
Table of Contents

.....	1
Question 1	1
Question 2	2
Question 3	3

```
clc, clear, close all
```

Question 1

```
overfiltered = load("OverFiltered_Response.mat");
overfiltered = overfiltered.rawdata;

well_filtered = load("Filtered_Response.mat");
well_filtered = well_filtered.rawdata;

underfiltered = load("Underfiltered_Response.mat");
underfiltered = underfiltered.rawdata;

% Create a figure window
figure;

% Plot Overfiltered Data
subplot(3,1,1); % 3 rows, 1 column, 1st subplot
plot(overfiltered(:,1), overfiltered(:,2), 'b-', 'DisplayName', 'Raw Data');
hold on;
plot(overfiltered(:,1), overfiltered(:,3), 'r-', 'DisplayName', 'Filtered
Data');
title('Overfiltered Data ( $\alpha = 0.001$ )');
xlabel('Time (s)');
ylabel('Distance (m)');
legend;

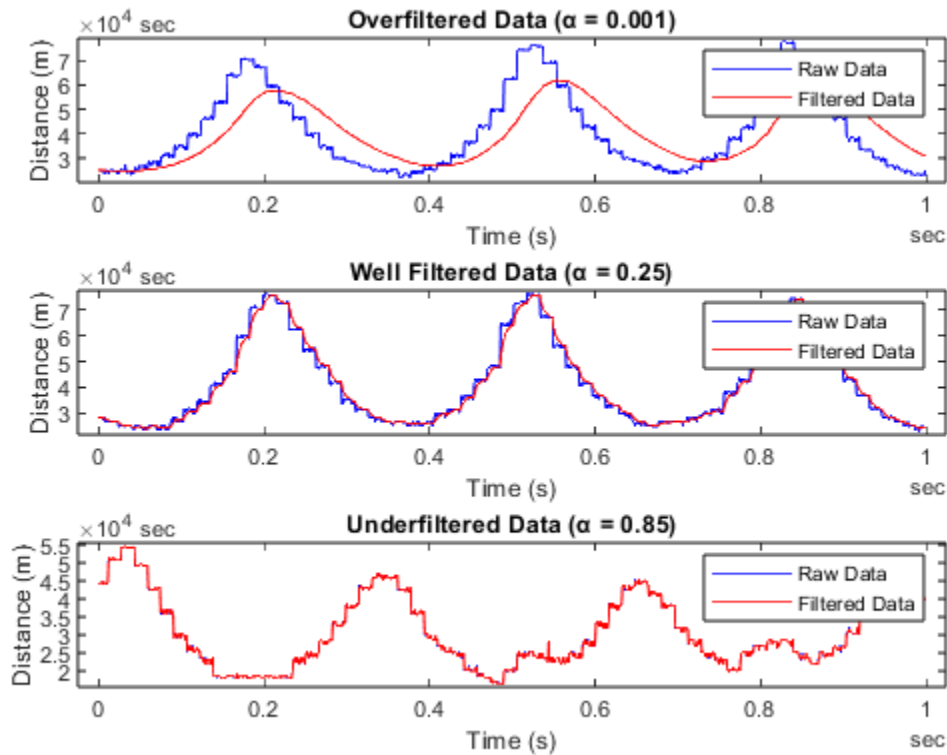
% Plot Well Filtered Data
subplot(3,1,2); % 2nd subplot
plot(well_filtered(:,1), well_filtered(:,2), 'b-', 'DisplayName', 'Raw
Data'); hold on;
plot(well_filtered(:,1), well_filtered(:,3), 'r-', 'DisplayName', 'Filtered
Data');
title('Well Filtered Data ( $\alpha = 0.25$ )');
xlabel('Time (s)');
ylabel('Distance (m)');
legend;

% Plot Underfiltered Data
subplot(3,1,3); % 3rd subplot
plot(underfiltered(:,1), underfiltered(:,2), 'b-', 'DisplayName', 'Raw
Data'); hold on;
plot(underfiltered(:,1), underfiltered(:,3), 'r-', 'DisplayName', 'Filtered
```

```

Data');
title('Underfiltered Data ( $\alpha = 0.85$ )');
xlabel('Time (s)');
ylabel('Distance (m)');
legend;

```



Question 2

```

% Data from table 4.2.1
distances_cm = [2, 3, 4, 6, 8, 10, 14, 18, 22, 26, 30, 34];
sensor_voltages_V = [1.731, 1.41, 1.217, 0.898, 0.7083, 0.5754, 0.4235,
0.3251, 0.2623, 0.2007, 0.1757, 0.1719];

% Create a figure
figure;

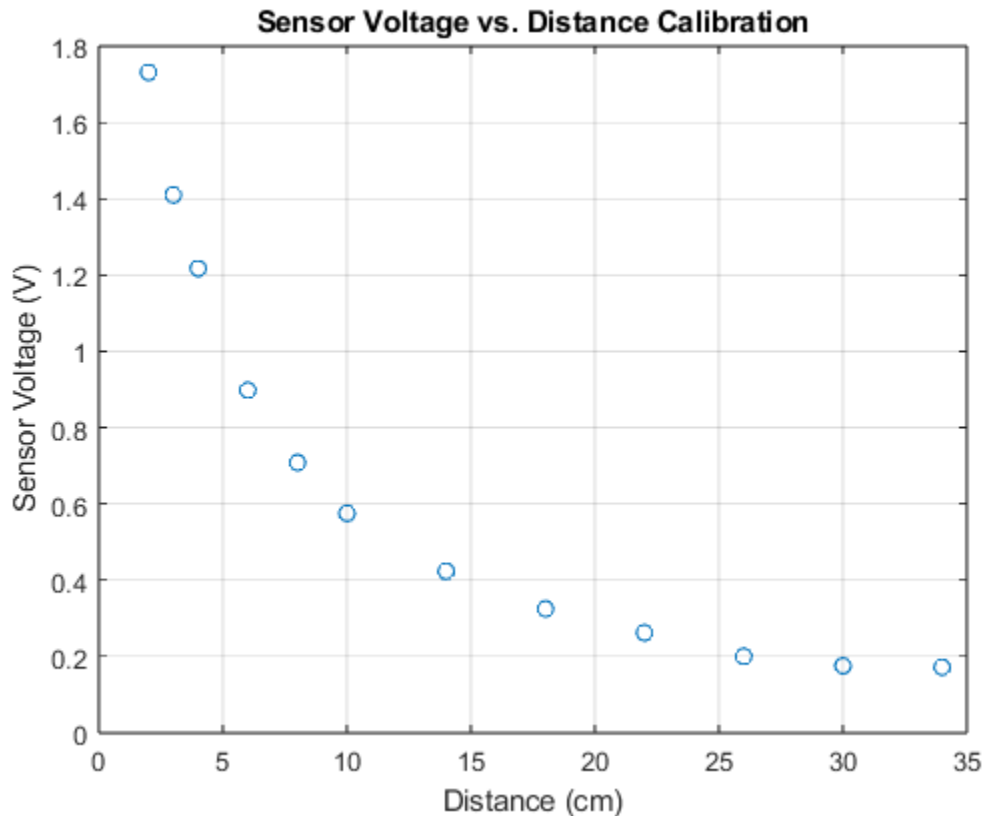
% Plot the data
plot(distances_cm, sensor_voltages_V, 'o');

% Set the axes labels
xlabel('Distance (cm)');
ylabel('Sensor Voltage (V)');

% Set the title of the plot
title('Sensor Voltage vs. Distance Calibration');

```

```
% Enable grid
grid on;
```



Question 3

```
% Data from table 4.2.2
distances_cm = [2.5, 3.5, 5, 7, 9, 12, 16, 20, 24, 28, 32, 36];
sensor_voltages_V = [1.708, 1.354, 1.05, 0.8, 0.6372, 0.4995, 0.37, 0.2924,
0.2317, 0.1956, 0.1596, 0.1335];

% Create a figure
figure;

% Plot the data as points
plot(distances_cm, sensor_voltages_V, 'o');

% Set the axes labels
xlabel('Distance (cm)');
ylabel('Sensor Voltage (V)');

% Set the title of the plot
title('Sensor Voltage vs. Distance Verification');

% Perform a linear fit
coefficients = polyfit(distances_cm, sensor_voltages_V, 1);
% Create a vector of distances for the fit line to cover
```

```

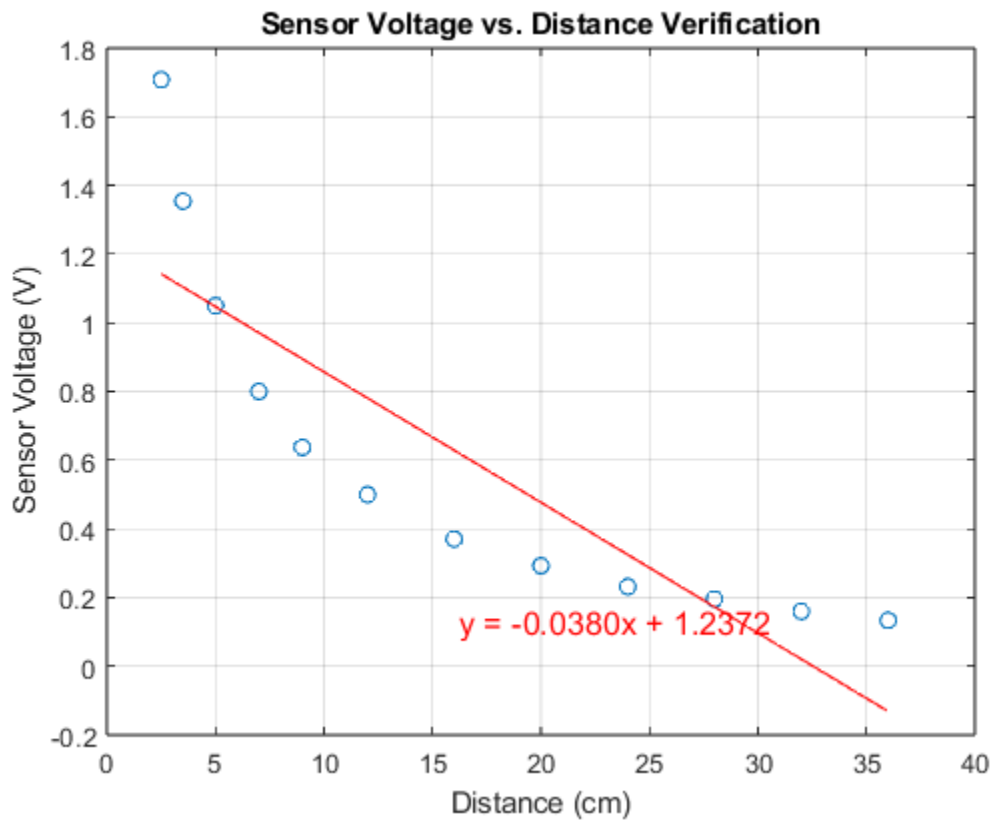
fitX = linspace(min(distances_cm), max(distances_cm), 100);
% Evaluate the fit line
fitY = polyval(coefficients, fitX);

% Plot the fit line
hold on; % Hold on to the current plot
plot(fitX, fitY, '-r', 'DisplayName', 'Fit Line');
hold off;

% Display the fit equation on the plot
fit_equation = sprintf('y = %.4fx + %.4f', coefficients(1), coefficients(2));
text(mean(distances_cm), min(sensor_voltages_V), fit_equation, 'FontSize',
12, 'Color', 'red');

% Enable grid
grid on;

```



Published with MATLAB® R2023b