-\frac{1}{4}} e^{-\frac{t^2}{2}} \cos(\omega t)$$

$$\psi_{mexican} = \pi^{-\frac{1}{4}} e^{-\frac{t^2}{2}} (1 - t^2)$$

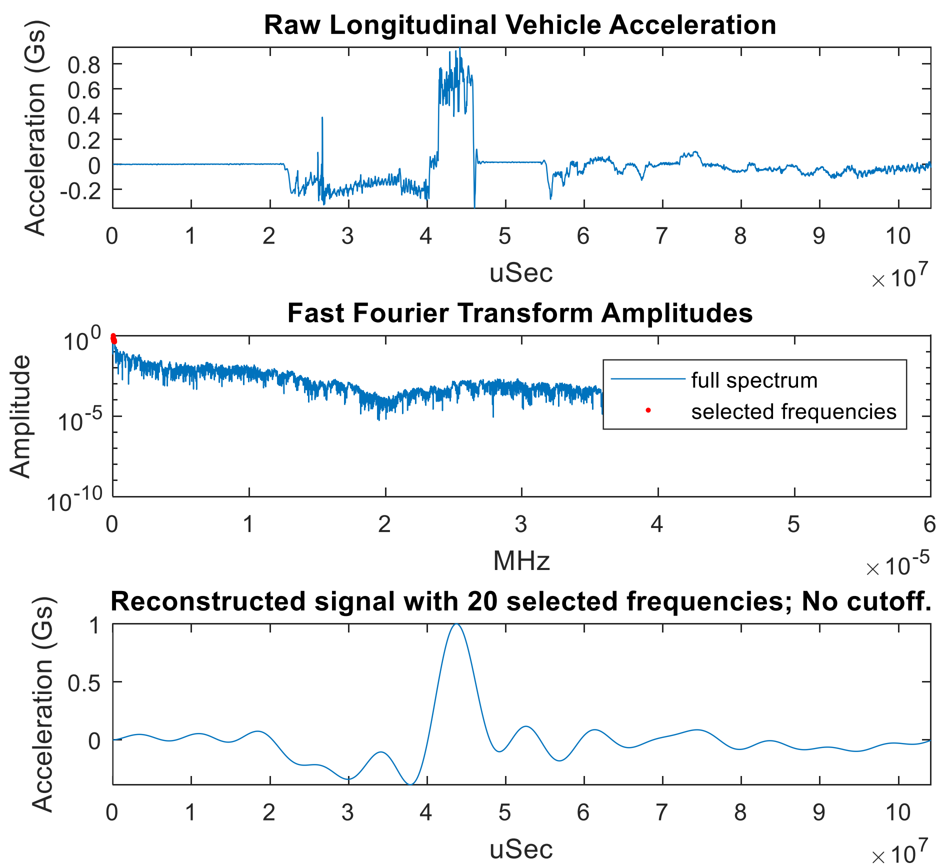
Applications

Like Fourier Series, Wavelets are used to decompose signals into components for applications such as filtering, machine vision, fingerprint compression, musical tones, and image recognition. For this report, the application of filtering was selected for acceleration analysis.

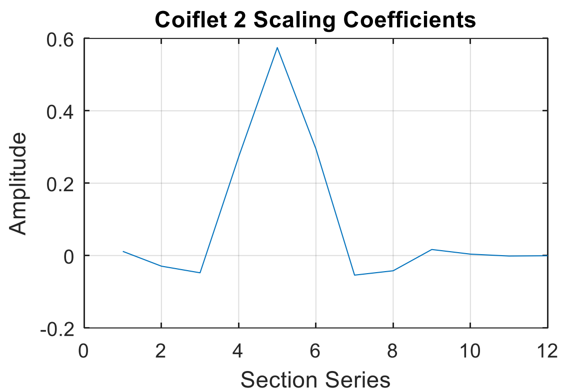
The purpose of the analysis is to explore the acceleration behavior of a vehicle under braking scenarios while traversing split surfaces of contact. The coefficient of friction for the road has a crucial influence in the level of braking a car can perform. Thus, split surface scenarios contribute to different levels on braking on each wheel individually. These scenarios motivated an investigation into the acceleration profiles a vehicle demonstrates under split surfaces.

The acceleration data comes from a testing scenario performed at Midwest Roadside Safety Facility Testing Grounds. An accelerometer was placed approximately at the center of gravity of the vehicle, and the testing scenario consisted of a vehicle driving up to 45 mph, entering a split surface in which both right wheels of the vehicle are in gravel, and both left wheels are in concrete. Immediately after entering this split surface, the vehicle produces a full brake until stopped.

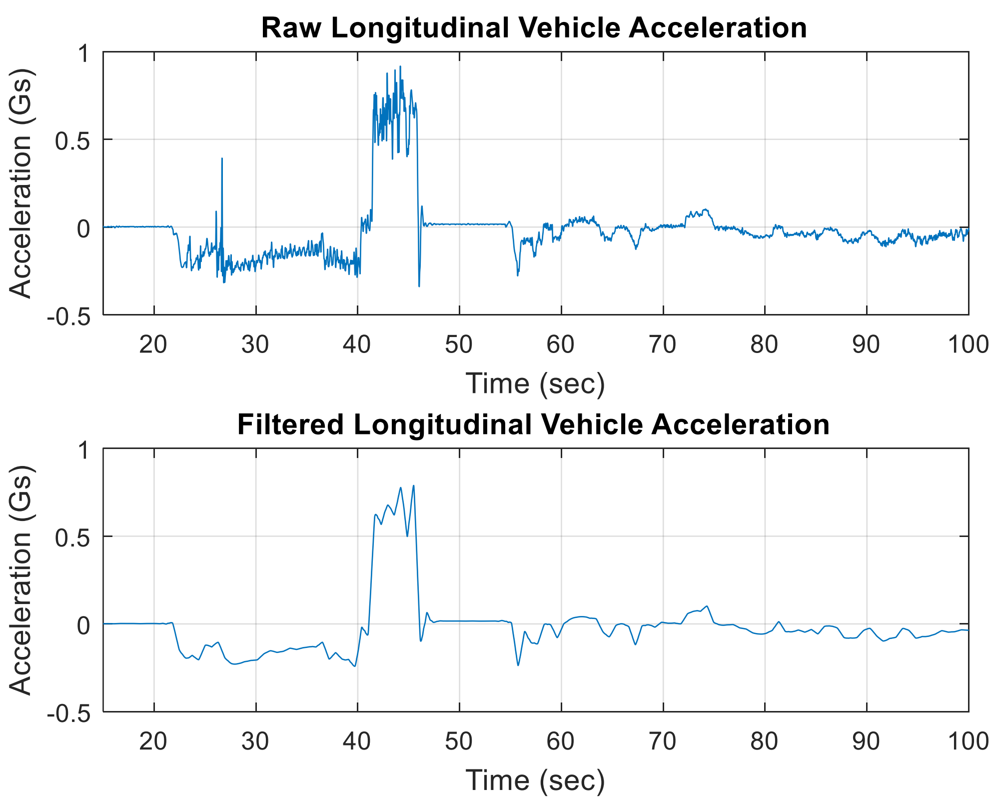
Ideal acceleration profiles for this behavior are in the form of linear functions, which are impossible to obtain through testing due to the noisy nature of the instrumentation. For this reason, filtering techniques such as Fast Fourier Transforms are needed. However, an FFT is shown below to compare the filtered signals from an FFT Analysis to a Wavelet Analysis [3].



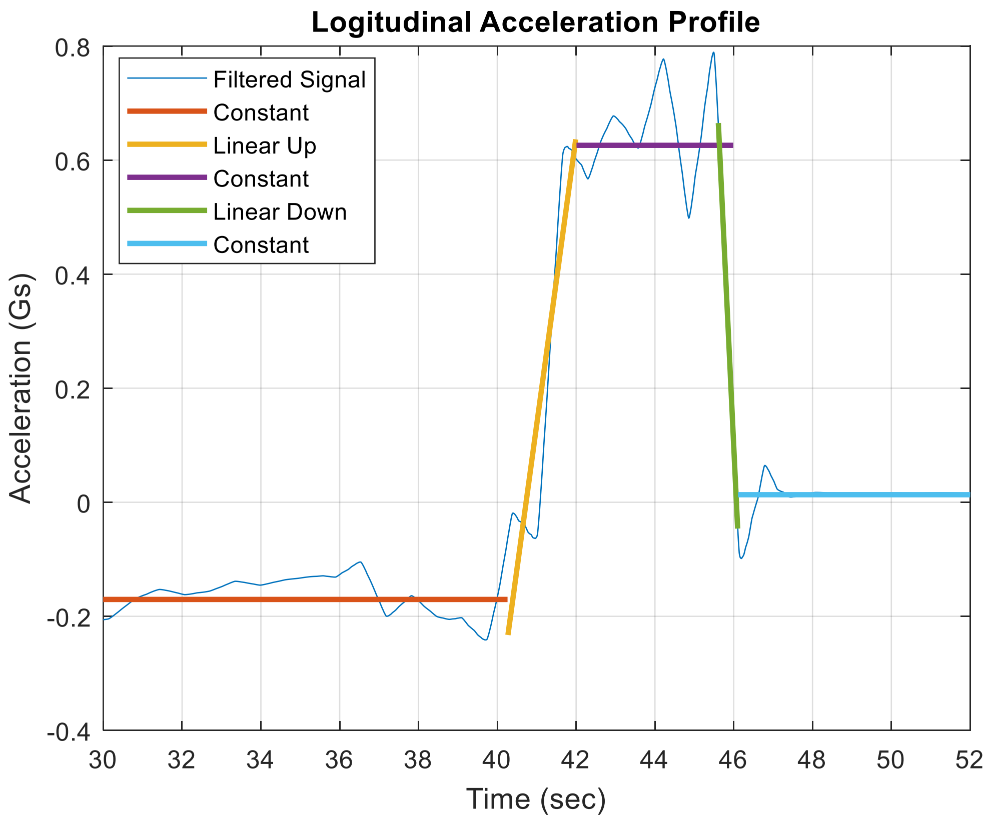
As it is noticed, the reconstructed signal by the FFT method, offers an oscillatory behavior that captures the “overall shape” of the curve, but does not capture the linear behavior of the actual signal as an ideal acceleration profile would offer. For the Wavelet comparison, the acceleration data was analyzed with the Wavelet Analyzer from MATLAB. The Wavelet Family selected was Coiflet 2 as shown below, with up to 6 levels of decomposition [A.1]. The choice of this wavelet was due to its “linear” shape that can capture linear slopes better than other Wavelets.



To obtain the appropriate DWT Coefficients, a method of cone of influence and spectrograms can be used that are provided in the Code Section. These methods offer a range of reasonable levels of decomposition for the signal to be analyzed. The resulting filtered signal from the Wavelet Analyzer shows a more linear behavior that is expected from an acceleration profile.



For the acceleration profile, curve fitting was implemented with the desired polynomials being of the degree that the ideal acceleration profile would normally have. In the case for the braking scenario, the ideal profile would be: Constant, Linearly Increase, Constant, Linearly Decrease, and Constant slopes. The results are shown in the figure below.



Conclusions

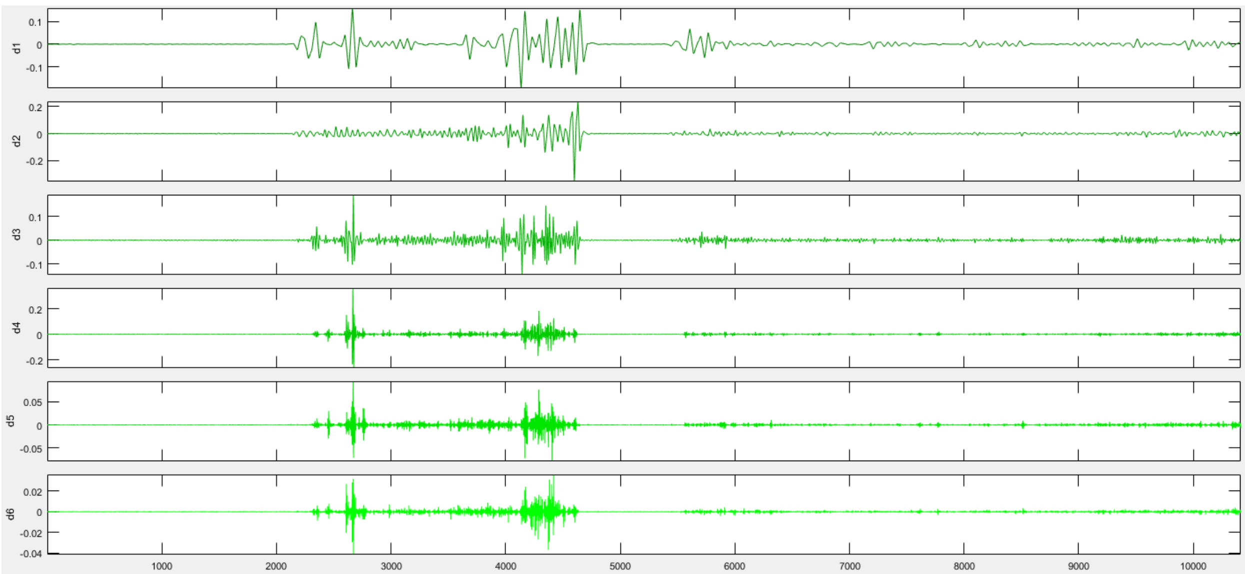
The Wavelet Analyzer tools proved to be a better alternative to filtering an acceleration signal due to the capacity of obtaining a linear basis function. Data fitting techniques can be used to approximate the ideal acceleration profile during braking. These fitting techniques such as Least Squares Method, are often limited by the noisy nature of the signal which is addressed through the Wavelet Transform Method.

References

1. Burrus, C. Sidney, Ramesh A. Gopinath, and Haitao Guo. *Introduction to Wavelets and Wavelet Transforms: A Primer*. 1 edition. Upper Saddle River, N.J: Pearson, 1997.
2. Mathworks – Wavelet Families (https://www.mathworks.com/help/wavelet/ug/wavelet-families-additional-discussion.html) Retrieved December 9, 2019.
3. Shmuel Ben-Ezra (2019). FFT filter (https://www.mathworks.com/matlabcentral/fileexchange/25017-fft-filter-clean-your-signals-and-display-results), MATLAB Central File Exchange. Retrieved December 9, 2019.

Appendix

A.1 - Coefficients of Decomposition Level



Decomposition Levels Extracted from MATLAB Wavelet 1-D Analyzer

Basic Code Utilized

```
%% Wavelet Coefficients
close all; clc; clear all
load('SGT.mat'); load('SGA.mat')
SGA = SpliGravelAccel; SGT =
SplitGravelTime;
ts = abs(SGT(1) - SGT(2));
fr = 1/ts;
plot(SGT,SGA); xlabel('sec');
ylabel('gs');
grid on; title('Data')
figure; [wt,f,coi] = cwt(SGA,fr);
cwt(SGA,fr)
figure; [wt,periods,coi] =
cwt(SGA,seconds(ts));
cwt(SGA,seconds(ts))
figure; cwt(SGA,'morse');
colormap jet; colorbar;
figure;
cwt(SGA,1:128,'haar','plot');
colormap jet; colorbar;


%% Sample Wavelets
clear all; close all; clc
[psi,xval] = wavefun('morl',10);
plot(xval,psi); title('Morlet
Wavelet');
hold on;
x = -8:.01:8;
y = pi^(-1/4)*exp(-
x.^2/2).*cos(5*x);
plot(x,y); grid on;
legend('MATLAB Morlet
Wavelet','Analytical Wavelet')
xlim([-4 4])

figure
lb = -5; ub = 5; N = 1000;
[psi,xval] = mexihat(lb,ub,N);
plot(xval,psi); hold on
title('Mexican Hat Wavelet');
xm = -5:.01:5;
ym = pi^(-1/4)*exp(-xm.^2/2).*(1-
xm.^2);
plot(xm,ym); grid on;
legend('MATLAB Mexican Hat
Wavelet','Analytical Wavelet')
xlim([-5 5])

clc; close all; clear all
wname = 'coif2';
f = coifwavf(wname);
plot(f); grid on
title('Coiflet 2 Scaling
Coefficients')
ylabel('Amplitude');
xlabel('Section Series')

%%Coilet Analysis

load('ApproxCoif2L6.mat');
load('SGT.mat');
Time = SplitGravelTime;
figure; subplot(211)
Signal = ApproxCoif2L6(1,:)*-1;
Approx = ApproxCoif2L6(6,:)*-1;
plot(Time(1:numel(Time)/2),Signal(1
:numel(Signal)/2)); hold on
xlabel('Time (sec)');
ylabel('Acceleration (Gs)'); grid
on
title('Raw Longitudinal Vehicle
Acceleration')
xlim([15 100])
subplot(212)
plot(Time(1:numel(Time)/2),Approx(1
:numel(Signal)/2))
xlabel('Time (sec)');
ylabel('Acceleration (Gs)'); grid
on;
title('Filtered Longitudinal
Vehicle Acceleration')
ylim([-.5 1]); xlim([15 100])

%% FFT Trial from Mathworks Code
clear all; clc; close all
load('SGAHalf.mat');
load('SGT.mat');
Time =
SplitGravelTime(1:numel(SplitGravel
Time)/2);
SGAHalf = SGAHalf*-1;
fftf(Time, SGAHalf, [], 20);
```

```
%% Fitting Code

clear all;clc; close all
load('ApproxCoif2L6.mat');
load('SGT.mat');
Time = SplitGravelTime;
Time = Time(1:numel(Time)/2);
Signal =  ApproxCoif2L6(6,:)*-1;
Signal = Signal(1:numel(Signal)/2);
plot(Time,Signal); hold on
Constant = [22.57,40.26];
Linear = [40.26, 41.52];
Constant2 = [41.52, 45.62];
LinearDown = [45.62 47.85];
Stopped = [47.85 55.06];

idx1 =find(Time > 22.57 & Time
<40.26);
idx2 =find(Time > 40.26 & Time
<42);
idx3 =find(Time > 42 & Time <46);
idx4 =find(Time > 45.6 & Time
<46.1);
idx5 =find(Time > 46.1 & Time
<55.06);
T1 = Time(idx1); S1 = Signal(idx1);
T2 = Time(idx2); S2 = Signal(idx2);
T3 = Time(idx3); S3 = Signal(idx3);
T4 = Time(idx4); S4 = Signal(idx4);
T5 = Time(idx5); S5 = Signal(idx5);


[p, S] = polyfit(T1,S1',0);
[s1_fit] = polyval(p,T1,S);
plot(T1,s1_fit,'linewidth',2); hold
on;

[p, S] = polyfit(T2,S2',1);
[s2_fit] = polyval(p,T2,S);
plot(T2,s2_fit,'linewidth',2); hold
on;

[p, S] = polyfit(T3,S3',0);
[s3_fit] = polyval(p,T3,S);
plot(T3,s3_fit,'linewidth',2); hold
on;

[p, S] = polyfit(T4,S4',1);
[s4_fit] = polyval(p,T4,S);
plot(T4,s4_fit,'linewidth',2); hold
on;

[p, S] = polyfit(T5,S5',0);
[s5_fit] = polyval(p,T5,S);
plot(T5,s5_fit,'linewidth',2); hold
on;

xlim([30 52]); grid on;
xlabel('Time (sec)');
ylabel('Acceleration (Gs)');

legend('Filtered
Signal','Constant','Linear Up',...
'Constant','Linear
Down','Constant','location','best')
;
title('Logitudinal Acceleration
Profile');
```