Uppsala Universitet

Scientific Computing I (1TD393)

Mini Project 1

Report By (Group 36)

Gnana Spandana Akumalla

Nandini M S

INTRODUCTION

The market for battery electric vehicles (BEVs) is growing steadily because of the environmental benefits, compared to conventional vehicles. The energy consumption of an electric vehicle (EV) is one of the most important components since it dictates the dynamic performance and charging time of the vehicle. The speed can be used to compute the energy consumption of an electric vehicle. Another important attribute is the range or the maximum mileage the vehicle can go before charging is required. In order to calculate the range of vehicle, we need two main inputs: the speed of the vehicle and the energy consumption per kilometer.

We are interested in approximating the energy consumption as a function of speed for two different models of electric cars, Volkswagen Passat and Tesla roadster.

In this report, it is summarized how to approximate energy consumption of two different electric vehicle models based on the input data given. To achieve this purpose, we have computed the coefficients of different degree polynomials manually using the sum of least squares method and verified them using inbuilt functions in Matlab. Calculations and observations are done using Matlab software. Based on the observations the 'best fit' curve was concluded for different electric vehicles models. The range or maximum distance that can be travelled is also computed at different speeds and a comparison is made for both the models.

RESULTS

Part 1

a) To approximate a second-degree polynomial, we determine the coefficients a_0, a_1 and a_2 so that the polynomial $p_2(v) = a_0 + a_1v + a_2v^2$ fits best to the measurements. We set up an equation A*z* = e, for each set of measurements and solve in the least square sense to find *z*. In this case, we define the matrices A, *z* and e as follows:

$$A = \begin{bmatrix} 1 & 20 & 400 \\ 1 & 40 & 1600 \\ 1 & 60 & 3600 \\ 1 & 80 & 6400 \\ 1 & 100 & 10000 \\ 1 & 120 & 14400 \end{bmatrix} \qquad z = \begin{bmatrix} a0 \\ a1 \\ a2 \end{bmatrix} \qquad e = \begin{bmatrix} 270 \\ 230 \\ 210 \\ 190 \\ 240 \\ 260 \end{bmatrix}$$

The solution is given mathematically by the normal equations $A^T Az = A^T e$. Substituting with the values, we get,

$$\underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 20 & 40 & 60 & 80 & 100 & 120 \\ 400 & 1600 & 3600 & 6400 & 10000 & 14400 \end{bmatrix} * \begin{bmatrix} 1 & 20 & 400 \\ 1 & 40 & 1600 \\ 1 & 60 & 3600 \\ 1 & 80 & 6400 \\ 1 & 100 & 10000 \\ 1 & 120 & 14400 \end{bmatrix}}_{\begin{bmatrix} 6 & 420 & 36400 \\ 420 & 36400 & 3528000 \\ 36400 & 3528000 & 364000000 \end{bmatrix}} * \begin{bmatrix} a0 \\ a1 \\ a2 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 20 & 40 & 60 & 80 & 100 & 120 \\ 400 & 1600 & 3600 & 6400 & 10000 & 14400 \end{bmatrix} * \begin{bmatrix} 270 \\ 230 \\ 210 \\ 190 \\ 240 \\ 260 \end{bmatrix}}_{\begin{bmatrix} 1400 \\ 97600 \\ 8592000 \end{bmatrix}}$$

Solving the equation using the backslash operator in Matlab, we get,

$$z = \begin{bmatrix} a0 \\ a1 \\ a2 \end{bmatrix} = \begin{bmatrix} 334.000 \\ -3.682 \\ 0.026 \end{bmatrix}$$

Hence our polynomial approximation becomes $p_2(v) = 334.000 - 3.698\ v + 0.026\ v^2$. The plot with the measurements and the polynomial is presented below.

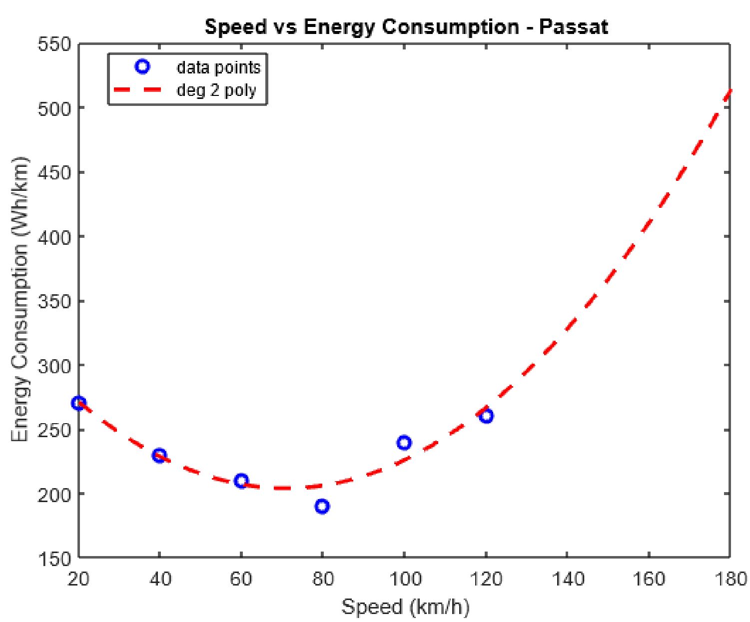

Figure 1 Speed vs Energy Consumption - Passat

b) The code of the Matlab script that performs the calculation in the previous task is presented below, along with the plot.

```
% miniproject_part1_script

v = [20;40;60;80;100;120];      % speed - km/hr
e = [270;230;210;190;240;260]; % energy consumption - Wh/km

% Fitting a 2nd degree polynomial using least squares method
A = (ones(size(v)));           % Defining the measurement matrix
A(:,end+1) = v;
A(:,end+1) = v.^2;
B = A'*A;
b = A'*e;
z = B\b;
z = round(z,3);             % To round the digits to 3
poly = flip(z');            % To use ‘polyfit’

% Plotting graph
x = linspace(20,180,400);
y = polyval(poly,x);
plot(v,e,'bo',LineWidth=2);
hold on;
plot(x,y,'r--',LineWidth=2);
hold off;
title('Speed vs Energy Consumption - Passat');
xlabel('Speed (km/h)');
ylabel('Energy Consumption (Wh/km)');
legend('data points','deg 2 poly',Location='best');
```

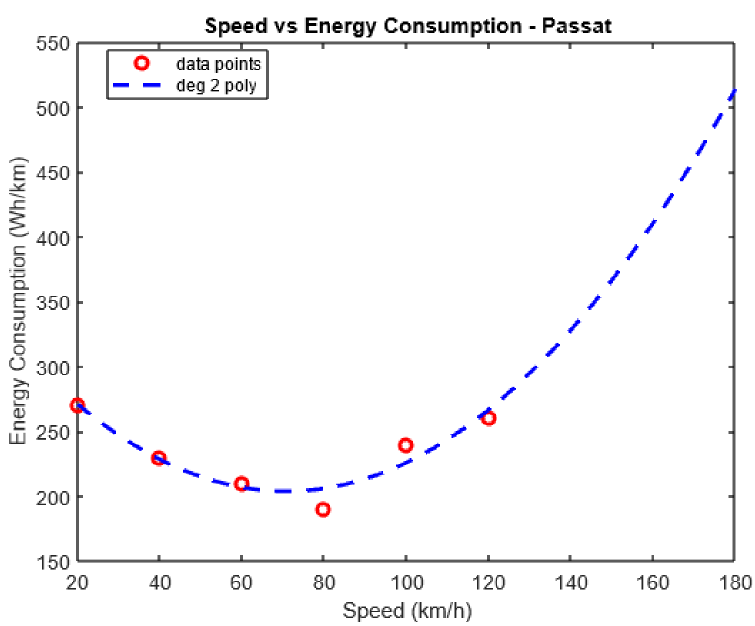

Figure 2 Speed vs Energy Consumption - Passat (Matlab)

c) Using the polyfit function in Matlab provides the second degree polynomial, which is the same as in tasks (a) and (b).

poly2 = [0.0259 -3.6821 334.0000]

d) The estimated energy consumption at 180 km/hr is 513.6400 Wh/km.

Part 2

a) The plot for power consumption as a function of speed.

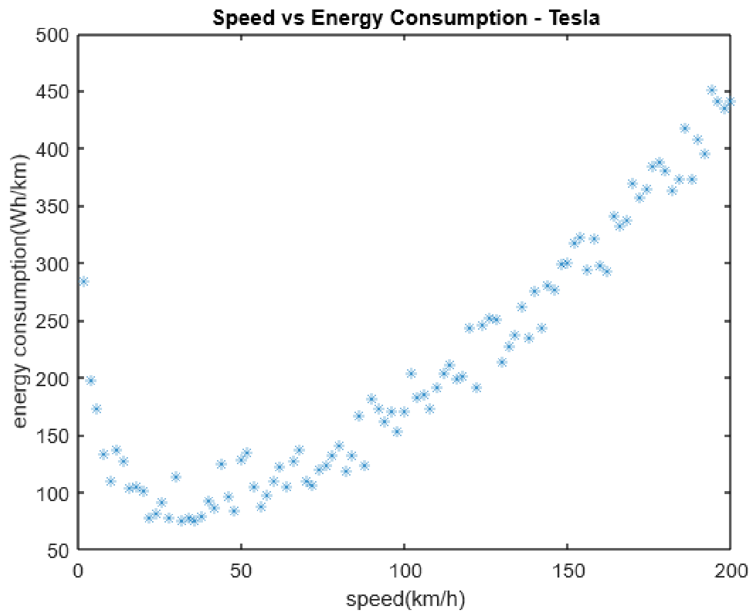

Figure 3 Speed vs Energy Consumption - Tesla

b)

```
% miniproject_part2_b
load elbilTesla.mat;          % Loading the dataset

% Fitting a 2nd degree polynomial using least squares method
A = (ones(size(speed_kmph)))';
A(:,end+1) = speed_kmph;
A(:,end+1) = speed_kmph.^2;
B = A'*A;
b = A'*consumption_Whpkm';
z = B\b;
poly_Tesla = flip(z');

% Curve fitting using 'polyfit' in Matlab
poly2Tesla= polyfit(speed_kmph,consumption_Whpkm,2);

% Plotting the graph
x = linspace(0,220,400);
y2 = polyval(polyTesla,x);
plot(speed_kmph,consumption_Whpkm,'b*' );
hold on;
plot(x,y2,'r');
title('Speed vs Energy Consumption - Tesla');
xlabel('Speed(km/h)');
ylabel('Energy Consumption(Wh/km)');
legend('data','deg2','Location','best');
```

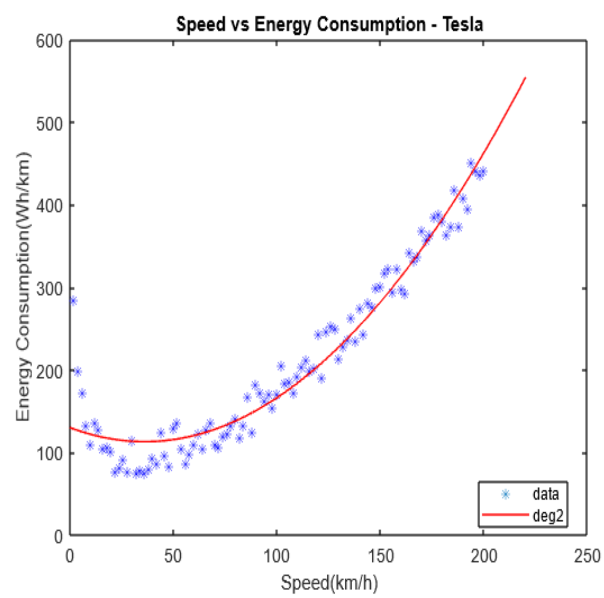

Figure 4 Speed vs Energy Consumption – Tesla (deg 2 polynomial)

c) Using 'polyfit' function in Matlab, polynomials of degree 2, 4, 8 and 12 were fit to the measurements to check the best approximation of the dataset. The plot of all the polynomials is also presented.

From the graph, it is seen that best fit for the dataset is a degree 4 polynomial. The polynomials of degree 8 and 12 seems to be a better fit for the data for lower speeds, however for speeds higher than 200 km/hr, both these polynomials increase exponentially, which is not realistic, and hence they cannot be used a good approximation for energy consumption.
The 4 degree polynomial is approximated is Matlab is poly4 = [0.000001, -0.000640, 0.112549, -6.473738, 203.86249]. Since the coefficient of the fourth degree variable is so small, the polynomial has very less dependency on the fourth degree and hence the possibility of a 3 degree polynomial being the best fit was checked. However, it was found that for a degree 3 polynomial, the energy consumption at higher speeds reduces, which is not the real case scenario and hence that possibility was rejected.

The worst approximation for the dataset is degree 12 polynomial. Although it provides very close approximations at lower speed, at higher speeds, the energy consumption is increasing exponentially, which is not the real case scenario.

Also, the condition number in approximating these polynomials can also be used to decide on a good fit for the dataset. For a degree 12 polynomial, since the condition number is significantly larger, errors will be amplified in the results. Small errors in measurement or input data will cause large errors in the approximation. This is true for higher degree polynomials. Hence higher degree polynomial approximations cannot be used for reliable approximations.

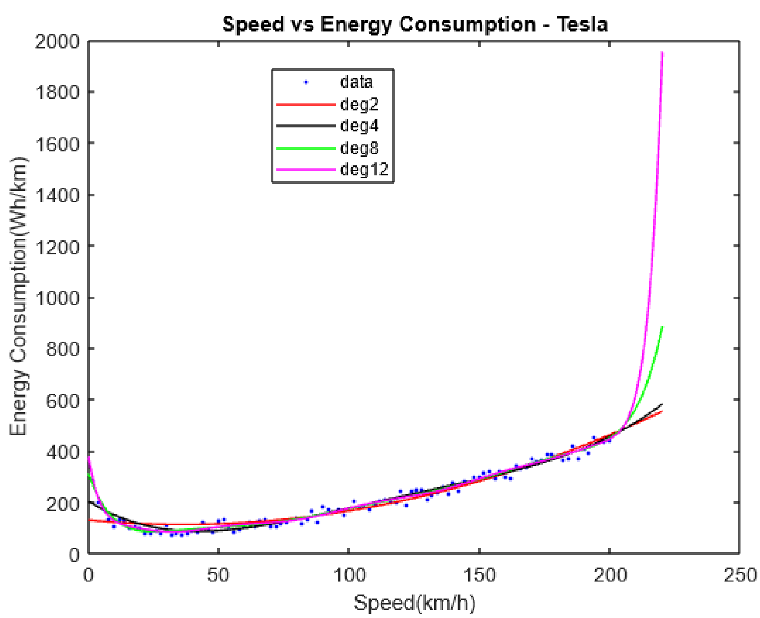

Figure 5 Speed vs Energy Consumption – Tesla (comparison)

d) The energy consumption approximation using 'splines' is presented below. Cubic piece-wise polynomial approximation is done to find the relation between speed and energy consumption.
A spline fits the dataset better than other methods due to 'piecewise' approximation. Hence this is a better approximation method than most others and very good for interpolation. But extrapolation or estimation for values outside the dataset is not very reliable using splines. In this example, as evident from the graph, it cannot be used to estimate the consumption at speeds higher than 200 km/hr, which is the maximum speed included in the dataset. For higher values, the plot is not giving realistic values.

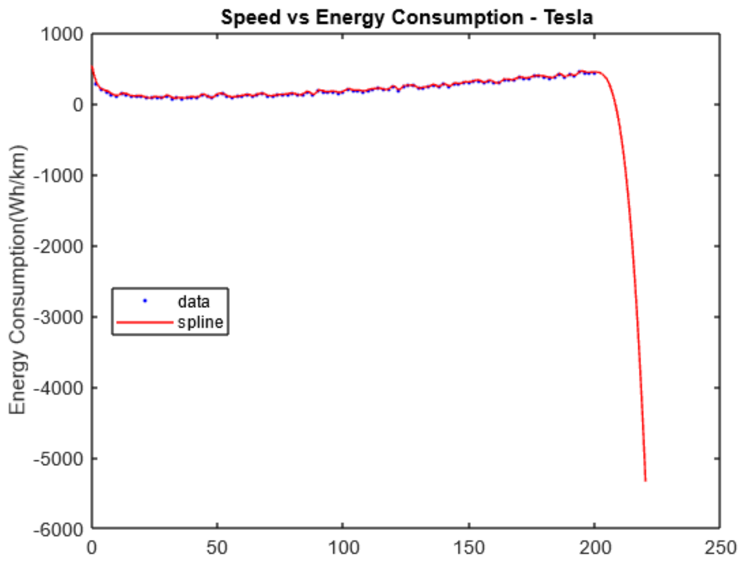

Figure 6 Speed vs Energy Consumption – Tesla (splines)

e) The plot of speed vs range or expected total mileage is presented below. Based on the plot, it can be concluded that Tesla has a higher range for all the speeds. Hence Tesla roadster can be driven the longest before battery recharge. The longest range or mileage for Passat is approximately 64 km at a constant speed of 72 km/hr. For Tesla roadster, the longest mileage is at around 630 km, at a constant speed of 42 km/hr.

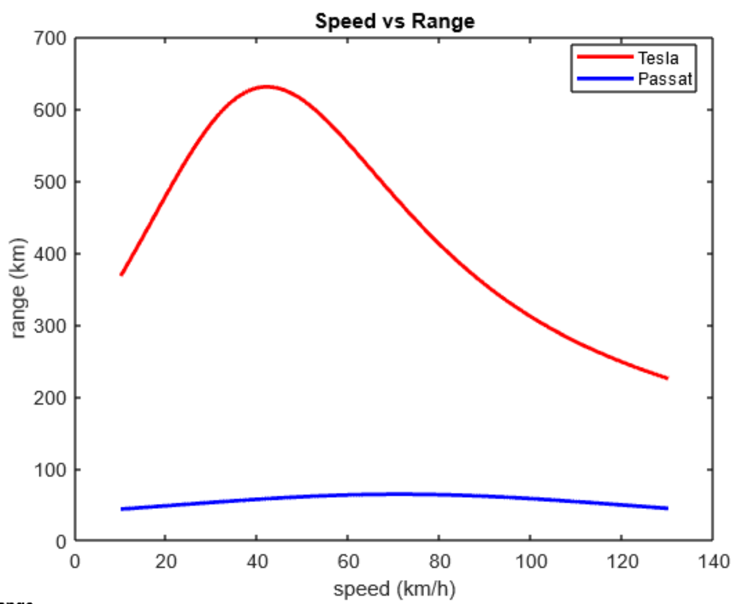

Figure 7 Speed vs Range

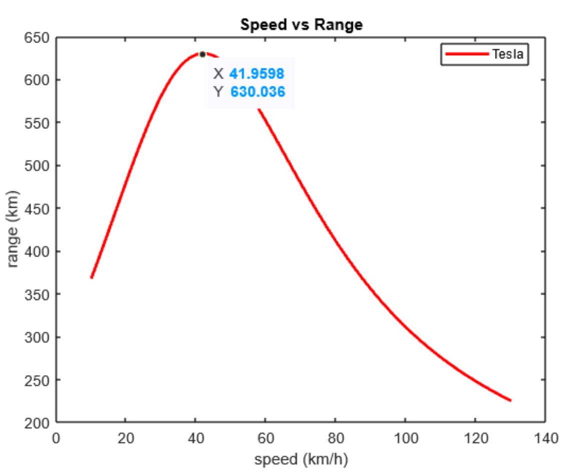

Figure 8 Speed vs Range - Tesla

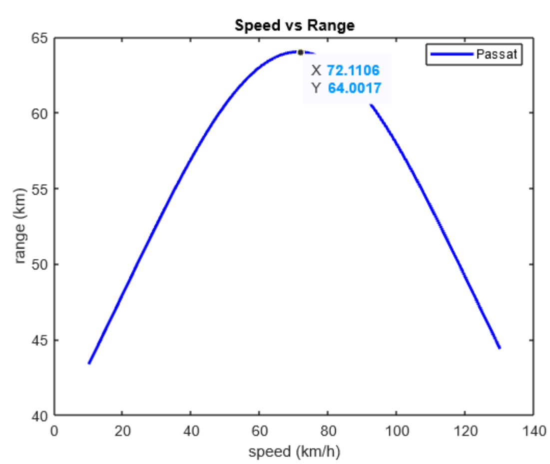

Figure 9 Speed vs Range - Passat

DISCUSSION

For the energy consumption approximation of Tesla roadster, polynomials of second and fourth degree were good fit for the data. However, polynomial of degree 4 was considered the best fit. This was because for the second-degree polynomial, the energy consumption at lower speeds were lesser than the actual energy consumption. At lesser speeds, normally we have more energy consumption due to initial ignition, and hence the second-degree polynomial was not a reliable fit for the data. The average energy consumption at steady state of the dataset is observed to be closer to the steady state energy consumption of degree 4 polynomial in this task. And the curve provided a good estimation of the overall data also.

Observing the Speed vs Range and Speed vs Energy Consumption graphs for both the models, we can easily conclude the obvious that maximum range is obtained when the energy consumption is lowest. For Tesla roadster, the energy consumption is lowest at around 40 km/hr speed as evident from the graph, and the maximum range is obtained at 42 km/hr. For Volkswagen Passat, the energy consumption is lowest at around 80 km/hr speed, and the maximum range is obtained at 72 km/hr.

APPENDIX

Part 1

```
% miniproject_part1
v = [20;40;60;80;100;120];    % speed - km/hr
e = [270;230;210;190;240;260];% energy consumption - Wh/km

% Fitting a 2nd degree polynomial using least squares method
A = (ones(size(v)));
A(:,end+1) = v;
A(:,end+1) = v.^2;
B = A'*A;
b = A'*e;
z = B\b;
z = round(z,3);
poly_passat = flip(z');

% Curve fitting using 'polyfit' function in Matlab
poly2_passat = polyfit(v,e,2);

% Plotting graph
x = linspace(20,180);
y = polyval(poly_passat,x);
plot(v,e,'ro',LineWidth=2);
hold on;
plot(x,y,'b--',LineWidth=2);
hold off;
title('Speed vs Energy Consumption - Passat');
xlabel('Speed (km/h)');
ylabel('Energy Consumption (Wh/km)');
legend('data points','deg 2 poly',Location='best');

%Energy consumption at speed = 180 km/hr
e180 = polyval(poly_passat,180);
```

Part 2

```
% miniproject_part2
load elbilTesla.mat;         % Loading the dataset

% Fitting a 2nd degree polynomial using least squares method
A = (ones(size(speed_kmph)))';
A(:,end+1) = speed_kmph;
A(:,end+1) = speed_kmph.^2;
B = A'*A;
b = A'*consumption_Whpkm';
z = B\b;
poly_Tesla = flip(z');

% Curve fitting using 'polyfit' function in Matlab
poly2Tesla = polyfit(speed_kmph,consumption_Whpkm,2);
%poly3Tesla = polyfit(speed_kmph,consumption_Whpkm,3);
poly4Tesla = polyfit(speed_kmph,consumption_Whpkm,4);
poly8Tesla = polyfit(speed_kmph,consumption_Whpkm,8);
poly12Tesla = polyfit(speed_kmph,consumption_Whpkm,12);

% Fitting a spline to the dataset
```

```
splineTesla = spline(speed_kmph,consumption_Whpkm);

% Plotting the graph for different curves
x = linspace(0,220,900);
y2 = polyval(poly2Tesla,x);
%y3 = polyval(poly3Tesla,x);
y4 = polyval(poly4Tesla,x);
y8 = polyval(poly8Tesla,x);
y12 = polyval(poly12Tesla,x);
yspline = ppval(splineTesla,x);

plot(speed_kmph,consumption_Whpkm,'b.',LineWidth=2 );
hold on;
plot(x,y2,'r');
plot(x,y4,'k');
plot(x,y8,'g');
plot(x,y12,'m');
%plot(x,yspline,'r');
hold off;
title('Speed vs Energy Consumption - Tesla');
xlabel('Speed(km/h)');
ylabel('Energy Consumption(Wh/km)');
legend('data','deg2','deg4','deg8','deg12','Location','best');
```

Voluntary Part

```
% miniproject_voluntary_part

% Using the polynomials computed in task 1 & 2
poly2_Passat = poly2_Passat;
poly4_Tesla = poly4Tesla
capacity_passat = 13000;    %unit - Wh
capacity_Tesla = 55000;     %unit - Wh

speed_range = linspace(10,130,200);

% Finding the range of the electric vehicle
% distance (km) = capacity (Wh) / consumption(Wh/km)
consumption_passat = polyval(poly2_Passat,speed_range);
distance_passat = capacity_passat./consumption_passat;
consumption_Tesla = polyval(poly4_Tesla,speed_range);
distance_Tesla = capacity_Tesla./consumption_Tesla;

%Plotting the graph
plot(speed_range,distance_Tesla,'r','LineWidth',2);
hold on;
plot(speed_range,distance_passat,'b','LineWidth',2);
hold off;
title('Speed vs Range');
legend('Tesla','Passat','Location','best');
xlabel('speed (km/h)');
ylabel('range (km)');
```