Table of Contents

Group 3- ASEN 2002

```
%Luca Barton, Eliza Bourne, Hayden Gebhardt, AJ Lauffer, Linus
 Schmitz, Blake Wilson
% 12/3/2020

close all
clear
clc
% PP = pitot static probe
% VT = Ventury tube
% PT = Pressure transducer
```

Read in Ventury tube vs Pressure Transducer data

```
rho = 1000;    %kg/m^3
g = 9.81;   %m/s^2
R = 287;    % [J/Kg-K]Universal gas constant


A1 = 9.5;
A2 = 1;

VTPTdata = [];
for i = 3:1:14
    files = dir('/MATLAB Drive/ASEN 2002 Lab/Lab 2/Data collection/
VTtoPT');
    long_name = strcat(files(i).folder,'/',files(i).name);
    VTPTdata  = [VTPTdata;load(long_name)];

end
```

Read in Pitot Static Probe vs Pressure Transducer data

```
PPPTdata = [];
for i = 3:1:14
    files = dir('/MATLAB Drive/ASEN 2002 Lab/Lab 2/Data collection/
PPtoPT');
    long_name = strcat(files(i).folder,'/',files(i).name);
    PPPTdata  = [PPPTdata;load(long_name)];

end
```

Read in Water Manometer data and seperate to make sense of it

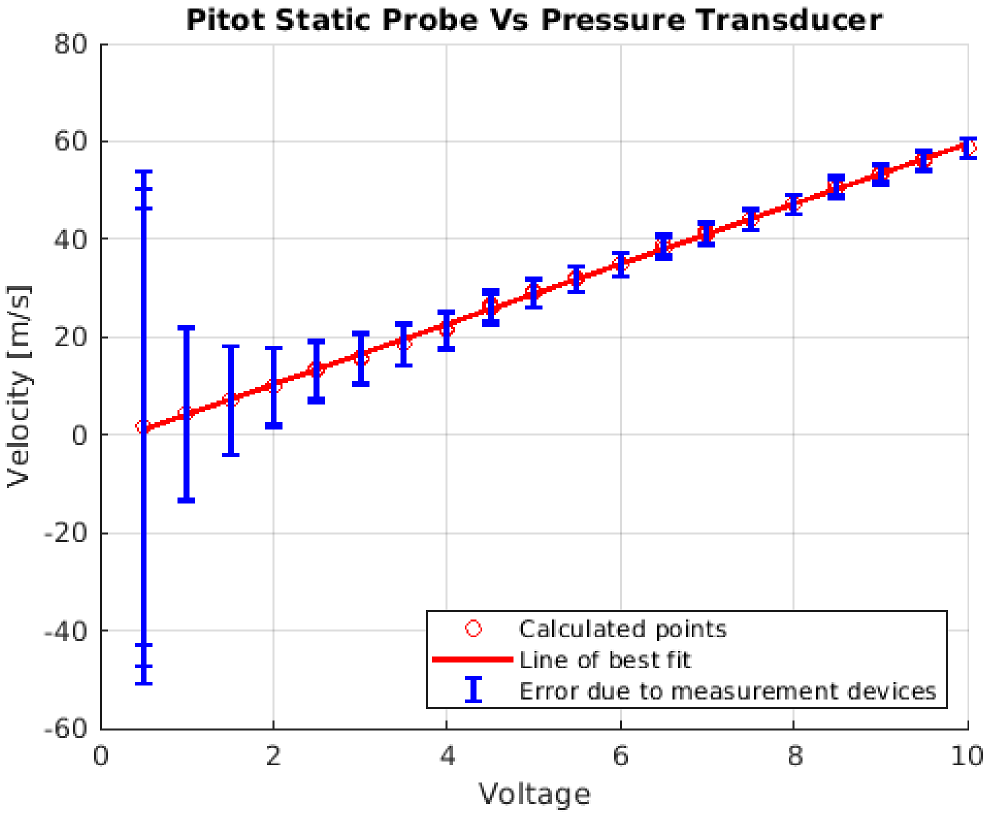
```
WMdata = readtable("water_manometer_data.xlsx");
WMdata(2,:) =[];        %Elliminate erronius data points from a
 retarded group
WMdata = sortrows(WMdata, 3);   %Sort the data based off of
 measurement device
WMdata_PSP = WMdata(1:14,:);    %Select the data that was collected
 with the Pioto-static port
WMdata_VT = WMdata(15:end,:);   %Select the data that was collected
 with the venturi tube

T = mean(VTPTdata(:,2)) ;    %Temperature Atmospheric
P_atm = mean(VTPTdata(:,1));    % Pressure Atmospheric
```

```
Warning: Column headers from the file were modified to make them valid
 MATLAB
identifiers before creating variable names for the table. The original
 column
headers are saved in the VariableDescriptions property.
Set 'VariableNamingRule' to 'preserve' to use the original column
 headers as
table variable names.
```

Pitot Static Probe Velocity calculations (Pressure Transducer)

```
%Calculate the velocities for the Pitot-static port

%uncertainties
unT = .25;
unP_atm = 3450; %Pa 1.5% of operating range 20-250 kPa
unDelP = 68.94; %Pa Appendix A
dvdDel_P = [];
dvdP_atm = [];
dvdT = [];
```

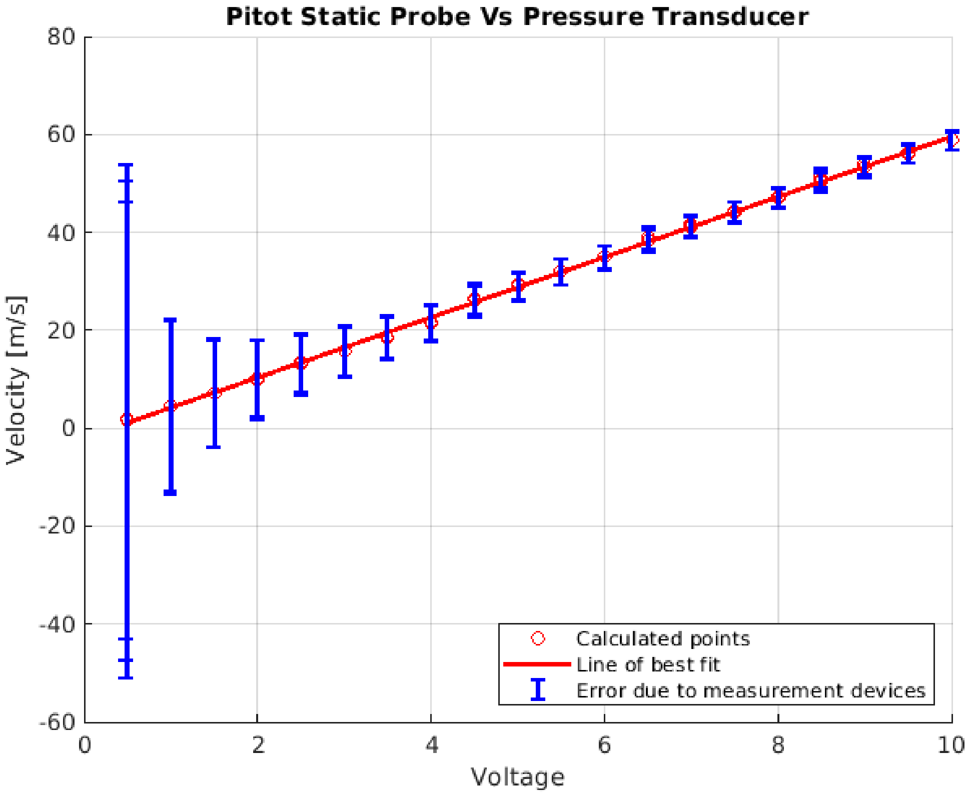
```
unV = [];

for i = 1:1:60
    input1 = PPPTVel(PPPTdata((500 * (i-1) + 1):500 * i, :));
    input2 = PPPTdata((500 * (i-1) + 1):500 * i, end);
    Velocity_pitot(i) = mean(input1);
    Voltages_Pitot(i) = mean(input2);
    %Uncertainty in airspeed
    dvdDel_P(i) = (R*T*unDelP)/(P_atm*sqrt(2*PPPTdata(i*500,3)*(R*T)/
(P_atm)));
    dvdP_atm(i) = (-PPPTdata(i*500,3)*R*T*unP_atm)/
((P_atm)^(2)*sqrt(2*PPPTdata(i*500,3)*(R*T)/(P_atm)));
    dvdT(i) = (PPPTdata(i*500,3)*R*unT)/
(P_atm*sqrt(2*PPPTdata(i*500,3)*(R*T)/(P_atm)));

    unV_PitotTrans(i) = sqrt(dvdDel_P(i)^2+dvdP_atm(i)^2+dvdT(i)^2);
end
%calculate the best fit line and error for the PPPT data
[bestFit_Pitot,Error_Pitot,b1,m1] =
 LSFunc(Voltages_Pitot',Velocity_pitot');

%plot all of the data and associated error bars
figure(1)
hold on
grid on
plot(Voltages_Pitot, Velocity_pitot, 'ro')
plot(Voltages_Pitot,bestFit_Pitot,'r',"LineWidth",2)
z =
 errorbar(Voltages_Pitot,Velocity_pitot,unV_PitotTrans,'b',"LineWidth",2);
z.LineStyle = 'none';
% x = errorbar(Voltages_Pitot,Velocity_pitot,Error_Pitot);
x.LineStyle = 'none';
title("Pitot Static Probe Vs Pressure Transducer")
xlabel("Voltage")
ylabel("Velocity [m/s]")
legend("Calculated points", "Line of best fit", "Error due to
 measurement devices", "Location","southeast")

hold off
```

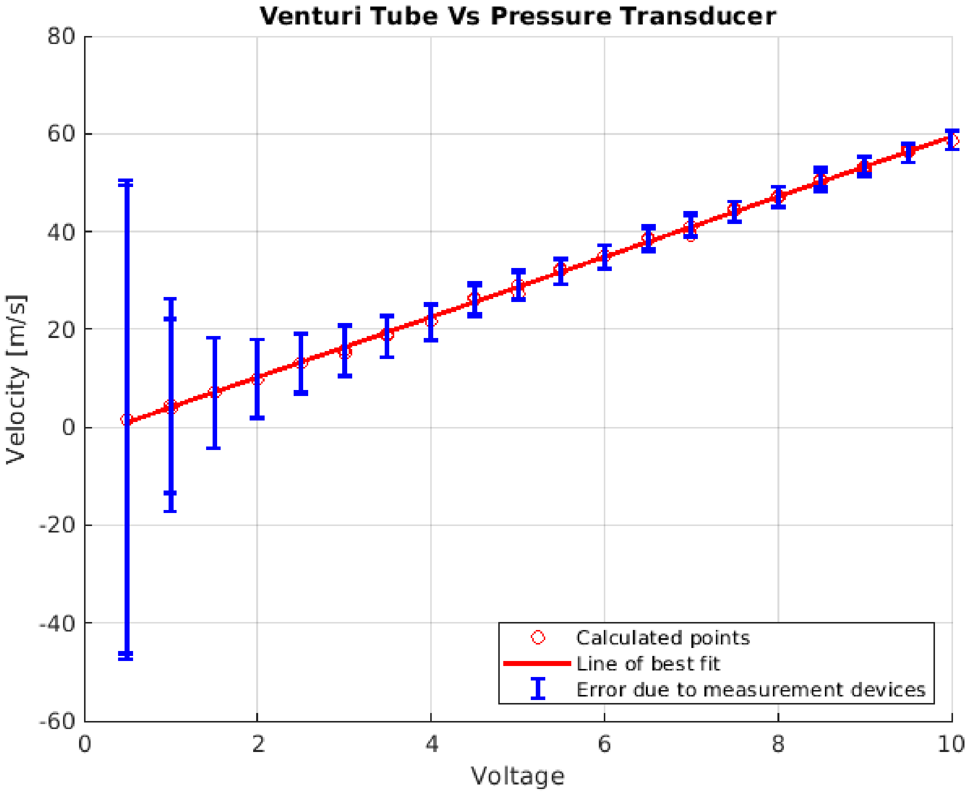


Ventury Tube Velocity calculations (Pressure Transducer)

```
%Calculate the velocities for the Venturi tube

%uncertainties
unT = .25;
unP_atm = 3450;
unDelP = 68.94;
A = [];
B = [];
C = [];

for i = 1:1:60
    input1 = VTPTVel(VTPTdata((500 * (i-1) + 1):500 * i, :));
    input2 = VTPTdata((500 * (i-1) + 1):500 * i, end);
    Velocity_Venturi(i) = mean(input1);
    Voltages_Venturi(i) = mean(input2);

    %uncertainty
    A(i) = ((R * T * unDelP) / (P_atm * (1 - (A2/A1)^2) * sqrt ((2 *
 VTPTdata(i * 500,3) * R * T) / (P_atm * (1 - (A2/A1)^2)))));
```

```
    B(i) = (((-(1 - (A2/A1)^2)) * VTPTdata(i * 500 ,3) * R
 * T * unP_atm) / ((P_atm * (1 - (A2/A1)^2))^2 * sqrt ((2 *
 VTPTdata(i*500,3) * R * T) / (P_atm * (1 - (A2/A1)^2)))));
    C(i) = ((VTPTdata(i * 500,3) * R * unT) / (P_atm * (1 - (A2/
A1)^2) * sqrt ((2 * VTPTdata(i*500,3) * R * T) / (P_atm * (1 - (A2/
A1)^2)))));

    Venturi_Error_Trans(i) = sqrt((A(i)^2)+(B(i)^2)+(C(i)^2));
end
%calculate the best fit line and error for the VTPT data
[bestFit_Venturi,Error_Venturi,b2,m2] =
 LSFunc(Voltages_Venturi',Velocity_Venturi');

%plot all of the data and associated error bars
figure(2)
hold on
grid on
plot(Voltages_Venturi, Velocity_Venturi, 'or')
plot(Voltages_Venturi,bestFit_Venturi,'r',"LineWidth",2)
z =
 errorbar(Voltages_Pitot,Velocity_pitot,Venturi_Error_Trans,'b',"LineWidth",2);
z.LineStyle = 'none';
% x =
 errorbar(Voltages_Venturi,Velocity_Venturi,Error_Venturi,'b',"LineWidth",2);
x.LineStyle = 'none';

title("Venturi Tube Vs Pressure Transducer")
xlabel("Voltage")
ylabel("Velocity [m/s]")
legend("Calculated points", "Line of best fit", "Error due to
 measurement devices", "Location","southeast")
hold off
```

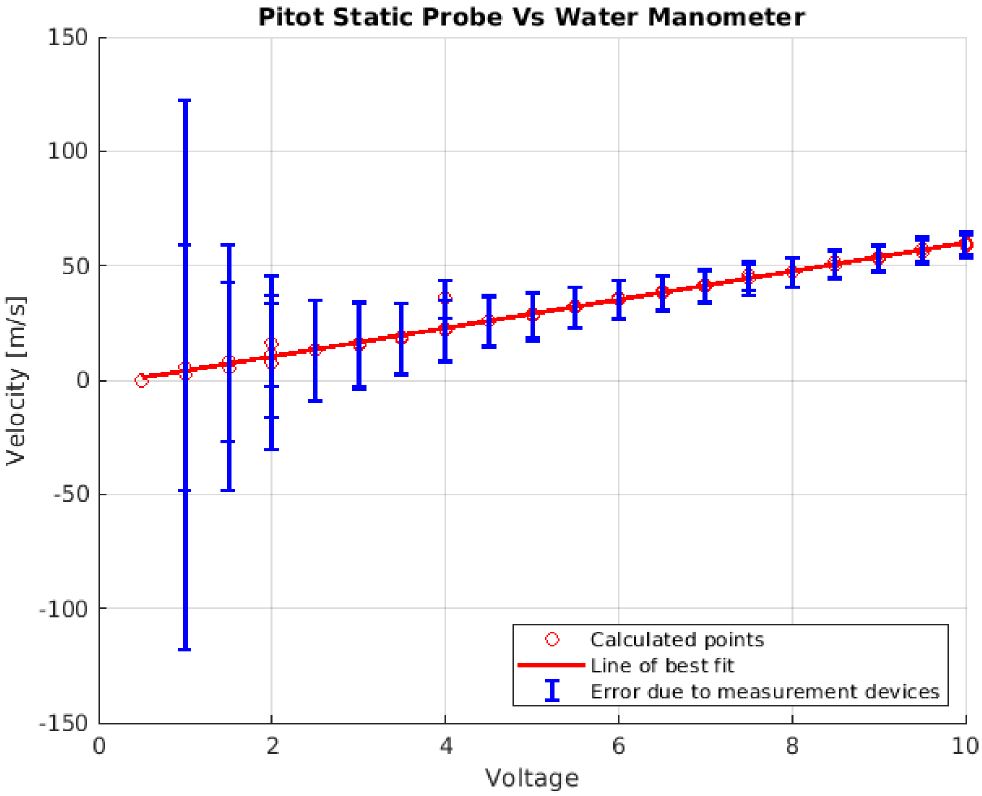

Pitot Static Probe Vs Pressure Transducer
Velocity [m/s]
80
60
40
20
0
-20
-40
-60
0
2
4
6
8
10
Voltage
Calculated points
Line of best fit
Error due to measurement devices


Venturi Tube Vs Pressure Transducer
Velocity [m/s]
80
60
40
20
0
-20
-40
-60
0
2
4
6
8
10
Voltage
Calculated points
Line of best fit
Error due to measurement devices

Water Manometer height to pressure calculations (for Pitot-static)

```
%Voltage vs height vector

% This one we have dynamic pressure so we need to just calculate
 velocity
% with this
VH_data = table2array(WMdata_PSP(:,4:5));
VH_data = [VH_data;table2array(WMdata_PSP(:,6:7))];
VH_data = [VH_data;table2array(WMdata_PSP(:,8:9))];
VH_data = [VH_data;table2array(WMdata_PSP(:,10:11))];
VH_data = [VH_data;table2array(WMdata_PSP(:,12:13))];

%constants
LD = length(VH_data(:,2));  %Length of VH_data (with respect to pitot-
static)
VH_data(:,3:5) = zeros(LD,3);   %Creating three new columns
VH_data(:,3) = VH_data(:,2) * 0.0254;   %converting from in to m and
 populating a new column to that


%uncertainties for using water manometer
unT = .25;
unP_atm = 3450;
unDelP = 0.0254*rho*g; %.1 inches to m then multiplied by rho g
dvdDel_P = [];
dvdP_atm = [];
dvdT = [];
unV = [];

for i = 1:LD
    P = (VH_data(i,3))*rho*g;    % Calculating pressure (this will be
 in pa)
    VH_data(i,4) = P;
    VH_data(i,5) = sqrt((2* P * (R * T)/P_atm));
    %Uncertainty in airspeed
    dvdDel_P(i) = (R*T*unDelP)/(P_atm*sqrt(2*VH_data(i,4)*(R*T)/
(P_atm)));
    dvdP_atm(i) = (-VH_data(i,4)*R*T*unP_atm)/
((P_atm)^(2)*sqrt(2*VH_data(i,4)*(R*T)/(P_atm)));
    dvdT(i) = (VH_data(i,4)*R*unT)/(P_atm*sqrt(2*VH_data(i,4)*(R*T)/
(P_atm)));

    unV_UTubePitotStatic(i) =
 sqrt(dvdDel_P(i)^2+dvdP_atm(i)^2+dvdT(i)^2);
end


% Call LSFunc
[bestFit_WM_Pitot,Error_WM_Pitot,b3,m3] =
 LSFunc(VH_data(:,1),VH_data(:,5));
```

```
% plot Everything
figure(3)
hold on
grid on
plot(VH_data(:,1), VH_data(:,5), 'ro')
plot(VH_data(:,1),bestFit_WM_Pitot,'r',"LineWidth",2)
z =
 errorbar(VH_data(:,1),VH_data(:,5),unV_UTubePitotStatic,'b',"LineWidth",2);
z.LineStyle = 'none';
% x =
 errorbar(VH_data(:,1),VH_data(:,5),Error_WM_Pitot,'b',"LineWidth",2);
x.LineStyle = 'none';

title("Pitot Static Probe Vs Water Manometer")
xlabel("Voltage")
ylabel("Velocity [m/s]")
legend("Calculated points", "Line of best fit", "Error due to
 measurement devices", "Location","southeast")
hold off
```

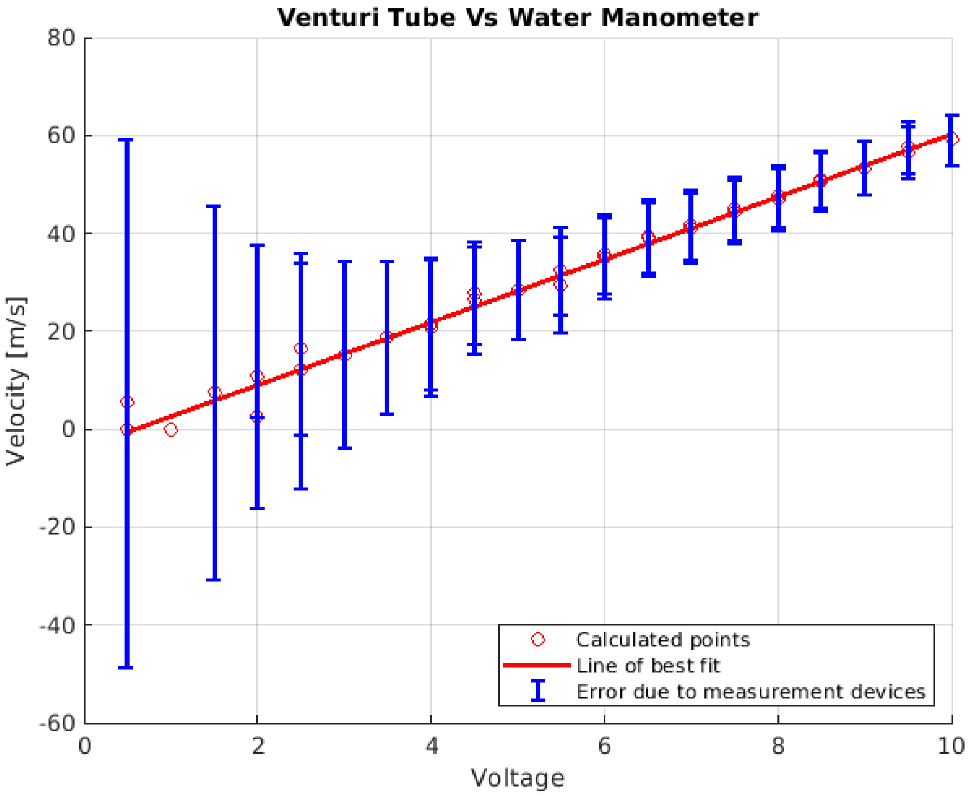


Water Manometer height to pressure calculations (for venturi tube)

```
%Voltage vs height vector
```

```
% This one we have static pressure
VHT_data = table2array(WMdata_VT(:,4:5));
VHT_data = [VHT_data;table2array(WMdata_VT(:,6:7))];
VHT_data = [VHT_data;table2array(WMdata_VT(:,8:9))];
VHT_data = [VHT_data;table2array(WMdata_VT(:,10:11))];
VHT_data = [VHT_data;table2array(WMdata_VT(:,12:13))];

LDT = length(VHT_data(:,2));
VHT_data(:,3:5) = zeros(LDT,3);
VHT_data(:,3) = VHT_data(:,2) * 0.0254;     % Convert from in to m

%uncertainties

A = [];
B = [];
C = [];
Venturi_Error_Utube = [];

for i = 1:LDT
    P = (VHT_data(i,3))*rho*g;    % This will be in pa
    VHT_data(i,4) = P;
    VHT_data(i,5) = sqrt(2*P * (R * T)/(P_atm * (1- (A2/A1)^2)));
    %uncertainty
    A(i) = ((R * T * unDelP) / (P_atm * (1 - (A2/A1)^2) * sqrt ((2 *
 VHT_data(i,4) * R * T) / (P_atm * (1 - (A2/A1)^2)))));
    B(i) = (((-(1 - (A2/A1)^2)) * VHT_data(i,4) * R * T * unP_atm) /
 ((P_atm * (1 - (A2/A1)^2))^2 * sqrt ((2 * VHT_data(i,4) * R * T) /
 (P_atm * (1 - (A2/A1)^2)))));
    C(i) = ((VHT_data(i,4) * R * unT) / (P_atm * (1 - (A2/A1)^2) *
 sqrt ((2 * VHT_data(i,4) * R * T) / (P_atm * (1 - (A2/A1)^2)))));

    Venturi_Error_Utube(i) = sqrt((A(i)^2)+(B(i)^2)+(C(i)^2));

end

Venturi_Error_Utube(1) = 0;
Venturi_Error_Utube(3) = 0;
Venturi_Error_Utube(4) = 0;

% plot Everything
% Call LSFunc
[bestFit_WM_Venturi,Error_WM_Venturi,b4,m4] =
 LSFunc(VHT_data(:,1),VHT_data(:,5));

% plot Everything
figure(4)
hold on
grid on
plot(VHT_data(:,1), VHT_data(:,5), 'ro')
plot(VHT_data(:,1),bestFit_WM_Venturi,'r',"LineWidth",2)
z =
 errorbar(VHT_data(:,1),VHT_data(:,5),Venturi_Error_Utube,'b',"LineWidth",2);
z.LineStyle = 'none';
```

```
% x = errorbar(VHT_data(:,1),VHT_data(:,5),Error_WM_Venturi);
x.LineStyle = 'none';

title("Venturi Tube Vs Water Manometer")
xlabel("Voltage")
ylabel("Velocity [m/s]")
legend("Calculated points", "Line of best fit", "Error due to
 measurement devices", "Location","southeast")
hold off
```



Equations for finding the velocity from the input votage

```
%venturi tube Presure transducer
y1 = @(x) m1 * x + b1;
%Pitot tube Presure transducer
y2 = @(x) m2 * x + b2;
%venturi tube Water Manometer
y3 = @(x) m3 * x + b3;
%Pitot tube Water Manometer
y4 = @(x) m4 * x + b4;
```

Functions

```
function [output1] = PPPTVel(input1)
    deltaP = input1(:,3);
    R = 287;
    temp = input1(:,2);
    Pressure_atm = input1(:,1);
    output1 = sqrt(2.*deltaP .* (R .* temp)./Pressure_atm);
end

% Least Squares Function
function [VelAvg,VelErr,A,B] = LSFunc(x,y)

    %Find number of data points in the vectors
    N = length(x);

    % Find linear best fit coefficients A and B
    % Create H matrix
    H = [ones(N,1) x];

%     del = N*sum(x.^2)-(sum(x))^2;
%     A = (sum(x.^2)+sum(y.^2)-sum(x)*sum(x.*y))/del;
%     B = (N*sum(x.*y)-sum(x)*sum(y))/del;
%
%     SigmaY = sqrt(sum((y-A-B.*x).^2))/sqrt(N-2);

    SigmaY = 0.1;

    % Create W matrix (hint: type <help diag> in command line)
    W = ones(N,1); % set W as identity matrix
    W = (1/SigmaY^2)*W;% recompute the W matrix using SigmaY
    W = diag(W);

    % Solve for P matrix
    p1 = H' * W;
    p2 = p1 * H;
    P = inv(p2);

    % Solve for x_hat matrix and extract A and B parameters
    x_hat = P * H' * W * y;
    A = x_hat(1);
    B = x_hat(2);

    % extract uncertainty in A and uncertainty in B from P matrix
   % A_error = sqrt(P(1));
    %B_error = sqrt(P(4));

    VelAvg = A + B * x; % best fit line

    VelErr = H * P * H'; % predicted error
    VelErr = diag(VelErr);

end
```

```
function [output1] = VTPTVel(input1)
    deltaP = input1(:,3);
    R = 287;
    temp = input1(:,2);
    Pressure_atm = input1(:,1);
    A2 = 1;
    A1 = 9.5;
    output1 = sqrt(2.*deltaP .* (R .* temp)./(Pressure_atm .* (1- (A2/
A1)^2)));
end
```

Published with MATLAB® R2020b