Appendix A
m - files

iwshista.m

```
%This m-file determines the aggregate amplitude distribution for the wheel load.  The file
%determines the histogram for each data file and averages the results to form the
%aggregate distribution.

clear

%Load list of file names for collected Wheel Load Data.

load filest

%Load matrix which correlates each wheel load its channel number in each data file.

load channelt

%Initialize specified parameters.

clear ev1 el1 hv hl
k = 0;

%Loop through each data file.

for z = 1:12

%Load time history data file determined by 'z'.
%The time history data was created by subdividing the reduced complete loop data into
%each of the five significant sections.

 load([files(z,9:11),'time.mat'])

%Selects the range and bin width used for the histogram routine.  As noted, xl is for the
%lateral loads and xv is for the vertical loads.

xl = linspace(-20,30,100);
xv = linspace(0,80,80);
v = 0;
l = 0;

%Determines the histogram for the lateral and vertical wheel load in each of the five
%sections.  Hv1 is the histogram data for the vertical loads and lv1 is the histogram data
%for the lateral loads.

for i = 1:40;

  j = i - floor((i-.05)/8)*8;
```

```
  if ((j-.5)/4)<1
   v = v +1;
   hv1(v,:) = hist(t(1:2002,i),xv);
  else
   l = l +1;
   hl1(l,:) = hist(t(1:2002,i),xl);
  end;

end;

%Averages the histogram data to form an aggregate distribution.

if z == 1
 hv = hv1;
 hl = hl1;
else
 hv = hv1 + hv;
 hl = hl1 + hl;
end;

%Uses the aggregate histogram data to from aggregate exceedence data.

ev = cumsum(hv');
el = cumsum(hl');
ev = ev';
el = el';

end;

%Once the analysis is completed, the data is save to a file in the mat format.

save hist hv hl xv xl ev el
```

iwspsd.m

```
% This m-file determines the frequency spectra of the wheel load data for each section.
%Each data file is loaded, the PSD is calculated and then saved to another file for future
%use.

%Loads a list of the wheel load file names.

load files

%Loads a matrix relating a wheel load to the correct data channel for each file.

load channel

%Loads a series of variables containing the starting and ending distance for each section
%of interest.  These variable consist of an S for start or F for finish and the appropriate
%section number.

load sections

%Loops through the processes for each of the data files.

for z = 1:length(files(:,1))

%Loads the wheel load data files specified by z.

  load(files(z,:))

% Finds PSD for Section 3 for each wheel load

for I = 1:length(channel(1,:)),

% Subdivides the data into the appropriate section and force.

  x = data(s3:f3,channel(z,I))-mean(data(s3:f3, channel(z,I )));

% Uses the Matlab PSD command with a 512 point hanning window.

  [p,freq] = psd(x,512,128,512);
  p3(1:length(p),I) = p;

end;

% Find PSD for Section 7 for every Column of Data

for I = 1: length(channel(1,:)),
```

```
  x = data(s7:f7, channel(z,I))-mean(data(s7:f7, channel(z,I)));
  [p,freq] = psd(x,512,128,512);
  p7(1:length(p),I) = p;

end;

% Find PSD for Section 25 for every Column of Data

for I = 1: length(channel(1,:)),

  x = data(s25:f25, channel(z,I))-mean(data(s25:f25, channel(z,I)));
  [p,freq] = psd(x,512,128,512);
  p25(1:length(p),I) = p;

end;

% Find PSD for Section 29 for every Column of Data

for I = 1: length(channel(1,:)),

  x = data(s29:f29, channel(z,I))-mean(data(s29:f29, channel(z,I)));
  [p,freq] = psd(x,512,128,512);
  p29(1:length(p),I) = p;

end;

% Find PSD for Section 33 for every Column of Data

for I = 1: length(channel(1,:)),

  x = data(s33:f33, channel(z,I))-mean(data(s33:f33, channel(z,I)));
  [p,freq] = psd(x,512,128,512);
  p33(1:length(p),I) = p;

end;

% Create Plots of the PSD for each Section and each Force.

figure(1)
 plot(freq,p3(:,1),'g')
 hold on
 axis([0 30 0 2000])
 plot(freq,p3(:,2),'w')
 plot(freq,p3(:,3),'c-.')
 plot(freq,p3(:,4),'r--')
```

```
 legend('VA11','VB11','VA12','VB12')
 xlabel('Frequency, Hz')
 title(['PSDs for Vertical Loads, Section 3 (5 Degree Curve),'  files(z,(length(files(1,:))-
9):(length(files(1,:))-7)) ' MGT'])
 grid on; hold off
print -dcdjcolor; pause(180)

figure(2)
 plot(freq,p3(:,5),'g')
 hold on
 axis([0 15 0 2000])
 plot(freq,p3(:,6),'w')
 plot(freq,p3(:,7),'c-.')
 plot(freq,p3(:,8),'r--')
 legend('LA11','LB11','LA12','LB12')
 xlabel('Frequency, Hz')
 title(['PSDs for Lateral Loads, Section 3 (5 Degree Curve), ' files(z,(length(files(1,:))-
9):(length(files(1,:))-7)) ' MGT'])
 grid on; hold off
print -dcdjcolor; pause(180)

figure(3)
 plot(freq,p7(:,1),'g')
 hold on
 axis([0 30 0 2000])
 plot(freq,p7(:,2),'w')
 plot(freq,p7(:,3),'c-.')
 plot(freq,p7(:,4),'r--')
 legend('VA11','VB11','VA12','VB12')
 xlabel('Frequency, Hz')
 title(['PSDs for Vertical Loads, Section 7 (5 Degree Rev. Curve),'
files(z,(length(files(1,:))-9):(length(files(1,:))-7)) ' MGT'])
 grid on; hold off
print -dcdjcolor; pause(180)

figure(4)
 plot(freq,p7(:,5),'g')
 hold on
 axis([0 15 0 2000])
 plot(freq,p7(:,6),'w')
 plot(freq,p7(:,7),'c-.')
plot(freq,p7(:,8),'r--')
 legend('LA11','LB11','LA12','LB12')
 xlabel('Frequency, Hz')
 title(['PSDs for Lateral Loads, Section 7 (5 Degree Rev. Curve), '
files(z,(length(files(1,:))-9):(length(files(1,:))-7)) ' MGT'])
```

```
 grid on; hold off
print -dcdjcolor; pause(180)

figure(5)
 plot(freq,p25(:,1),'g')
 hold on
 axis([0 30 0 2000])
 plot(freq,p25(:,2),'w')
 plot(freq,p25(:,3),'c-.')
 plot(freq,p25(:,4),'r--')
 legend('VA11','VB11','VA12','VB12')
 xlabel('Frequency, Hz')
 title(['PSDs for Vertical Loads, Section 25 (6 Degree Curve), ' files(z,(length(files(1,:))-
9):(length(files(1,:))-7)) ' MGT'])
 grid on; hold off
print -dcdjcolor; pause(180)

figure(6)
 plot(freq,p25(:,5),'g')
 hold on
 axis([0 15 0 2000])
 plot(freq,p25(:,6),'w')
 plot(freq,p25(:,7),'c-.')
 plot(freq,p25(:,8),'r--')
 legend('LA11','LB11','LA12','LB12')
 xlabel('Frequency, Hz')
 title(['PSDs for Lateral Loads, Section 25 (6 Degree Curve), ' files(z,(length(files(1,:))-
9):(length(files(1,:))-7)) ' MGT'])
 grid on; hold off
print -dcdjcolor; pause(180)

figure(7)
 plot(freq,p29(:,1),'g')
 hold on
 axis([0 30 0 2000])
 plot(freq,p29(:,2),'w')
 plot(freq,p29(:,3),'c-.')
 plot(freq,p29(:,4),'r--')
 legend('VA11','VB11','VA12','VB12')
 xlabel('Frequency, Hz')
 title(['PSDs for Vertical Loads, Section 29 (Tangent, Soft Subgrade), '
files(z,(length(files(1,:))-9):(length(files(1,:))-7)) ' MGT'])
 grid on; hold off
print -dcdjcolor; pause(180)

figure(8)
```

```
 plot(freq,p29(:,5),'g')
 hold on
 axis([0 15 0 2000])
 plot(freq,p29(:,6),'w')
 plot(freq,p29(:,7),'c-.')
 plot(freq,p29(:,8),'r--')
 legend('LA11','LB11','LA12','LB12')
 xlabel('Frequency, Hz')
 title(['PSDs for Lateral Loads, Section 29 (Tangent, Soft Subgrade), '
files(z,(length(files(1,:))-9):(length(files(1,:))-7)) ' MGT'])
 grid on; hold off
print -dcdjcolor; pause(180)

figure(9)
 plot(freq,p33(:,1),'g')
 hold on
 axis([0 30 0 2000])
 plot(freq,p33(:,2),'w')
 plot(freq,p33(:,3),'c-.')
 plot(freq,p33(:,4),'r--')
 legend('VA11','VB11','VA12','VB12')
 xlabel('Frequency, Hz')
 title(['PSDs for Vertical Loads, Section 33 (Tangent), ' files(z,(length(files(1,:))-
9):(length(files(1,:))-7)) ' MGT'])
 grid on; hold off
print -dcdjcolor; pause(180)

figure(10)
 plot(freq,p33(:,5),'g')
 hold on
 axis([0 15 0 2000])
 plot(freq,p33(:,6),'w')
 plot(freq,p33(:,7),'c-.')
 plot(freq,p33(:,8),'r--')
 legend('LA11','LB11','LA12','LB12')
 xlabel('Frequency, Hz')
 title(['PSDs for Lateral Loads, Section 33 (Tangent), ' files(z,(length(files(1,:))-
9):(length(files(1,:))-7)) ' MGT'])
 grid on; hold off
print -dcdjcolor; pause(180)

 figure(11)
 plot(freq,p3(:,1),'g')
 hold on
 axis([0 30 0 2000])
 plot(freq,p7(:,1),'w')
```

```
 plot(freq,p25(:,1),'c-.')
 plot(freq,p29(:,1),'r--')
 plot(freq,p33(:,1),'b-')
 legend('Section 3', 'Section 7', 'Section 25', 'Section 29', 'Section 33')
 xlabel('Frequency, Hz')
 title(['PSDs for Vertical Loads VA11, ' files(z,(length(files(1,:))-9):(length(files(1,:))-7)) '
MGT'])
 grid on; hold off
print -dcdjcolor; pause(180)

figure(12)
 plot(freq,p3(:,5),'g')
 hold on
 axis([0 30 0 2000])
 plot(freq,p7(:,5),'w')
 plot(freq,p25(:,5),'c-.')
 plot(freq,p29(:,5),'r--')
 plot(freq,p33(:,5),'b-')
 legend('Section 3', 'Section 7', 'Section 25', 'Section 29', 'Section 33')
 xlabel('Frequency, Hz')
 title(['PSDs for Lateral Loads LA11,  ' files(z,(length(files(1,:))-9):(length(files(1,:))-7)) '
MGT'])
 grid on; hold off
print -dcdjcolor; pause(180)

figure(13)
 plot(freq,p3(:,2),'g')
 hold on
 axis([0 30 0 2000])
 plot(freq,p7(:,2),'w')
 plot(freq,p25(:,2),'c-.')
 plot(freq,p29(:,2),'r--')
 plot(freq,p33(:,2),'b-')
 legend('Section 3', 'Section 7', 'Section 25', 'Section 29', 'Section 33')
 xlabel('Frequency, Hz')
 title(['PSDs for Vertical Loads VB11, ' files(z,(length(files(1,:))-9):(length(files(1,:))-7)) '
MGT'])
 grid on; hold off
print -dcdjcolor; pause(180)

figure(14)
 plot(freq,p3(:,6),'g')
 hold on
 axis([0 30 0 2000])
 plot(freq,p7(:,6),'w')
 plot(freq,p25(:,6),'c-.')
```

```
 plot(freq,p29(:,6),'r--')
 plot(freq,p33(:,6),'b-')
 legend('Section 3', 'Section 7', 'Section 25', 'Section 29', 'Section 33')
 xlabel('Frequency, Hz')
 title(['PSDs for Lateral Loads LB11,  ' files(z,(length(files(1,:))-9):(length(files(1,:))-7)) '
MGT'])
 grid on; hold off
print -dcdjcolor; pause(180)

figure(15)
 plot(freq,p3(:,3),'g')
 hold on
 axis([0 30 0 2000])
 plot(freq,p7(:,3),'w')
 plot(freq,p25(:,3),'c-.')
 plot(freq,p29(:,3),'r--')
 plot(freq,p33(:,3),'b-')
 legend('Section 3', 'Section 7', 'Section 25', 'Section 29', 'Section 33')
 xlabel('Frequency, Hz')
 title(['PSDs for Vertical Loads VA12, ' files(z,(length(files(1,:))-9):(length(files(1,:))-7)) '
MGT'])
 grid on; hold off
print -dcdjcolor; pause(180)

figure(16)
 plot(freq,p3(:,7),'g')
 hold on
 axis([0 30 0 2000])
 plot(freq,p7(:,7),'w')
 plot(freq,p25(:,7),'c-.')
 plot(freq,p29(:,7),'r--')
 plot(freq,p33(:,7),'b-')
 legend('Section 3', 'Section 7', 'Section 25', 'Section 29', 'Section 33')
 xlabel('Frequency, Hz')
 title(['PSDs for Lateral Loads LA12, ' files(z,(length(files(1,:))-9):(length(files(1,:))-7)) '
MGT'])
 grid on; hold off
print -dcdjcolor; pause(180)

figure(17)
 plot(freq,p3(:,4),'g')
 hold on
 axis([0 30 0 2000])
 plot(freq,p7(:,4),'w')
 plot(freq,p25(:,4),'c-.')
 plot(freq,p29(:,4),'r--')
```

```
 plot(freq,p33(:,4),'b-')
 legend('Section 3', 'Section 7', 'Section 25', 'Section 29', 'Section 33')
 xlabel('Frequency, Hz')
 title(['PSDs for Vertical Loads VB12, ' files(z,(length(files(1,:))-9):(length(files(1,:))-7)) '
MGT'])
 grid on; hold off
print -dcdjcolor; pause(180)

figure(18)
 plot(freq,p3(:,8),'g')
 hold on
 axis([0 30 0 2000])
 plot(freq,p7(:,8),'w')
 plot(freq,p25(:,8),'c-.')
 plot(freq,p29(:,8),'r--')
 plot(freq,p33(:,8),'b-')
 legend('Section 3', 'Section 7', 'Section 25', 'Section 29', 'Section 33')
 xlabel('Frequency, Hz')
 title(['PSDs for Vertical Loads LB12, ' files(z,(length(files(1,:))-9):(length(files(1,:))-7)) '
MGT'])
 grid on; hold off
print -dcdjcolor; pause(180)

%Saves the results from each data file into a file with a name consisting of the Traffic
%Level and 'PSD'
%Example file name: 138psd.mat

save([files(z,(length(files(1,:))-9):(length(files(1,:))-7)) 'psd'])

%Returns to the beginning of the loop until z is the same size as the number of data files.

end;
```

iwspsda.m

```
%This m-file determines the aggregate frequency spectra for the wheel load data in
%section 25.  The file loads the previously determined PSD's of the data and performs an
%average and standard deviation for each frequency.

clear

n = 0;

%This variable is a list of the traffic level relating to each recorded wheel load file.

b = ['138';'159';'161';'175';'195';'210';'212';'222';'257';'310';'326';'338';'376';'460';'513';
'522';'614';'628';];

%This loop steps through each file recorded for the standard NACO Wedgelock truck.

for I=1:14;

%Loads the PSD data for the specified traffic level.

 load(['f:\',b,'psd.mat'])

%Organizes all of the manipulated data into specific variables.  As can be seen all of the
%data for a specific wheel load will be arranged together in order of traffic level.

p25va11(:,I) = p25(:,1);
p25vb11(:,I) = p25(:,2);
p25va12(:,I) = p25(:,3);
p25vb12(:,I) = p25(:,4);

p25la11(:,I) = p25(:,5);
p25lb11(:,I) = p25(:,6);
p25la12(:,I) = p25(:,7);
p25lb12(:,I) = p25(:,8);

clear distance time p25 p25 p25 p29 p253;
end;

%Determines the means at each frequency for all of the recorded traffic levels.

p25va11m = mean(p25va11.');
p25vb11m = mean(p25vb11.');
p25va12m = mean(p25va12.');
p25vb12m = mean(p25vb12.');
```

```
p25la11m = mean(p25la11.');
p25lb11m = mean(p25lb11.');
p25la12m = mean(p25la12.');
p25lb12m = mean(p25lb12.');

%Determines the standard deviation at each frequency for all of the recorded traffic levels.

p25va11s = std(p25va11.');
p25vb11s = std(p25vb11.');
p25va12s = std(p25va12.');
p25vb12s = std(p25vb12.');

p25la11s = std(p25la11.');
p25lb11s = std(p25lb11.');
p25la12s = std(p25la12.');
p25lb12s = std(p25lb12.');

clear p25va11 p25vb11 p25va12 p25vb12;
clear p25la11 p25lb11 p25la12 p25lb12;

%Assumes normality and determines the 95% confidence intervals accordingly.
%Remember the data for the different files has been averaged a number of times which
%should increase the normality of the data substantially.

p25va11cl = p25va11m - 1.96*p25va11s;
p25vb11cl = p25vb11m - 1.96*p25vb11s;
p25va12cl = p25va12m - 1.96*p25va12s;
p25vb12cl = p25vb12m - 1.96*p25vb12s;

p25la11cl = p25la11m - 1.96*p25la11s;
p25lb11cl = p25lb11m - 1.96*p25lb11s;
p25la12cl = p25la12m - 1.96*p25la12s;
p25lb12cl = p25lb12m - 1.96*p25lb12s;

p25va11cu = p25va11m + 1.96*p25va11s;
p25vb11cu = p25vb11m + 1.96*p25vb11s;
p25va12cu = p25va12m + 1.96*p25va12s;
p25vb12cu = p25vb12m + 1.96*p25vb12s;

p25la11cu = p25la11m + 1.96*p25la11s;
p25lb11cu = p25lb11m + 1.96*p25lb11s;
p25la12cu = p25la12m + 1.96*p25la12s;
p25lb12cu = p25lb12m + 1.96*p25lb12s;

clear data distance time curves
```

```
%Saves the aggregate PSD data for section 25 in mat format.

save psd25
```

dep.m

```
%This m-file determines the dependence of the standard deviation of each wheel load on
%the standard deviation of each track parameter

%Loads a file containing the mean and standard deviation of each wheel load and track
%geometry file for each section.

load tico

for i = 1:6;
for j = 1:8;

%Since the mean data showed little correlation, the standard deviation of each parameter
%was the focus.  The standard deviation of each of the parameters occupied only the even
%matrix indexes.

%The dependence was taken as the slope of the least squares fitted line to the data.  The
%Matlab polyfit command with a parameter of 1 was used for the least squares curve fit.

y = polyfit(tg3(:,2*i),iws3(:,2*j),1);
y3(i,j) = y(1);
y = polyfit(tg7(:,2*i),iws7(:,2*j),1);
y7(i,j) = y(1);
y = polyfit(tg25(:,2*i),iws25(:,2*j),1);
y25(i,j) = y(1);
y = polyfit(tg29(:,2*i),iws29(:,2*j),1);
y29(i,j) = y(1);
y = polyfit(tg33(:,2*i),iws33(:,2*j),1);
y33(i,j) = y(1);

end;
end;

%For comparison, the data was normalized so that the norm of each wheel loads
%dependence values were one.

for i = 1:8;

y3(:,i) = y3(:,i)/norm(y3(:,i));
y7(:,i) = y7(:,i)/norm(y7(:,i));
y25(:,i) = y25(:,i)/norm(y25(:,i));
y29(:,i) = y29(:,i)/norm(y29(:,i));
y33(:,i) = y33(:,i)/norm(y33(:,i));

end;
```

cor.m

```
%This m-file determines the correlation coefficient for the relationship between the
%standard deviation of the track geometry parameters and the standard deviation of the
%wheel loads.

%Loads a file containing the mean and standard deviation of each wheel load and track
%geometry file for each section.

load tico

%Loops through the mean and standard deviation values for each of the wheel loads and
%track parameters in each section.

for i = 1:12;
for j = 1:16;

%Calculate the correlation coefficient for each load and track parameter.

p = sum(tg3(:,i).*iws3(:,j))/(sqrt(sum(tg3(:,i).*tg3(:,i)))*sqrt(sum(iws3(:,j).*iws3(:,j))));
p3(i,j) = p(1,1);
p = sum(tg7(:,i).*iws7(:,j))/(sqrt(sum(tg7(:,i).*tg7(:,i)))*sqrt(sum(iws7(:,j).*iws7(:,j))));
p7(i,j) = p(1,1);
p =
sum(tg25(:,i).*iws25(:,j))/(sqrt(sum(tg25(:,i).*tg25(:,i)))*sqrt(sum(iws25(:,j).*iws25(:,j)))
);
p25(i,j) = p(1,1);
p =
sum(tg29(:,i).*iws29(:,j))/(sqrt(sum(tg29(:,i).*tg29(:,i)))*sqrt(sum(iws29(:,j).*iws29(:,j)))
);
p29(i,j) = p(1,1);
p =
sum(tg33(:,i).*iws33(:,j))/(sqrt(sum(tg33(:,i).*tg33(:,i)))*sqrt(sum(iws33(:,j).*iws33(:,j)))
);
p33(i,j) = p(1,1);

end;
end;
```