Contents

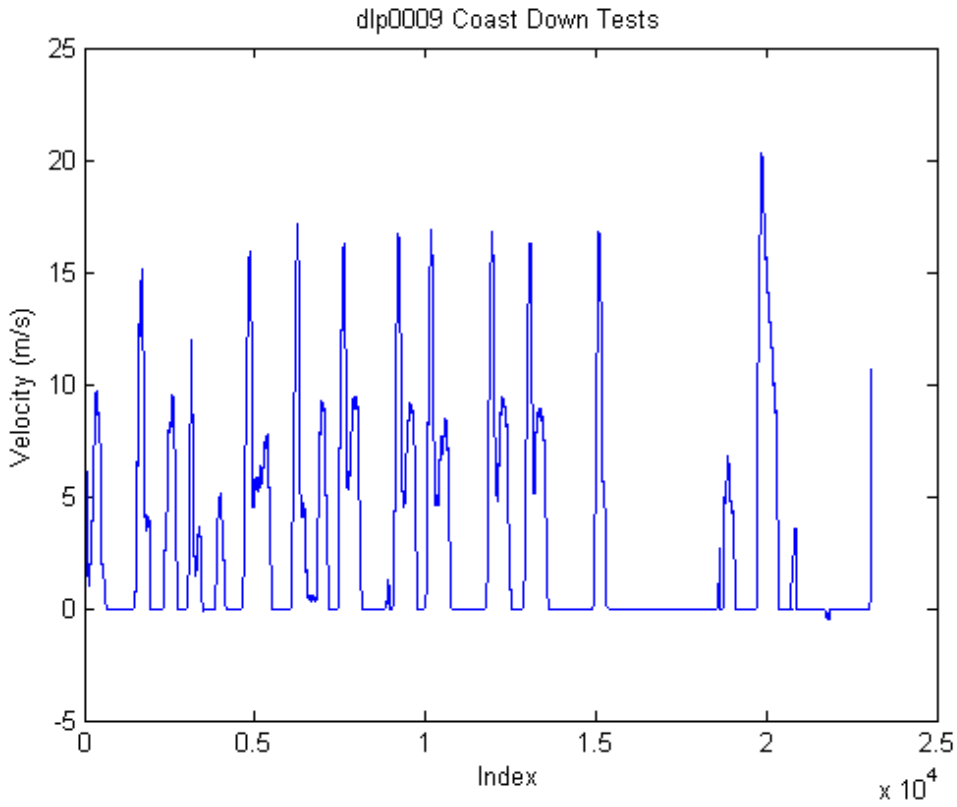

```
% Coast Down Data Analysis

close all
clear
clc

load dlp8-B.mat
rho=1.1839; % kg/m^3
N=319.193*9.8; % N
A=1.08; % m^2
```

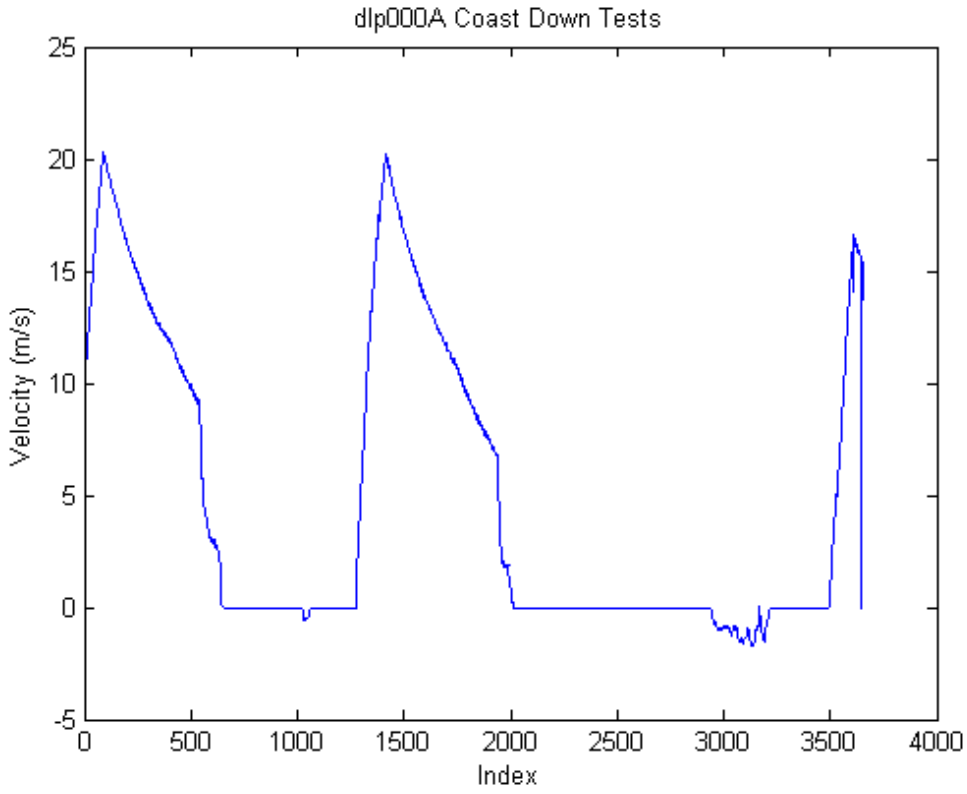
Finding velocity indices

```
close all

index9=0;
for k=1:size(dlp0009,2)
    if strcmp(dlp0009(k).desc,' Vehicle Velocity m/s Vehicle velocity in metres / second.')
        index9=k;
    end
end

indexA=0;
for k=1:size(dlp000A,2)
    if strcmp(dlp000A(k).desc,' Vehicle Velocity m/s Vehicle velocity in metres / second.')
        indexA=k;
    end
end
```

Plotting data

```
figure
plot(dlp0009(index9).values)
    title('dlp0009 Coast Down Tests')
    xlabel('Index')
    ylabel('Velocity (m/s)')

figure
plot(dlp000A(indexA).values)
    title('dlp000A Coast Down Tests')
    xlabel('Index')
    ylabel('Velocity (m/s)')
```

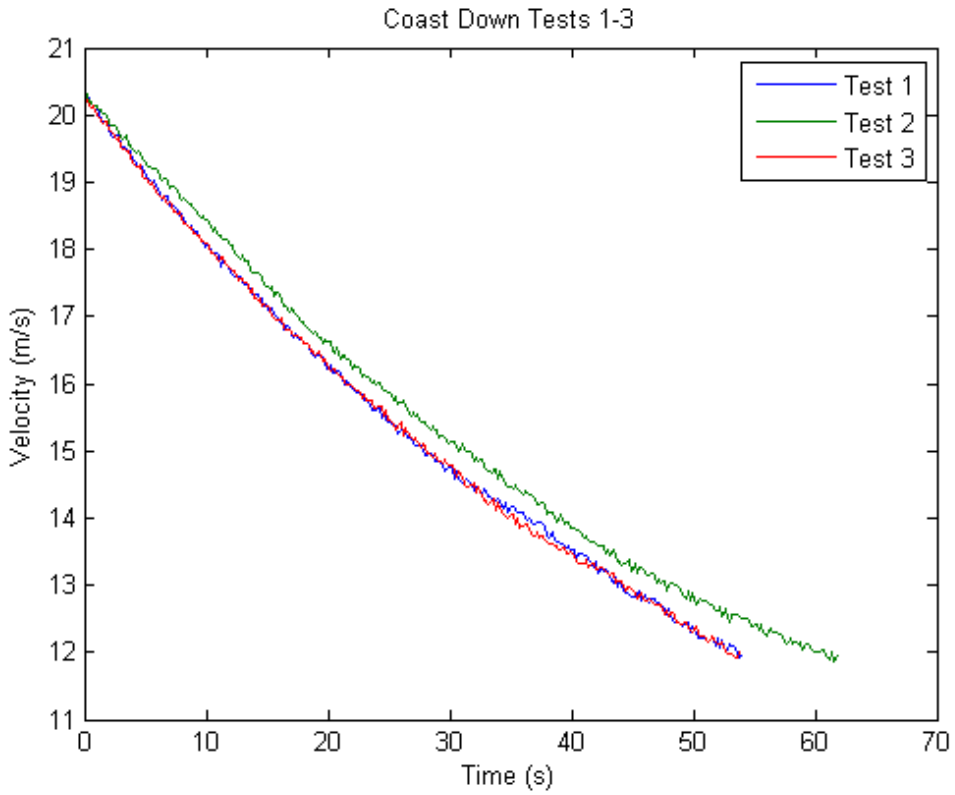




Isolating tests

```
close all

tin1=19870:20140;
tin2=89:398;
```

```
tin3=1415:1684;
figure
plot(dlp0009(index9).times(tin1)-dlp0009(index9).times(tin1(1)),...
    dlp0009(index9).values(tin1),...
    dlp000A(indexA).times(tin2)-dlp000A(indexA).times(tin2(1)),...
    dlp000A(indexA).values(tin2),...
    dlp000A(indexA).times(tin3)-dlp000A(indexA).times(tin3(1)),...
    dlp000A(indexA).values(tin3))
    title('Coast Down Tests 1-3')
    xlabel('Time (s)')
    ylabel('Velocity (m/s)')
    legend('Test 1','Test 2','Test 3')
```



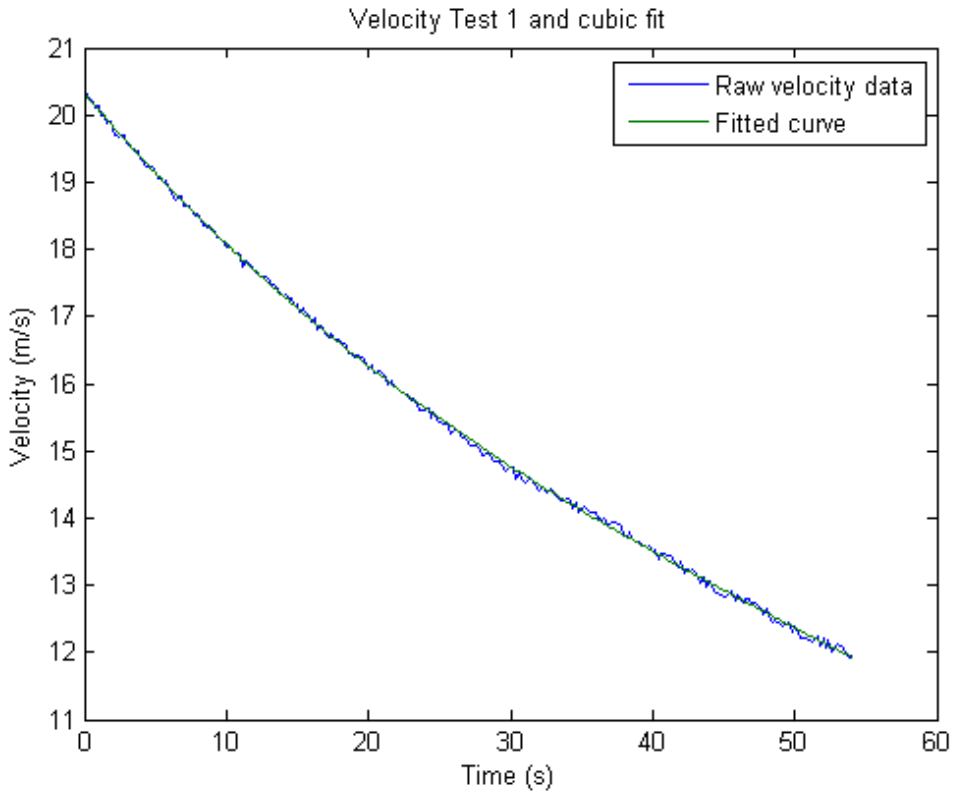
Creating time and velocity vectors

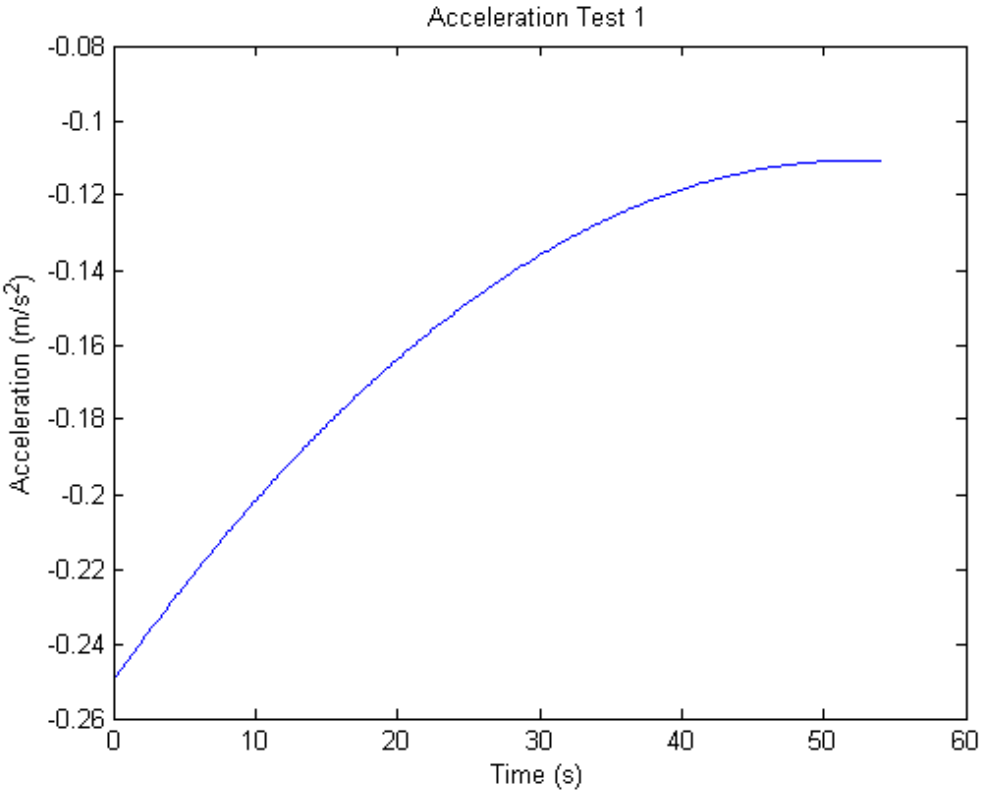
```
t1=dlp0009(index9).times(tin1)-dlp0009(index9).times(tin1(1));
v1=dlp0009(index9).values(tin1);

t2=dlp000A(indexA).times(tin2)-dlp000A(indexA).times(tin2(1));
v2=dlp000A(indexA).values(tin2);

t3=dlp000A(indexA).times(tin3)-dlp000A(indexA).times(tin3(1));
v3=dlp000A(indexA).values(tin3);
```

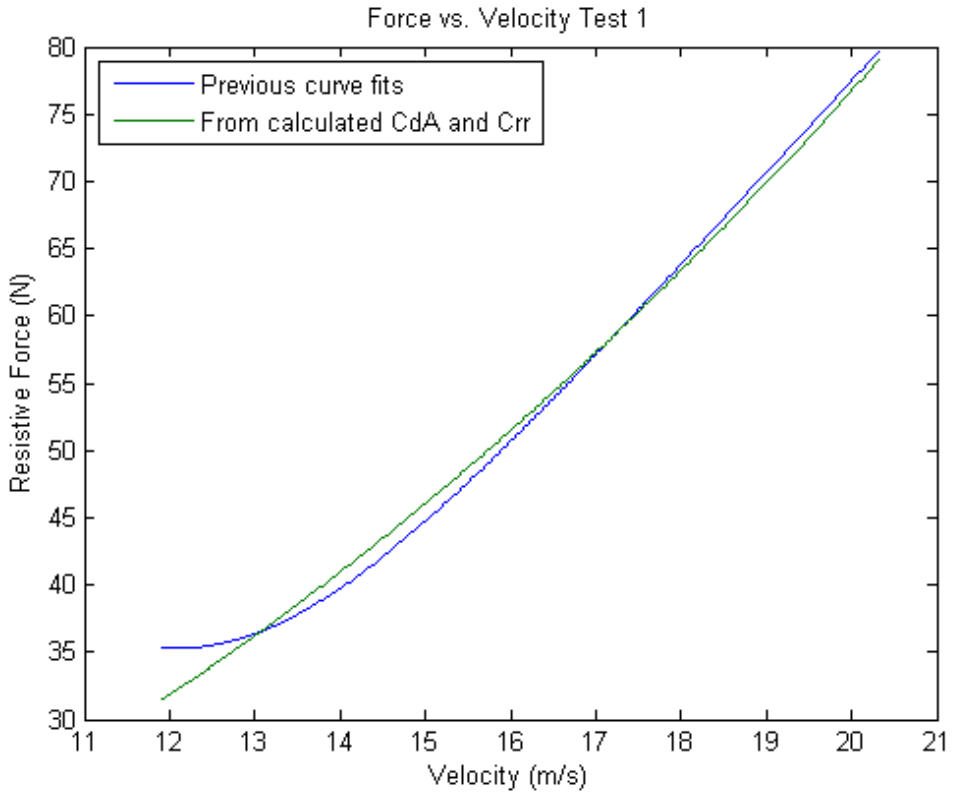
Test 1 Coefficient Calculations

```
% close all

% Cubic regression of velocity curve (coefficients stored in "v1fits")
v1fits=polyfit(t1,v1,3);
v1anon=@(t) v1fits(1).*t.^3+v1fits(2).*t.^2+v1fits(3).*t+v1fits(4);

% Plot to verify regression
figure
plot(t1,v1,t1,v1anon(t1))
```

```
    title('Velocity Test 1 and cubic fit')
    xlabel('Time (s)')
    ylabel('Velocity (m/s)')
    legend('Raw velocity data','Fitted curve')

% Differentiate v(t) curve to get acceleration
a1fits=[v1fits(1)*3 v1fits(2)*2 v1fits(3)];
a1anon=@(t) a1fits(1).*t.^2+a1fits(2).*t+a1fits(3);

% Plot acceleration to verify
figure
plot(t1,a1anon(t1))
    title('Acceleration Test 1')
    xlabel('Time (s)')
    ylabel('Acceleration (m/s^2)')

% F=ma
f1anon=@(t) -319.193.*a1anon(t);

sol1=[v1anon(t1).^2' ones(length(t1),1)]\f1anon(t1)';
CdA1=sol1(1)*2/rho;
Crr1=sol1(2)/N;

figure
plot(v1anon(t1),f1anon(t1),...
    v1anon(t1),0.5*rho*CdA1.*v1anon(t1).^2+Crr1*N)
    title('Force vs. Velocity Test 1')
    xlabel('Velocity (m/s)')
    ylabel('Resistive Force (N)')
    legend('Previous curve fits','From calculated CdA and Crr','Location','NorthWest')
```



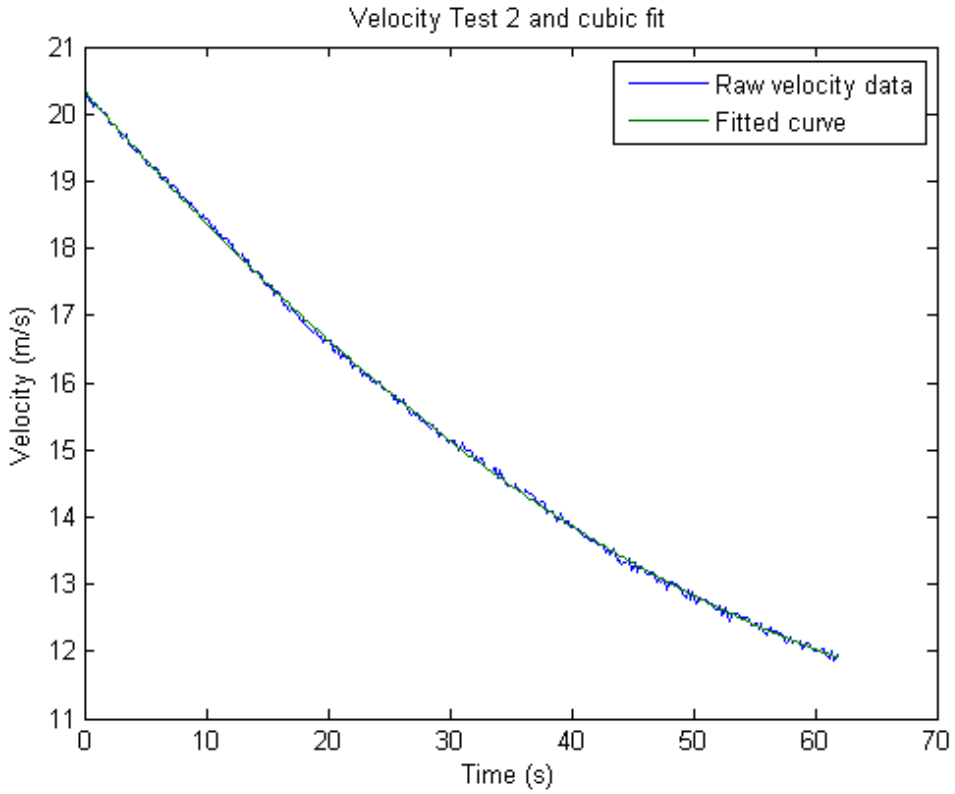


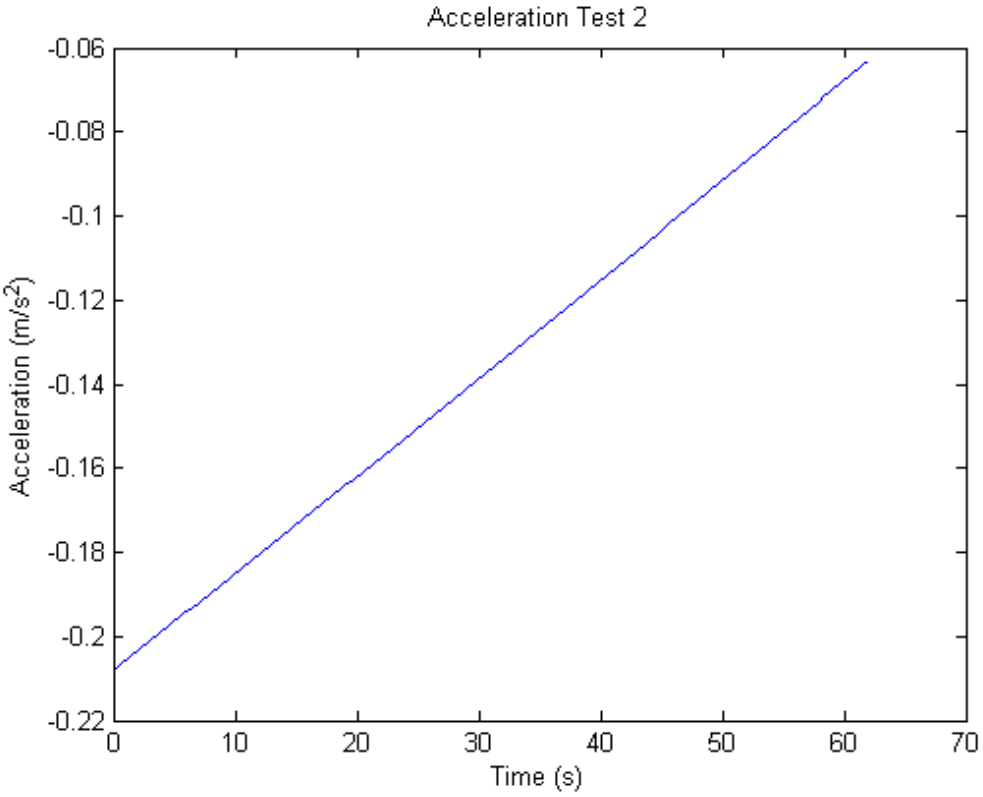


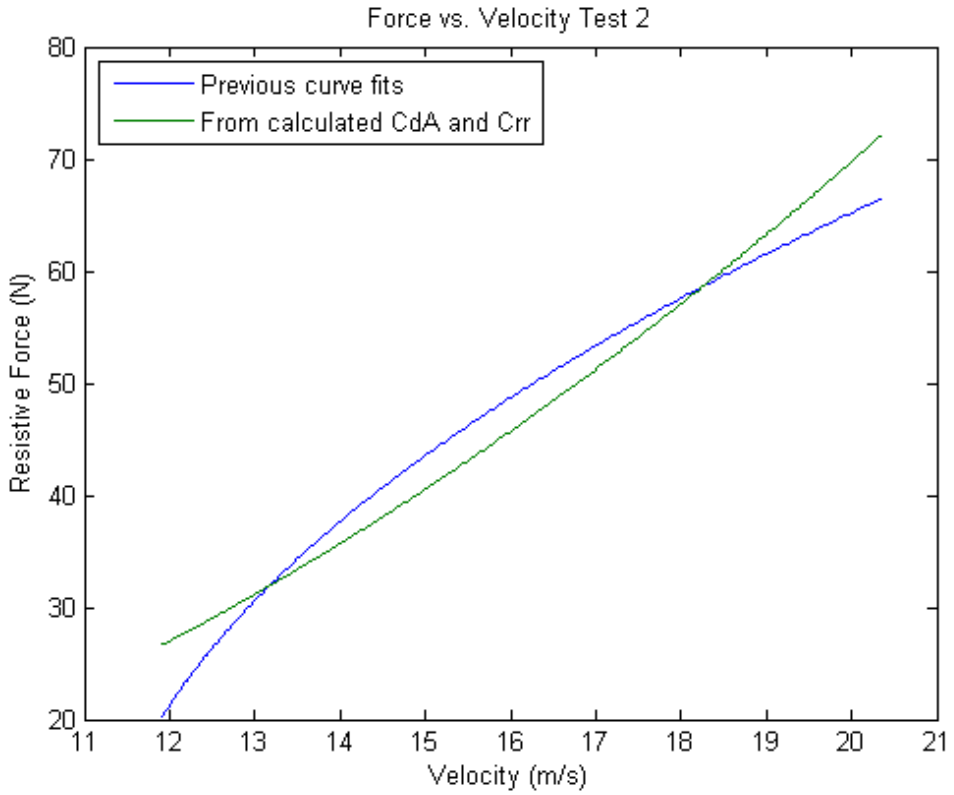
Test 2 Coefficient Calculations

```
% close all

% Cubic regression of velocity curve (coefficients stored in "v2fits")
v2fits=polyfit(t2,v2,3);
```

```
v2anon=@(t) v2fits(1).*t.^3+v2fits(2).*t.^2+v2fits(3).*t+v2fits(4);

% Plot to verify regression
figure
plot(t2,v2,t2,v2anon(t2))
    title('Velocity Test 2 and cubic fit')
    xlabel('Time (s)')
    ylabel('Velocity (m/s)')
    legend('Raw velocity data','Fitted curve')

% Differentiate v(t) curve to get acceleration
a2fits=[v2fits(1)*3 v2fits(2)*2 v2fits(3)];
a2anon=@(t) a2fits(1).*t.^2+a2fits(2).*t+a2fits(3);

% Plot acceleration to verify
figure
plot(t2,a2anon(t2))
    title('Acceleration Test 2')
    xlabel('Time (s)')
    ylabel('Acceleration (m/s^2)')

% F=ma
f2anon=@(t) -319.193.*a2anon(t);

sol2=[v2anon(t2).^2' ones(length(t2),1)]\f2anon(t2)';
CdA2=sol2(1)*2/rho;
Crr2=sol2(2)/N;

figure
plot(v2anon(t2),f2anon(t2),...
    v2anon(t2),0.5*rho*CdA2.*v2anon(t2).^2+Crr2*N)
    title('Force vs. Velocity Test 2')
    xlabel('Velocity (m/s)')
    ylabel('Resistive Force (N)')
    legend('Previous curve fits','From calculated CdA and Crr','Location','NorthWest')
```



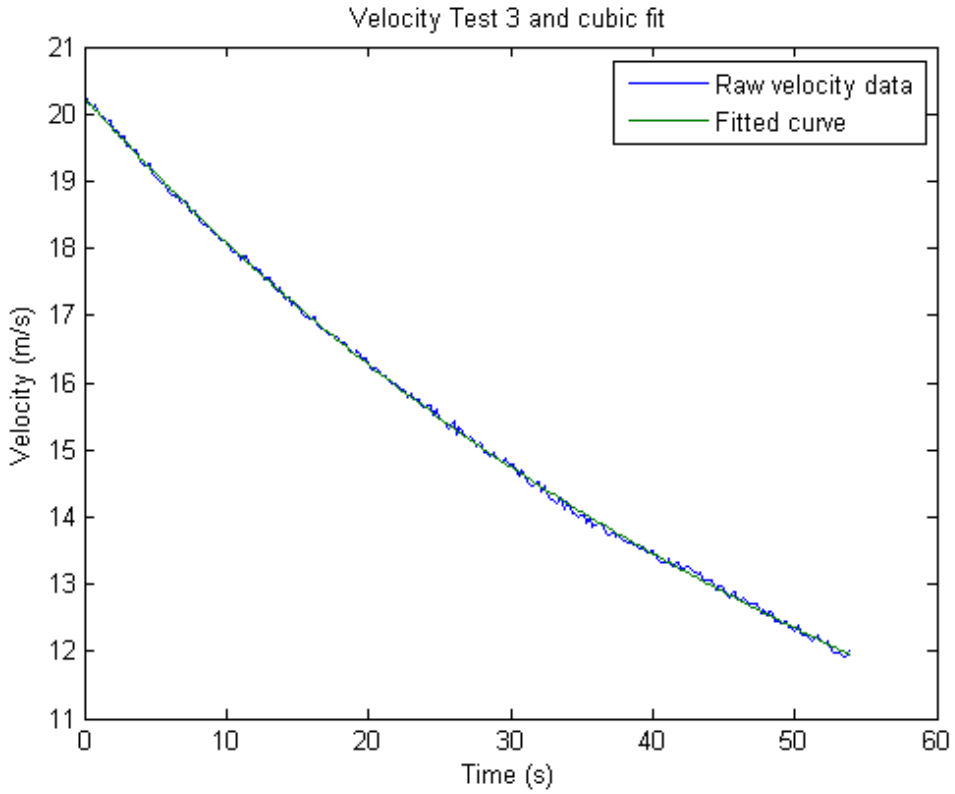


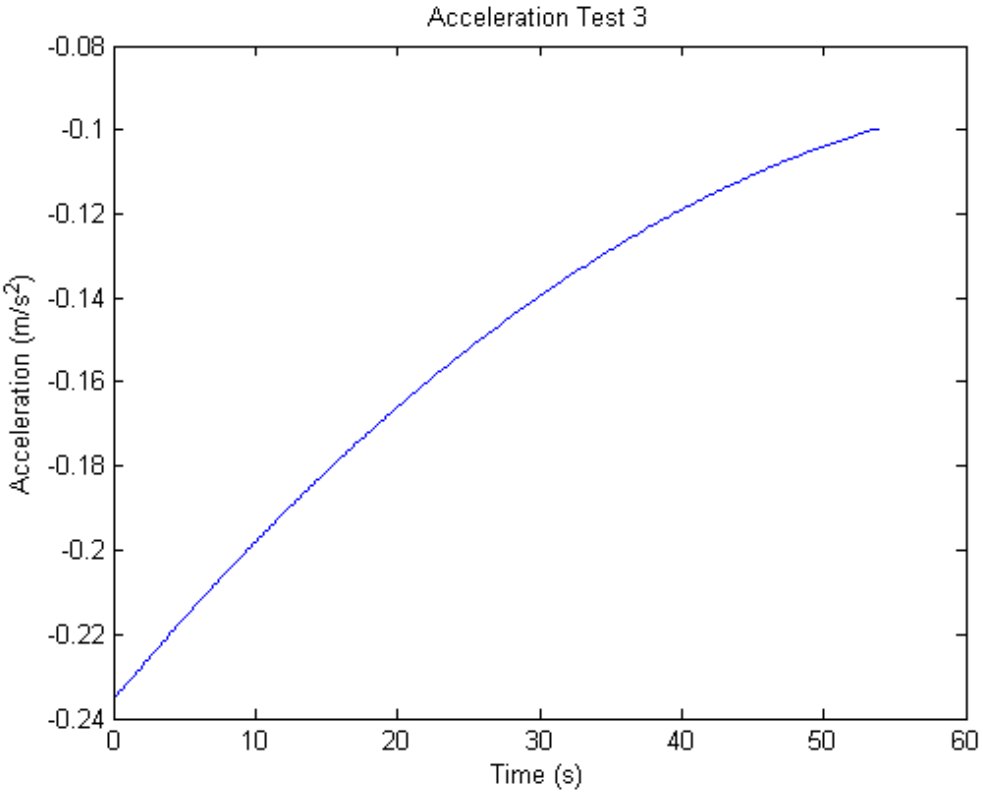


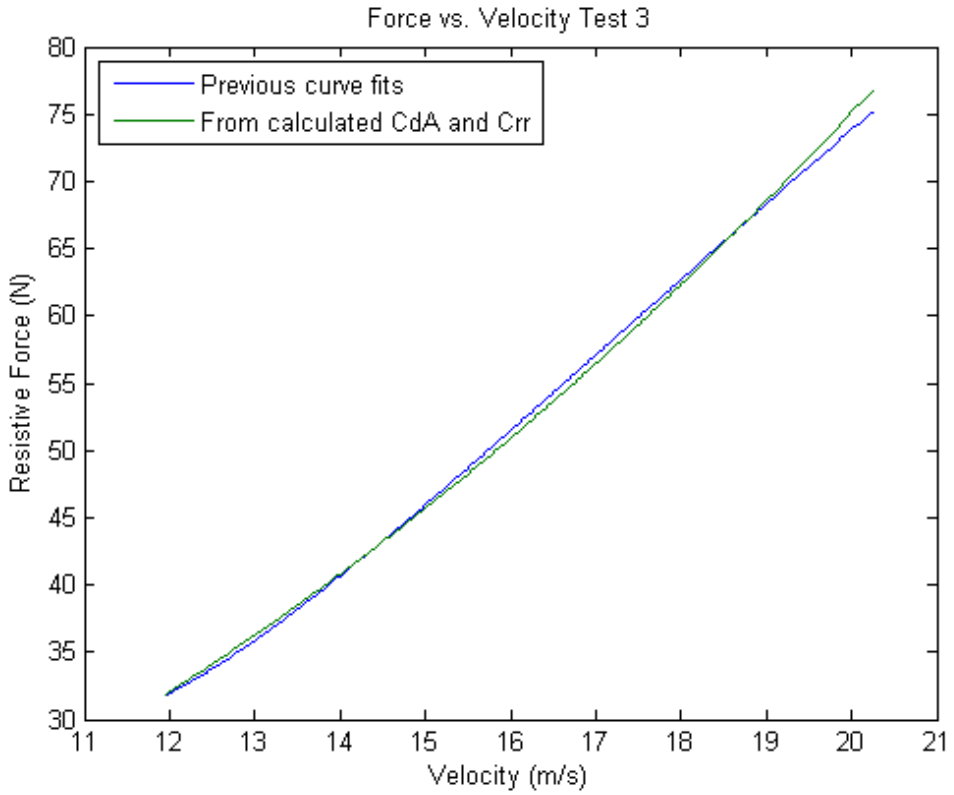
Test 3 Coefficient Calculations

```
% close all

% Cubic regression of velocity curve (coefficients stored in "v3fits")
v3fits=polyfit(t3,v3,3);
```

```
v3anon=@(t) v3fits(1).*t.^3+v3fits(2).*t.^2+v3fits(3).*t+v3fits(4);

% Plot to verify regression
figure
plot(t3,v3,t3,v3anon(t3))
    title('Velocity Test 3 and cubic fit')
    xlabel('Time (s)')
    ylabel('Velocity (m/s)')
    legend('Raw velocity data','Fitted curve')

% Differentiate v(t) curve to get acceleration
a3fits=[v3fits(1)*3 v3fits(2)*2 v3fits(3)];
a3anon=@(t) a3fits(1).*t.^2+a3fits(2).*t+a3fits(3);

% Plot acceleration to verify
figure
plot(t3,a3anon(t3))
    title('Acceleration Test 3')
    xlabel('Time (s)')
    ylabel('Acceleration (m/s^2)')

% F=ma
f3anon=@(t) -319.193.*a3anon(t);

sol3=[v3anon(t3).^2' ones(length(t3),1)]\f3anon(t3)';
CdA3=sol3(1)*2/rho;
Crr3=sol3(2)/N;

figure
plot(v3anon(t3),f3anon(t3),...
    v3anon(t3),0.5*rho*CdA3.*v3anon(t3).^2+Crr3*N)
    title('Force vs. Velocity Test 3')
    xlabel('Velocity (m/s)')
    ylabel('Resistive Force (N)')
    legend('Previous curve fits','From calculated CdA and Crr','Location','NorthWest')
```








```
CdA=[CdA1; CdA2; CdA3] % m^2
Cd=CdA/A % arb. unit
Crr=[Crr1; Crr2; Crr3] % arb. unit
close all
```

```
CdA =
    0.2963
    0.2821
    0.2842
Cd =
    0.2743
    0.2612
    0.2631
Crr =
    0.0021
    0.0010
    0.0025
```

Published with MATLAB® 7.10