Name: Akshat Hemang Jani

Roll No: 231010008

Sub: Measurement & Data Analysis

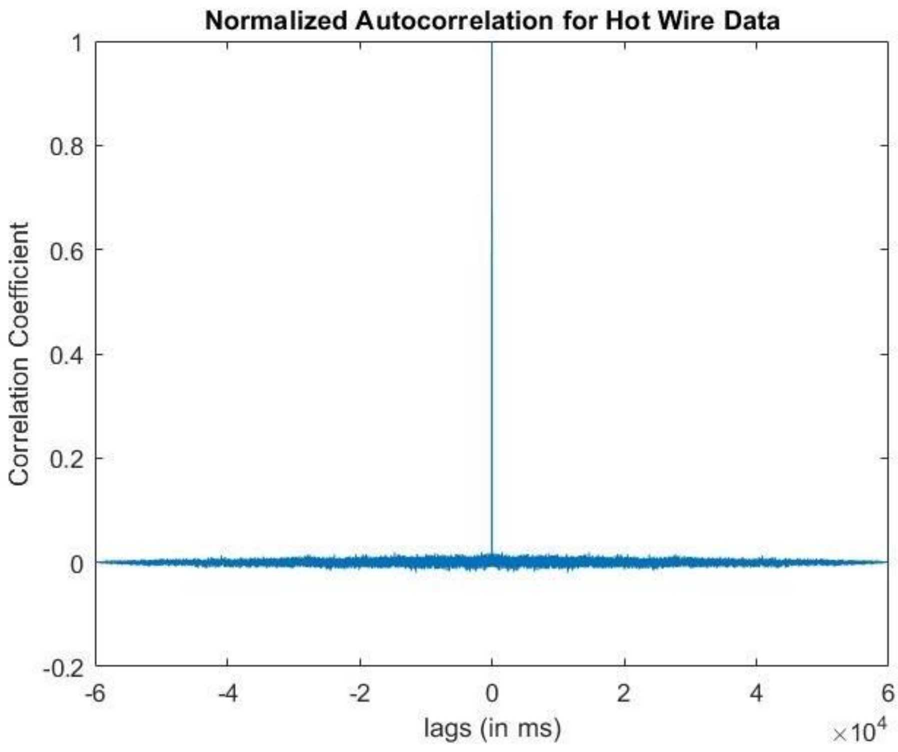
Assignment #2

Q1.

MATLAB Code

```
clc
clear all
format long
HW_Data = readmatrix('Hotwire.dat');
N = size(HW_Data, 1); %Total No. of Frames
T = round(max(HW_Data(:, 1))); %Total Time Period
f = N/T; %Sampling Frequency
[autocorr, lag] = xcorr(HW_Data(:, 2) - mean(HW_Data(:,2)),
'coeff');
lag_ms = lag/f *1000; %Lag in milliseconds
plot(lag_ms, autocorr)
title('Normalized Autocorrelation for Hot Wire Data')
xlabel('lags (in ms)')
ylabel('Correlation Coefficient')
TS = trapz(lag_ms, autocorr);
disp(TS);
```

The integral time scale is : 7.758658142620334e-07 ms



(Q2)

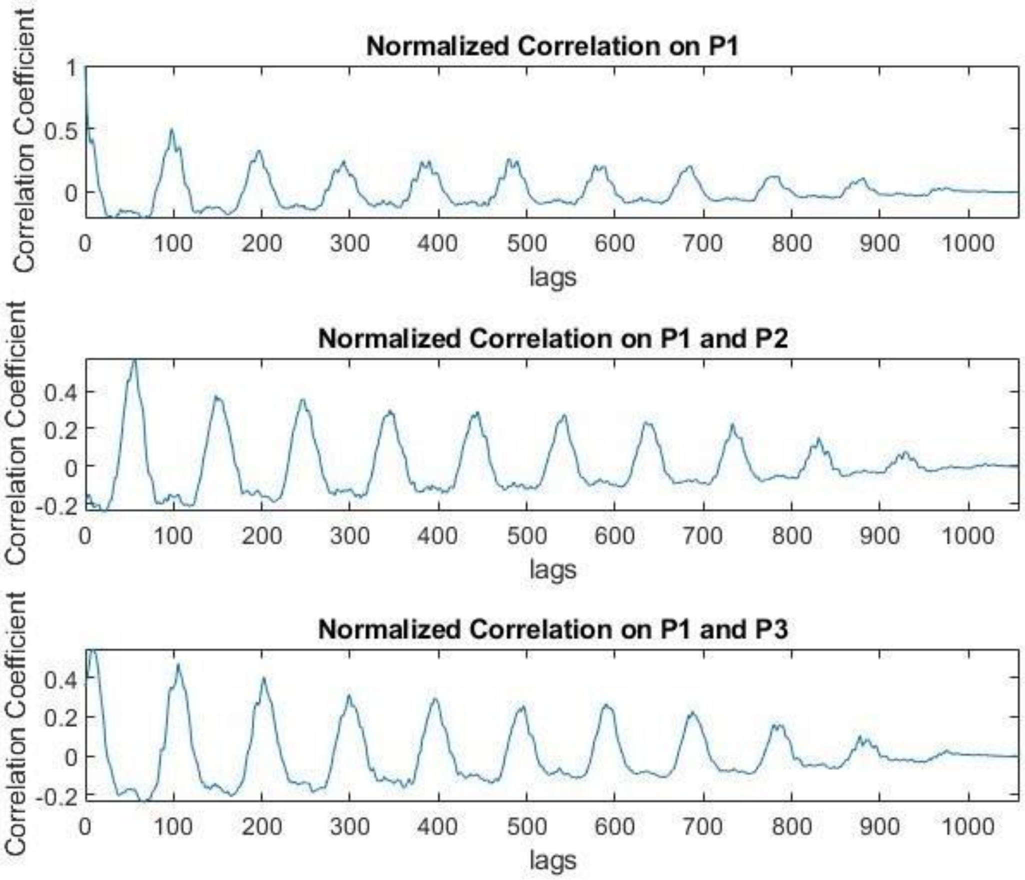
```
%% Extract the frames from video
clc
clear
v = VideoReader('ACM_2716_cylinder_water_tunneltrim_3.MP4');
n = 1;
while hasFrame(v)
    img = readFrame(v);
    imwrite(rgb2gray(img),strcat('img_', num2str(n), '.png'));
    n=n+1;
end
%% Extracting the value of pixel for P1, P2, P3 for each frame
clear all
clc
tstps = (0:(35/1059):35)';
tstps(end)=[];
%Enter the x & y coordinate of the pixel.
x1 = 2260;
x2 = 1930;
x3 = 1580;
y = 1397;
%Creation of an empty array
m1 = [ ];
m2 = [ ];
m3 = [ ];
i = 1;
%While loop reads each and every frame (i.e. an image) of the↙
video saved in the directory (images are saved in the form of↙
img_1.png, img_2.png ...... upto img_1059.png)
while i<=1059
    I = imread(strcat('img_', num2str(i), '.png'));
    %impixel() reads the value of the pixel at the specified↙
 location in the image.
    c1 = impixel(I, x1, y);
    c2 = impixel(I, x2, y);
    c3 = impixel(I, x3, y);
```

```
    %It adds up the pixel values as elements in the column in↙
empty array 'm'
    m1(end+1, 1) = c1(1,1);
    m2(end+1, 1) = c2(1,1);
    m3(end+1, 1) = c3(1,1);
    i = i+1;
end
P1f = m1 - mean(m1);
P2f = m2 - mean(m2);
P3f = m3 - mean(m3);
TS_f = [tstps P1f P2f P3f];
%% Normalized Correlation Coefficient
clc
tiledlayout(3,1)
%P1 & P1
[CC1, lags1] = xcorr(TS_f(:, 2),  'normalized');
%P1 and P2
[CC2, lags2] = xcorr(TS_f(:, 2), TS_f(:, 3),  'normalized');
%P1 and P3
[CC3, lags3] = xcorr(TS_f(:, 2), TS_f(:, 4), 'normalized');
xlim1 = find(lags1==0);
xlim2 = find(lags2==0);
xlim3 = find(lags3==0);
CC1 = CC1(xlim1:end);
CC2 = CC2(xlim2:end);
CC3 = CC3(xlim3:end);
lags1 = lags1(xlim1:end);
lags2 = lags2(xlim2:end);
lags3 = lags3(xlim3:end);
table1 = [lags1' CC1 CC2 CC3];
nexttile
plot(lags1, CC1)
xlabel('lags')
ylabel('Correlation Coefficient')
title('Normalized Correlation on P1')
```

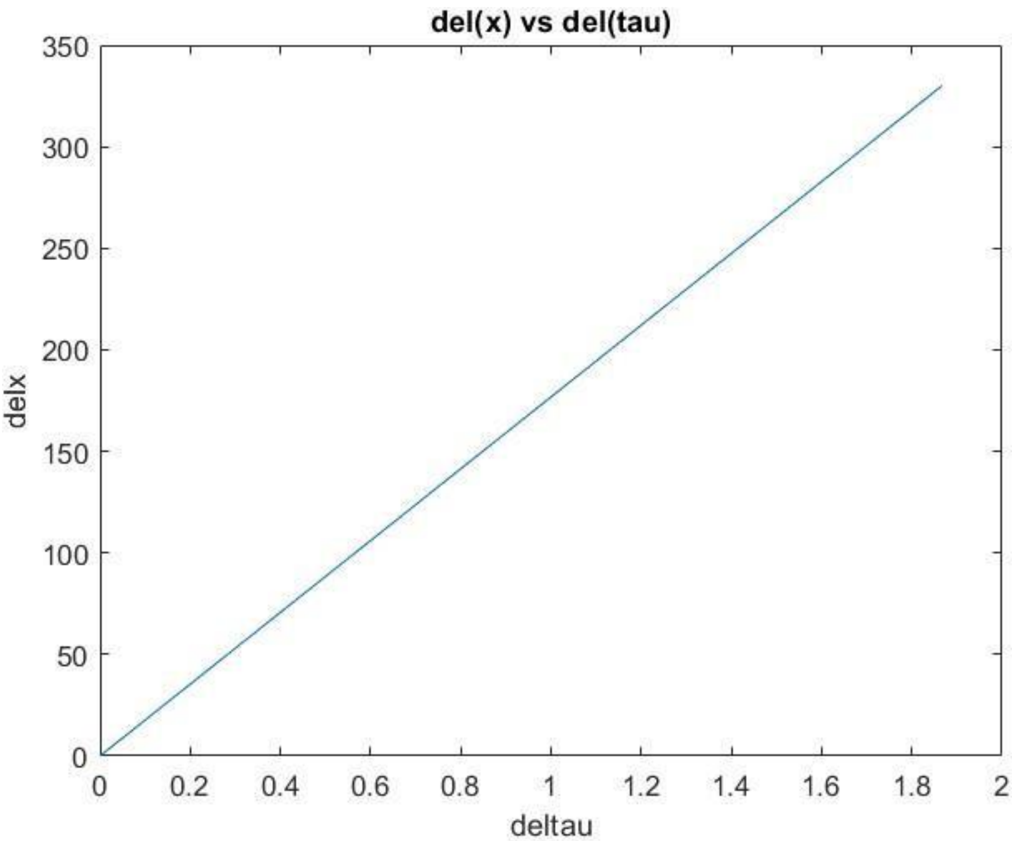
```
axis tight
nexttile
plot(lags2, CC2)
xlabel('lags')
ylabel('Correlation  Coefficient')
title('Normalized Correlation on P1 and P2')
axis tight
nexttile
plot(lags3, CC3)
xlabel('lags')
ylabel('Correlation  Coefficient')
title('Normalized Correlation on P1 and P3')
axis tight
xlabel('lags')
ylabel('Correlation Coefficient')
%% Finding the convection velocity
clc
delx = [0 330 680];
frame1 = find(CC1 == max(CC1));
frame2 = find(CC2 == max(CC2));
frame3 = find(CC3 == max(CC3));
t1 = lags1(frame1);
t2 = lags2(frame2);
t3 = lags3(frame3);
delt = [t1 t2 t3]/30;
plot(delt, delx);
V = (delx(3) - delx(2))*30/(t3-t2); %Convection Velocity

%The convection velocity obtained is 228 pixels/second"
```

Q2 Graphs



Normalized Correlation on P1
Correlation Coefficient
lags
Normalized Correlation on P1 and P2
Correlation Coefficient
lags
Normalized Correlation on P1 and P3
Correlation Coefficient
lags



del(x) vs del(tau)
delx
deltau

(Q3)

```
%% Importing the Data
clc
clear all
format long
a = readmatrix("PIVdata.txt");
b = cell(403, 1); %To store matrices for every realization
%Slicing the data for 400 different realizations
for i=1:400
   idx1 = (i-1)*10000 +1;
   idx2 = i*10000;
   b{i} = a(idx1:idx2, :);
end
%% mean values (U_mean & V_mean)
clc
b{401, 1} = b{1,1};
k=2;
while k<=400
    b{401, 1}(:, 3:4) = b{401, 1}(:, 3:4) + b{k,1}(:, 3:4);
    k=k+1;
end
b{401, 1}(:, 3:4) = b{401, 1}(:, 3:4)./400;
U_mean = b{401, 1}(:, 1:3);
V_mean = b{401, 1}(:, [1, 2, 4]);
%% rms of fluctuating values (U_rms & V_rms)
clc
c = b;
i = 1;
while i<=400
    c{i,1}(:, 3) = c{i,1}(:,3) - U_mean(:,3);
    c{i,1}(:, 4) = c{i,1}(:,4) - V_mean(:,3);
    i = i+1;
end
c{402, 1}(:, 1:2) = c{1,1}(:, 1:2);
c{402, 1}(:, 3:4) = c{1,1}(:, 3:4).^2;
k=2;
```

```
while k<=400
    c{402, 1}(:, 3:4) = c{402, 1}(:, 3:4) + c{k,1}(:, 3:4).^2;
    k=k+1;
end
c{402, 1}(:, 3:4) = sqrt(c{402, 1}(:, 3:4)./400);
U_rms = c{402, 1}(:, 1:3);
V_rms = c{402, 1}(:, [1,2,4]);
%% U_mean plot
clc
x1_idx = find(U_mean==30.3903);
y1 = U_mean(x1_idx, 2);
Um = [];
for i=1:length(y1)
    U_idx = find(U_mean(:,1)==30.3903 & U_mean(:,2)==y1(i));
    Um(end+1,1) = U_mean(U_idx, 3);
end
figure(1)
plot(y1, Um)
title('U-mean plot')
xlabel('y');
ylabel('U-mean')
axis padded
%% V_mean plot
clc
Vm = [];
for i=1:length(y1)
    V_idx = find(V_mean(:,1)==30.3903 & V_mean(:,2)==y1(i));
    Vm(end+1,1) = V_mean(V_idx, 3);
end
figure(3)
plot(y1, Vm)
title('V-mean plot');
xlabel('y')
ylabel('V-mean')
axis padded
```

```
%% U_rms plot
clc
Ur = [];
for i=1:length(y1)
    V_idx = find(U_rms(:,1)==30.3903 & U_rms(:,2)==y1(i));
    Ur(end+1,1) = U_rms(V_idx, 3);
end
figure(3)
plot(y1, Ur)
title('U-rms plot')
xlabel('y')
ylabel('U-rms')
axis padded
%% V_rms plot
clc
Vr = [];
for i=1:length(y1)
    V_idx = find(V_rms(:,1)==30.3903 & V_rms(:,2)==y1(i));
    Vr(end+1,1) = V_rms(V_idx, 3);
end
figure(4)
plot(y1, Vr)
title('V-rms plot')
xlabel('y')
ylabel('V-rms')
axis padded
%% r(x,y) contour at x=30 and y=0
% Positive Lags
clc
n = 4740;
lag = 0;
delx = zeros(21,1);
dely = zeros(21,1);
U_1 = zeros(21, 1);
while lag<=20
```

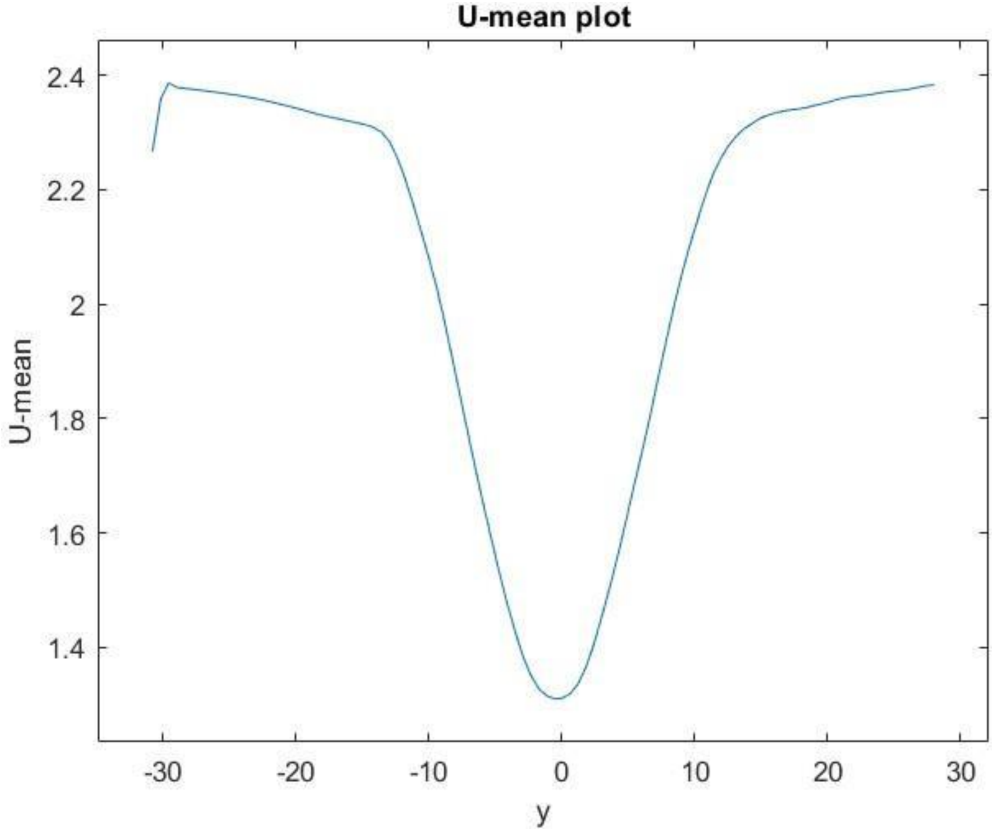
```
P = 0;
Q = 0;
for i=(n-1):-1:1
    U_1(lag+1, 1) = U_1(lag+1, 1) + (U_mean(i, 3)*U_mean(i+lag,↙
3))/(U_rms(i, 3)*U_rms(i+lag, 3));
    P = P + 1;
end
for i = n:1:max(size(U_mean, 1)-lag, n)
    U_1(lag+1, 1) = U_1(lag+1, 1) + (U_mean(i, 3)*U_mean(i+lag,↙
3))/(U_rms(i, 3)*U_rms(i+lag, 3));
    Q = Q + 1;
end
U_1(lag+1 ,1) = U_1(lag+1, 1)/(P + Q);
delx(lag+1, 1) = (U_mean (n, 1) - U_mean(n-lag, 1));
dely(lag+1, 1) = (U_mean (n, 2) - U_mean(n-lag, 2));
lag = lag + 1;
end
% Negative Lags
clc
n = 4740;
lag1 = 0;
delx1 = zeros(21,1);
dely1 = zeros(21,1);
U_2 = zeros(21, 1);
while lag1>=(-20)
P = 0;
Q = 0;
for i=(n-1):-1:1
    U_2(norm(lag1)+1, 1) = U_2(norm(lag1)+1, 1) + (U_mean(i, 3)↙
*U_mean(i-lag1, 3))/(U_rms(i, 3)*U_rms(i-lag1, 3));
    P = P + 1;
end
for i = n:1:max(size(U_mean, 1)+lag1, n)
    U_2(norm(lag1)+1, 1) = U_2(norm(lag1)+1, 1) + (U_mean(i, 3)↙
*U_mean(i-lag1, 3))/(U_rms(i, 3)*U_rms(i-lag1, 3));
```

```
    Q = Q + 1;
end
U_2(norm(lag1)+1 ,1) = U_2(norm(lag1)+1, 1)/(P + Q);
delx1(norm(lag1)+1, 1) = (U_mean (n, 1) - U_mean(n+lag1, 1));
dely1(norm(lag1)+1, 1) = (U_mean (n, 2) - U_mean(n+lag1, 2));
lag1 = lag1 - 1;
end
% Plot of r(del(x), del(y)) for U_mean
clc
lags = (0:1:20)';
lags1 = (-0:-1:-20)';
U_c1 = [U_1 lags];
U_c2 = flip([U_2 lags1]);
U_combined = [U_c2;U_c1];
plot(U_combined(:, 2), U_combined(:,1))
title('r(del(x), del(y)) plot')
xlabel('lags')
ylabel('U')
%% Time Series of Fluctuations at P1, P2 and P3.
clc
U_fluc = zeros(400, 3);
i = 1;
while i<=400
    N1 = griddata(c{i,1}(:,  1), c{i,1}(:, 2), c{i,1}(:, 3), 30,
5, 'nearest');
    U_fluc(i, 1) = N1;
    N2 = griddata(c{i,1}(:,  1), c{i,1}(:, 2), c{i,1}(:, 3), 33,
5, 'nearest');
    U_fluc(i, 2) = N2;
    N3 = griddata(c{i,1}(:,  1), c{i,1}(:, 2), c{i,1}(:, 3), 36,
5, 'nearest');
    U_fluc(i, 3) = N3;
    i = i+1;
end
%% Generating Time Series
```

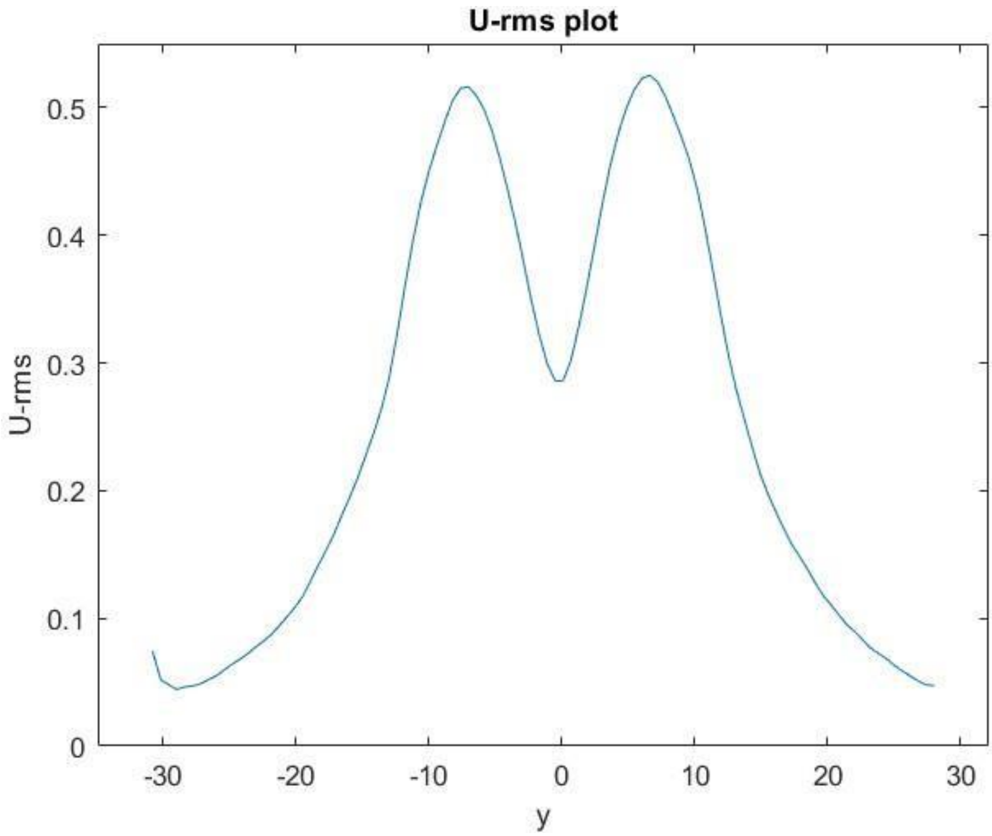
```
clc
tstps = 0:(1/365):400/365;
tstps = tstps';
TimeSeries = [tstps(1:end-1) U_fluc(:, :)];
plot(TimeSeries(:,1), TimeSeries(:,2))
hold on
plot(TimeSeries(:,1), TimeSeries(:,3))
plot(TimeSeries(:,1), TimeSeries(:,4))
xlabel('Fluctuating U velocity')
ylabel('time (t)')
legend('P1 (30, 5)', 'P2 (33,5)', 'P3 (36,5)')
%% Normalised Correlation Coefficient
clc
tiledlayout(3,1)
%For P1
[CC1, lags1] = xcorr(U_fluc(:, 1),  'normalized');
%For P1 and P2
[CC2, lags2] = xcorr(U_fluc(:, 1), U_fluc(:, 2),↙
'normalized');
%For P1 & P3
[CC3, lags3] = xcorr(U_fluc(:, 1), U_fluc(:, 3), 'normalized');
xlim1 = find(lags1==0);
xlim2 = find(lags2==0);
xlim3 = find(lags3==0);
CC1 = CC1(xlim1:end);
CC2 = CC2(xlim2:end);
CC3 = CC3(xlim3:end);
lags1 = lags1(xlim1:end);
lags2 = lags2(xlim2:end);
lags3 = lags3(xlim3:end);
table1 = [lags1' CC1 CC2 CC3];
nexttile
plot(lags1, CC1)
title('Normalized Correlation Coefficient for P1')
xlabel('lags')
```

```
ylabel('Correlation Coefficient')
axis tight
nexttile
plot(lags2, CC2)
title('Normalized Correlation Coefficient for P1 & P2')
xlabel('lags')
ylabel('Correlation Coefficient')
axis tight
nexttile
plot(lags3, CC3)
title('Normalized Correlation Coefficient for P1 & P3')
xlabel('lags')
ylabel('Correlation Coefficient')
axis tight
%% Convection Velocity
clc
delx = [0 3 6];
frame1 = find(CC1 == max(CC1));
frame2 = find(CC2 == max(CC2));
frame3 = find(CC3 == max(CC3));
t1 = lags1(frame1);
t2 = lags2(frame2);
t3 = lags3(frame3);
delt = [t1 t2 t3]*2.74; %frame rate = 365 fps
plot(delt(1:2), delx(1:2));
xlabel('del(tau) in ms')
ylabel('del(x) in mm')
title('del(x) vs del(tau)')
V = (delx(2) - delx(1))/(delt(2)-delt(1)); %Convection Velocity
```

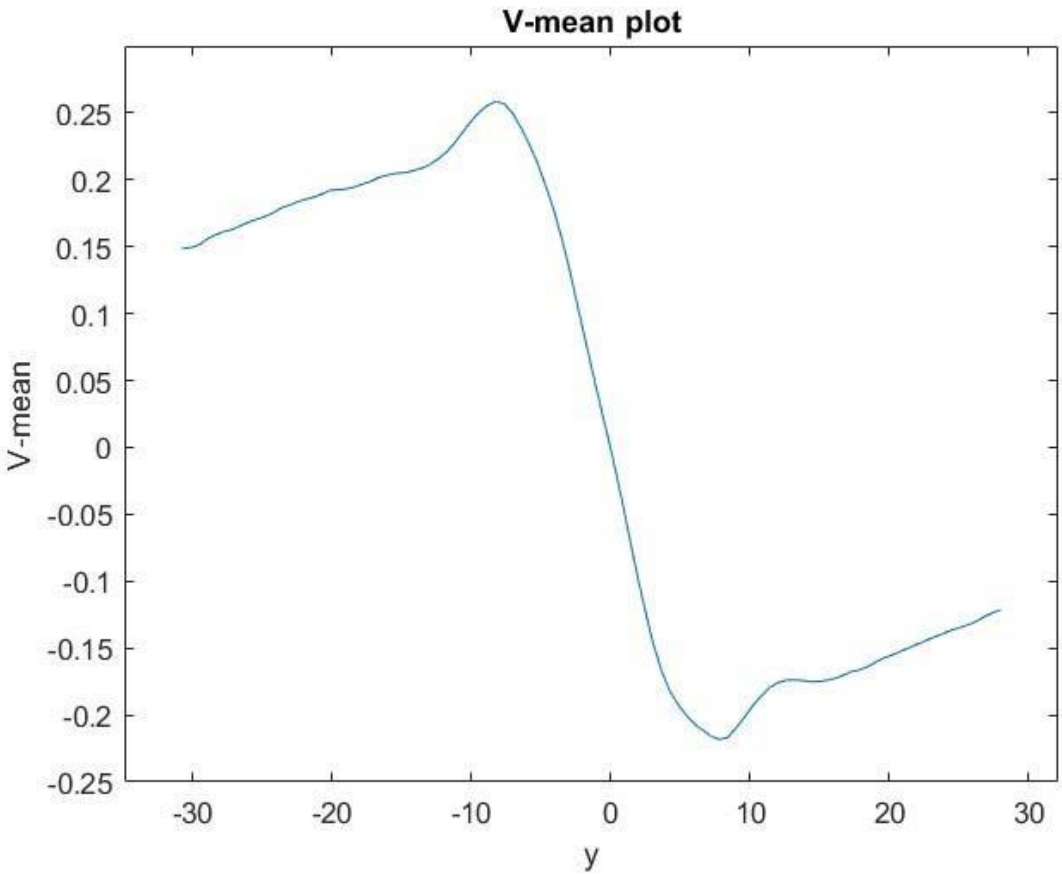
Q3 Graphs



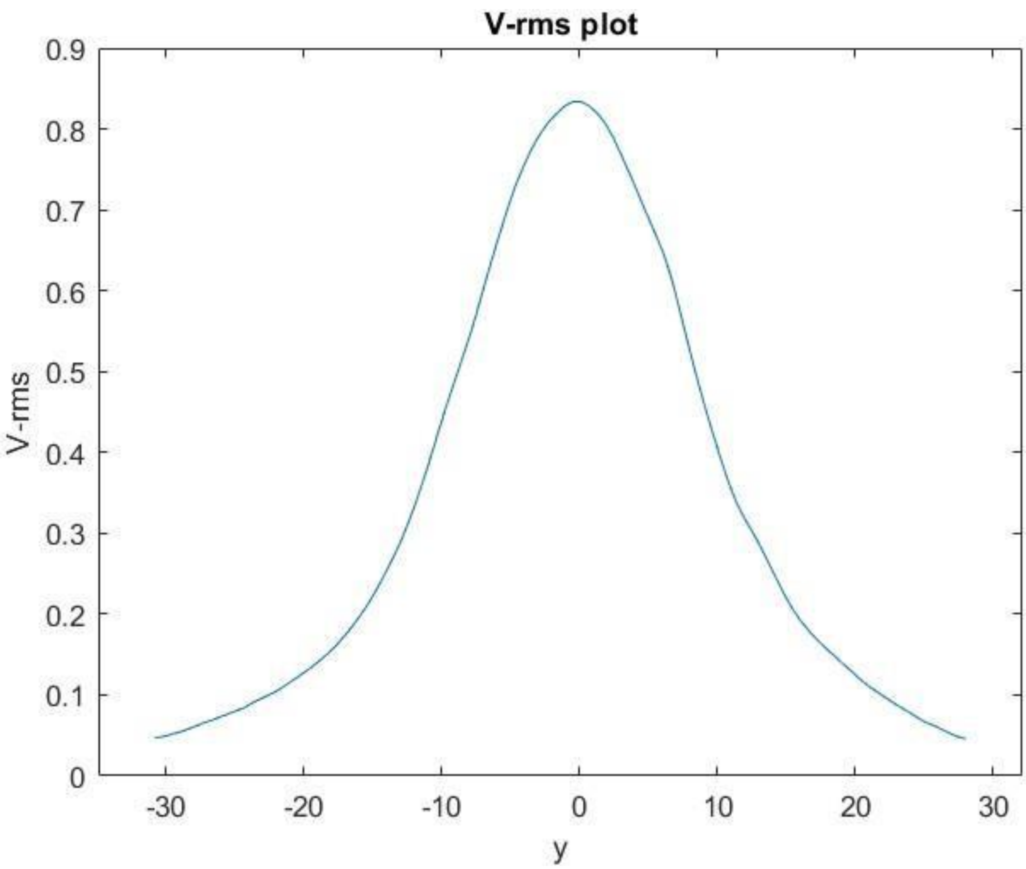
U-mean plot
2.4
2.2
2
1.8
1.6
1.4
U-mean
-30
-20
-10
0
10
20
30
y



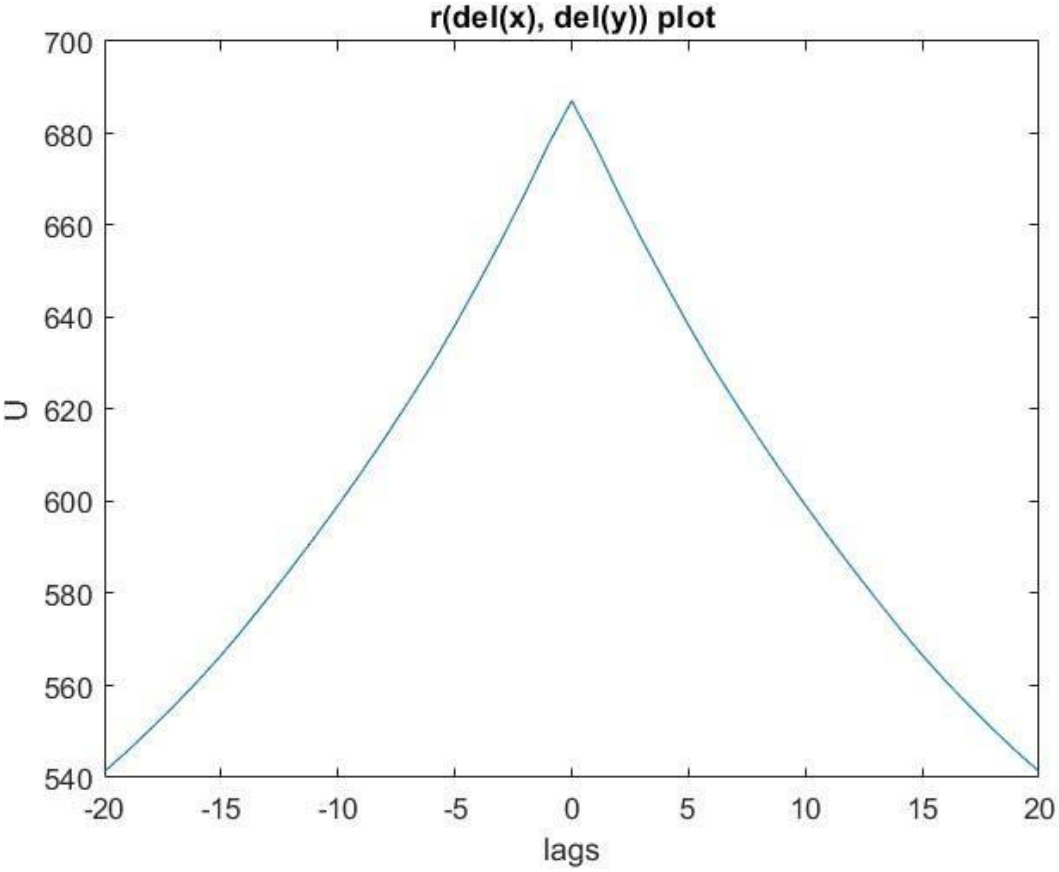
U-rms plot
0.5
0.4
0.3
0.2
0.1
0
U-rms
-30
-20
-10
0
10
20
30
y

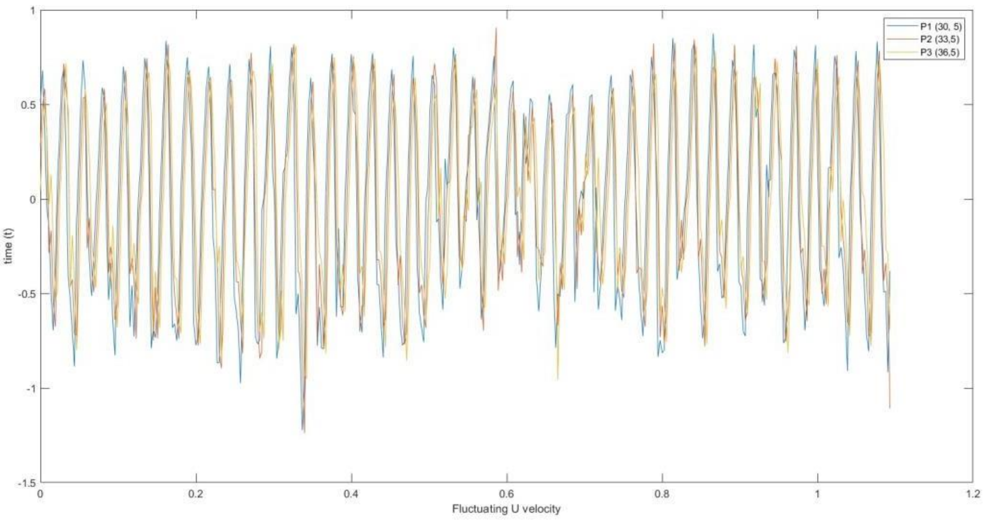


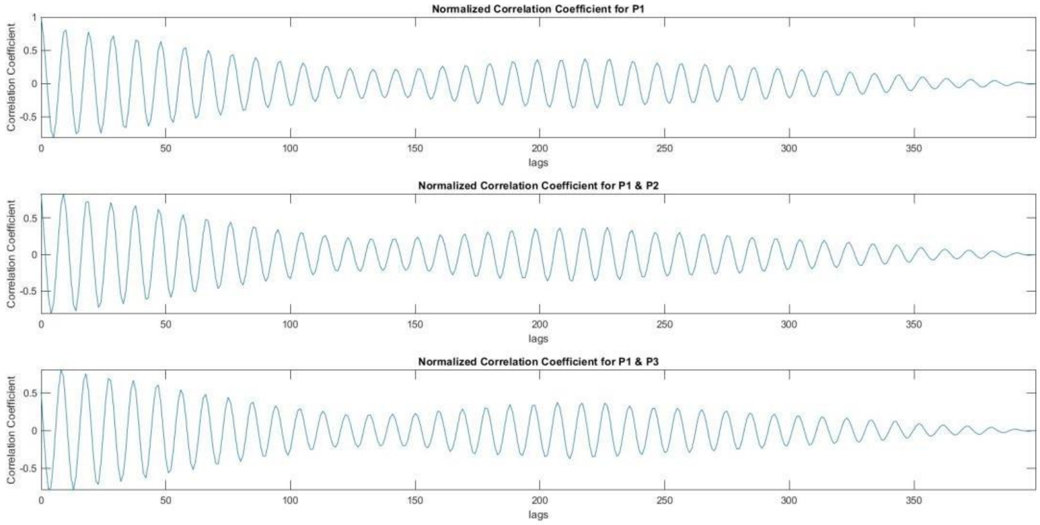
V-mean plot
V-mean
y

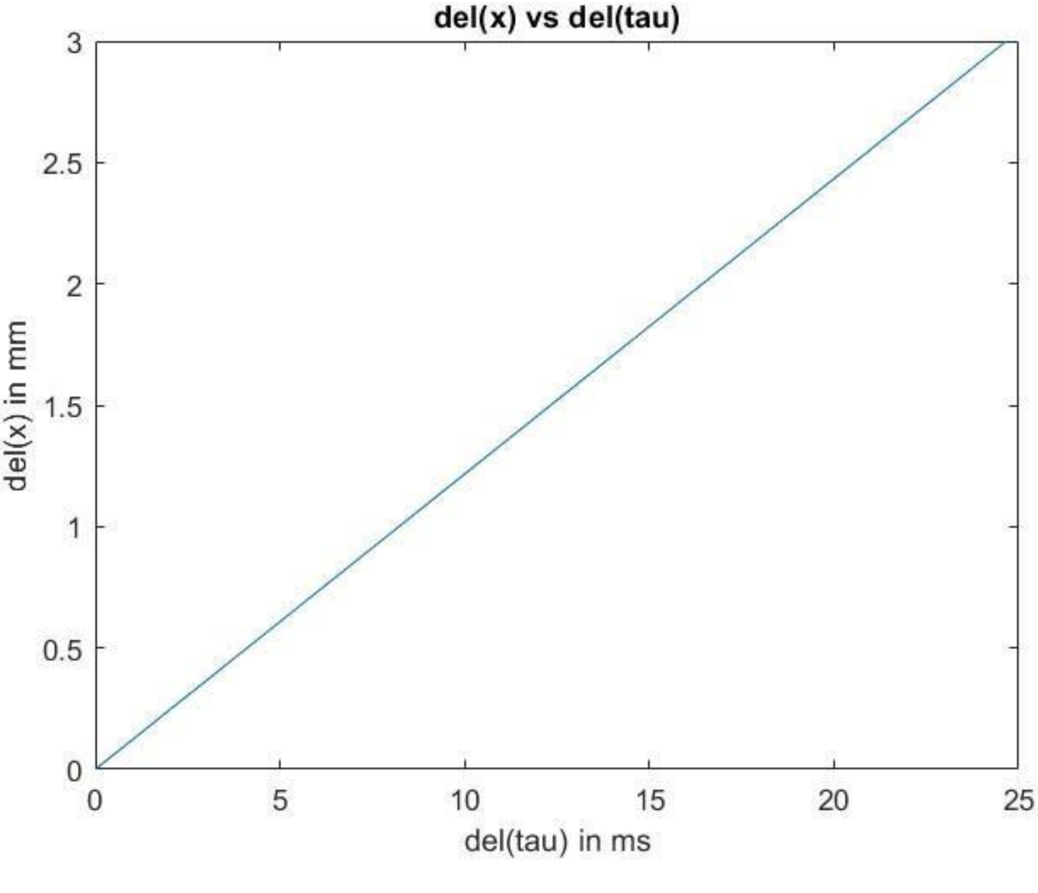


V-rms plot
V-rms
y


r(del(x), del(y)) plot
U
lags
700
680
660
640
620
600
580
560
540
-20
-15
-10
-5
0
5
10
15
20


P1 (30, 5)
P2 (33,5)
P3 (36,5)
time (t)
Fluctuating U velocity
1
0.5
0
-0.5
-1
-1.5
0
0.2
0.4
0.6
0.8
1
1.2


Normalized Correlation Coefficient for P1
Normalized Correlation Coefficient for P1 & P2
Normalized Correlation Coefficient for P1 & P3
Correlation Coefficient
lags


del(x) vs del(tau)
del(x) in mm
del(tau) in ms