CLEMSON UNIVERSITY INTERNATIONAL CENTER FOR AUTOMOTIVE RESEARCH

# AuE 8810_Automotive Systems: An Integrated Overview

# Final Group Project: Designing a Battery-Electric Vehicle

## Faculty:

Dr. Chris Paredis

Dr. David Schmueser


Group Number: 26

| Student Name | CUID | mail id |
|---|---|---|
| Rishabh Saravgi | C18139475 | rsaravg@clemson.edu |
| Shripad Kadam | C14038213 | shripak@clemson.edu |
| Som Dixit | C20280955 | somd@clemson.edu |
| Siva Jyothi Karna | C11669485 | skarna@clemson.edu |


**Objective:** To design a Battery electric two-seater roadster vehicle.

**Tools Used:** MATLAB, Simulink and Siemens NX.

**Problem Statement:**

(1) To “design” an entire battery-electric roadster using the concepts learnt throughout the semester.

(2) To gain a deeper understanding of vehicle subsystems such as structures, packaging, vehicle dynamics, powertrain, human factors, and systems integration.

(3) To learn about model-based design and how to validate requirements using simulations

(4) To learn about subsystem coupling and how it affects the systems design process.

**Inputs provided to design the vehicle:**

Below parameters are given for design considerations.

• Gravitational acceleration $g$ = 9.81 $m/s2$

• Drag coefficient $Cd$ = 0.27

• $W101$ = 1650 $mm$

• $W103$ = 2020 $mm$

• Tire size = 265/35$R$19 • Density of air $\rho$=1.26 $kg/m3$

**Costs of components that make up the vehicle:**

• Misc. Electronics: $3000

• Wheel Assemblies: $450/corner

• Gearbox (single-ratio):

- $4000 for low-torque gearbox (torque<300Nm)
- $7500 for high-torque gearbox (torque>300Nm)

• Differential: $600

• Axles: $350 per half-shaft

• Motor & Inverter (see below for motor and inverter information)

• Body: cost is proportional to bounding-box volume: 1000$/m3

• Frame rail: cost is proportional to mass: $2/kg

• Battery: cost is proportional to capacity: $145/kWh

• Prop shaft: cost is proportional to length: $600/m

**List of masses that make up the total vehicle mass:**

• 4 AM95 passengers (101kg/passenger)

• Cargo per EPA (7kg/passenger)

• Infotainment (15kg)

• Sunroof (10kg)

• Unsprung masses (60kg/corner)

• Gearbox (single-ratio):

- 28kg for low-torque gearbox (torque<300Nm)

- 45kg for high-torque gearbox (torque>300Nm)

• Differential (10kg)

• Axles (5kg per half shaft)

• Body (Length X height x 50kg/m2)

• Frame rail: (to simply account for the cross members, assume the total weight of the frame to be 3 times the weight of the frame rail in the side view) -- Steel: 7800kg/m3

• If selected, prop shaft (10kg/m)

• Battery (160Wh/kg)

• Motor+Inverter (see below for motor and inverter information)

**List of volumes that need to be packaged**

(Length x height — 2D layout is considered. CG is assumed in middle of volume)

• Passengers (1 row of 2)

• Misc. Electronics: negligible (0m x 0m)

• Wheel Assemblies: bounded by wheel diameter

• Gearbox (single speed):

- 0.25m x 0.4m for low-torque gearbox (torque<300Nm continuous)
- 0.3m x 0.5m for high-torque gearbox (torque>300Nm continuous)

• Differential:

- 0.24m x 0.24 if separate
- 0m x 0m if on same axle and integrated into transmission

• Driveline:

- motor-gearbox connection can be transverse/longitudinal
- Two motors may be coupled to a single gearbox (doubles the torque)
- Center-center distance between input and output flanges of gearbox is fixed to 250mm

• Frame rail (as per your design — runs the length of the car)

• Battery: 200Wh/liter (max battery width: 1250mm)

• Prop shaft: 0.075m x designed length

• Motor + Inverter (see below for motor and inverter information)

<u>**Task 1: Get oriented and get organized**</u>

The suitable measures for new vehicle development are most often a comparison of the new concept/ design against already existing one which is referred to as a baseline design. In this case, we have taken ***1992 Honda NSX*** as the benchmark/ baseline vehicle for designing our vehicle. Basically, to have a good representation of the new concept and to avoid the misleading inferences, the comparison with baseline design is done, all the design variables are refined and optimized accordingly.

1. To solve the constrained optimization problem the below list of objectives, performance measures, design variables and parameters are needed.
2. **Objective**:
3. To adopt a value-driven design perspective and maximize profit of 1 billion USD for the enterprise by achieving better vehicle than our competitors.
4. **Vehicle measures of performance**: To attain the maximum performance of the vehicle, following factors are considered.
5. ***0-100 acceleration time*** – The acceleration time is expected to be minimum, but it should not have any impact on the other factors like battery range, top speed and cost.
6. ***Range on one battery charge*** – Maximum battery range will give maximum profit but it also increases the weight of the battery which indirectly affects our acceleration, top speed and cost of the vehicle.
7. ***Top speed*** – If we want high torque, we might require additional motors and batteries which again increases the cost of the vehicle.
8. ***Sales Price*** – We assume the vehicle to have at least 1 billion USD profit. As our competitor has put a sales price of $40000 USD, to gain profit and customers we need to undercut the price of our vehicle. We plan to achieve this by reducing the cost as much as possible while keeping the performance measures intact.
9.
10. **Design variables:** Below list of design variables are used
11. ***Battery capacity in [kWh]***
12. ***Choice of electric motor & inverter pair: A, B, or C (see below for motor characteristics)***

*Motor-Inverter A (UQM Power Phase HD250)*

- ***Motor and driveline configuration: RWD 1 motor, RWD 2 motors, AWD 2 motors, AWD 4 motors***
- ***Final drive ratio***
- ***2-D vehicle layout (multiple dimensions are needed)***
- ***Frame rail***
- ***Suspension spring constant in [N/m]***
- ***Suspension damping coefficient in [Ns/m]***
- ***Stability***
-
-
- **Intermediate variables:**

Following are the intermediate variables taken into consideration for various above design variables.

For Battery capacity - SOC, DOD, internal resistance, VOC, Vt, I_demand,

For Electric motor - Torque and Power map, Energy consumption, Efficiency, Forces,

For Final Drive ratio - Torque maps,

For Material - stress vs strain, maximum deflection, max load, design steps, angle of view, Cg, entry egress and ingress height,

For Frame rail - stress vs strain, maximum deflection, max load, For Suspension - Unsprung to sprung mass ratio, damping Coeff. Spring Coeff., handling test, un/sprung mass trade off, (spring/damping/(un/sm)) constants, natural frequency, RMS acc., (Yaw rate, acceleration, curvature, body slip angle, lateral velocity response), Static margin, Mimura plot

- **Parameters/ Variables which have fixed values in the design considerations:**

| Variables | Values |
|---|---|
| min req tractive force | |
| yield stress | <215 MPA |
| Ground Clearance | >120mm |
| battery to passenger | >10mm |
| wheelbase | = 2600mm |
| overall width of the vehicle | < 2020mm |
| track width | = 1650mm |

2.Determine each variable and how it is computed based on the inputs and the assumptions considered.

**Battery characteristics**: The battery consists of Li-ion cells, with 17 Wh of energy capacity. To arrive at the desired nominal voltage of 700V, 192 cells must be used in series (typically they are packaged in modules of 24, which makes for 8 modules). The total number of cells in the battery must therefore be a multiple of 192. The useful energy is 90% of the total capacity. 10% of the useful energy is required for cooling (battery, motors and inverters). The maximum discharge rate is 2C continuous and 5C peak – that is for a battery of X kWh in capacity, the maximum discharge rate is 2X kW for large-duration use and 5X kW for short duration use such as a 0-100 km/h acceleration test.

***Battery capacity in [kWh]***

For sizing of the battery, the vehicle is equipped with one motor 'A' and its inverter and is run over the drive cycle US06, the energy requirement of the drive cycle is calculated by integrating the power required for the vehicle with motor 'A' to run over drive cycle US06.

The energy requirement for US06 is 2.0438kW over a distance of 12812.7 m or 12.812 km, i.e., **159.5171Wh/km**.

Now that we have the energy requirement for on a per kilometer basis we can set a desired range, in our case we chose to go as soon as the top limit keeping the battery stacks of 192 intact. We finally got the number of stacks in series and number of stacks (192 cells) in parallel,

Number of stacks in series: 1

Number of stacks in parallel: 14

Capacity for 14 stack was calculated based on per cell capacity and number of cells in 14 stacks of battery, **192*14*17Wh= 45.696kWh**. which is giving us a range of approx. 312 km. For the capacity rating of 2C and 5C, since the motor will be requiring the discharge of 5C during 0-100km time test or during heavy acceleration, the max power battery can give is saturated by the top value of 5*45.696kW, if the power was saturated at 2C the motor could not have extracted the discharge current required to operate at 250kW peak power.

- ***Choice of electric motor & inverter pair: A, B, or C (see below for motor characteristics)***

For electric motor sizing we must decide the motor on its ability to fulfill the torque requirement of the drive cycle, mass, dimensions, peak, and continuous power.

For our vehicle, we started implementation of single motor configuration for Motor A, B and C. Out of all motors, only motor A was able to satisfy the Torque requirement of the US06 drive cycle.

Next to motor A was Motor C which lacked in area of low-speed high torque requirement. We have tried multiple iterations by tuning the PID controller and changing the values for the final drive ratio to match and fulfill the high torque requirement at the lower speed. But changing the final drive ratio did not fulfill our requirement and after an extent the speed requirement of the drive cycle was not being fulfilled as the final drive ratio was too high.

For Motor B the vehicle speed was not matching the drive cycle speed at any way, we tried to tune the PID and the final drive ratio, in the end we dropped the motor B and motor C and went with Motor A as the primary powertrain. Also, using singular motor A rather than multiple motor C to achieve the same results was a lot cheaper as the using a dual configuration for Motor C was costing the customer about 11000$ and a single motor A is 7500$.

***Motor-Inverter A (UQM Power Phase HD250)***

Motor mass: 85 kg

Inverter mass: 35 kg

Power rating: 175 kW continuous, 250kW peak

Motor bounding-box dimensions: cylinder with 0.32m in length and 0.39 in diameter

Inverter bounding-box dimensions: 0.531m x .422m x 0.175m (can be mounted in any direction)

Cost: $7500

Efficiency map: (maximum speed = 5500 rpm)

- ***Motor and driveline configuration: RWD 1 motor, RWD 2 motors, AWD 2 motors, AWD 4 motors:***
- For the configuration of the driveline with Motor A we went with rear wheel-drive with a high torque gearbox (>300Nm), the gearbox will be providing us with the final drive ratio of “fd=5” and directly transmitting the torque to the wheels using transaxle.
- 
- We chose the rear wheel-drive configuration because the maximum traction force considering longitudinal weight transfer during acceleration is more for a rear wheel-drive vehicle than a front wheel drive vehicle.
- 
- The higher traction limit for the vehicle would mean that my powertrain would be able to operate at a higher torque point at the best gear for acceleration without worrying about slipping of wheel and having to shift to next best gear ratio to regain traction.
- 
- Fitting the motor in front along with the inverter, gearbox and transaxle also raise the challenge when it comes to the packaging of the vehicle.
- 
- ***Final drive ratio:***

For the Final drive ratio, we firstly prioritized the factor that torque demand of the drive cycle should be met, so that the vehicle speed and desired speed are same till the 2 decimal places. For this, we found out the range of the final ratio in which we can play around, trading off top speed for the better 0-100kmph time while our torque demand is still being met. For Motor A this range was “fd=3.8” to “fd=10.5”, after this value the torque demand for the drive cycle was not being fulfilled for the lower end speed operation or the higher end speed operation for the lower ratio of final drive ratio and higher ratio of final drive respectively. We shortlisted the narrow range of final drive ratio of 5.0 to 5.7 trying to keep the specifications of our vehicle better than the competitor and also maximizing the profit as profit is directly affected from the 0-100kmph acceleration time and top speed. We were getting the highest profit for the “fd=5.0”.

- ***2-D vehicle layout (multiple dimensions are needed)*** – As we are given a task of designing a two-seater roadster and we had an idea of having the rear wheel-drive vehicle with the battery and motor at the rear. Hence, we need a rear wheel drive and rear engine as our base vehicle. So, we have chosen the 1992 Honda NSX. The following are the dimensions taken from our baseline vehicle.

| Dimensions | Description | Value | Unit |
|---|---|---|---|
| **L104** | Front overhang | **788.8** | mm |
| **L105** | Rear Overhang | **907.3** | mm |
| **L108** | Overall vehicle length | **4296.1** | mm |
| **H101** | Overall height from the ground | **1248.3** | mm |
| **H5-1** | SGRP to the ground distance | **407.5** | mm |

- ***Frame rail:*** Compared to the standard specifications for frame rail used in automobiles, the following values were chosen.

| Frame Dimensions | Value | Units |
|---|---|---|
| Height | 100 | mm |
| Width | 35 | mm |
| Thickness | 5 | mm |

- 
- ***Suspension spring constant in [N/m] :*** The standard value of suspension spring constant is usually less than 100,000 [N/m] for an automobile like roadster. So, we assume this to be less than 50,000 [N/m]
- ***Suspension damping coefficient in [Ns/m]:***The standard value of suspension damping constant is usually less than 10,000 [Ns/m] for an automobile like roadster. So we assume this to be 10000 [Ns/m]
- 
- ***Stability:*** Stability depends on the suspension spring and damping coefficient. If the values are low, the suspension of the vehicle will become soft, and it will directly affect

the handling characteristics under high speed. If the suspension is too stiff, then the ride comfort would be compromised. Yaw, roll, and pitch are dependent on the spring constant and damping coefficient which directly affects the stability of the vehicle.

- 
- The order in which these models are computed to avoid the need for iteration.
- 
- 2-D Modelling was done using CAD software – Siemens NX considering the benchmark vehicle dimensions.
- Powertrain elements selection i.e., battery (capacity, dimensions), motor (selection of type), and gearbox (selection of type).
- Addition and placement of unsprung masses, placement of motor, battery, gearbox, invertor across the length of the frame.
- Finalizing the frame rail dimensions by doing Shear force and bending moment calculations using MATLAB.
- From the above computations, Overall mass of the vehicle and weight distribution can be obtained.
- Calculation of spring constant and damping coefficient was done using Vehicle Dynamics principle.
- Calculation of profit model.

**<u>Task 2: Create the models to evaluate each of the requirements and objective</u>**

1. Provide a plan for the implementation of each model. Consider how will you approach the model and how you will test each model.
2. 2D Design of Vehicle outline –

Iteration 1:

- Length of the overhangs were not adjusted according to the aspect ratio which is used in the sports car or roadster design. That is why the overhang seems to be on the longer side in iteration 1.
- As the overhangs were not in the aspect ratio the approach angle and departure angle limit of 12 deg. Was not satisfied and it intersects the frame rail.
- The incorporation of vision angle needs improvement in the iteration 1 as it was not tangent to the hood-line, so there was still place for improvement.
- Packaging and all the components were not able fit in the vehicle properly as it was not optimized according to the dimensions limit provided to us in the problem statement.

Iteration 2:

- According to Human Factors, the dimensions of the torso, thigh and shin length were taken according to the 95thpercentile male.
- Angle between the Torso, thigh and shin are adjusted according to the vehicle type i.e., roadster. Ergonomics are designed in such a way that driver and passenger will stay low to the ground and looks more like a sports car but also are in the comfort range for the passenger and the driver. We arrived at the optimized dimensions by interpolating the dimensions of the sports vehicle and saloon (luxury) vehicle.
- Dimensions of the overhangs were designed in such a way that it doesn't violate the aspect ratio of the car according to the class i.e., roadster (sports car). So, we had to

shorten the overhangs compared to iteration 1 which helped us to satisfy the approach angle and departure angle limit.

- Lower Vision angle was perfectly incorporated in the design where it is tangential to the hood.
- Packaging – All the powertrain components and luggage spaces are placed in such a way to satisfy all the constraints given in the problem statement.

**Vehicle Structures**: Standard specifications of the beams were analyzed from the vehicles of same class (roadster). By comparing the available data, we took the best suitable dimensions for the frame rail as below.

*SFBM Calculations:* To obtain the Shear force and Bending moment calculations, we have taken the point loads acting on the overhang beam. Two support reactions are assumed to be at the point where the wheels are placed.

*Deflection* has been calculated using the Double Integration method. Below is the formula.

**Vehicle Dynamics:** We have obtained the total mass of the vehicle from SFBM calculations from the frame rail and for the body by using overall dimensions (length, width, height) of the vehicle. We build upon the model that we did in vehicle dynamics assignment and used the value of resonant frequency given to us and calculated all the parameters successfully like spring coefficient, Damping coefficient, turning radius of the vehicle while meeting all the requirements like the weight distribution which is in the acceptable and range and designed our car to be understeering.

1. Describe each of the models. Include the MATLAB/Simulink code/screenshot of your scripts/models (and make sure the images are clear and have descriptions).

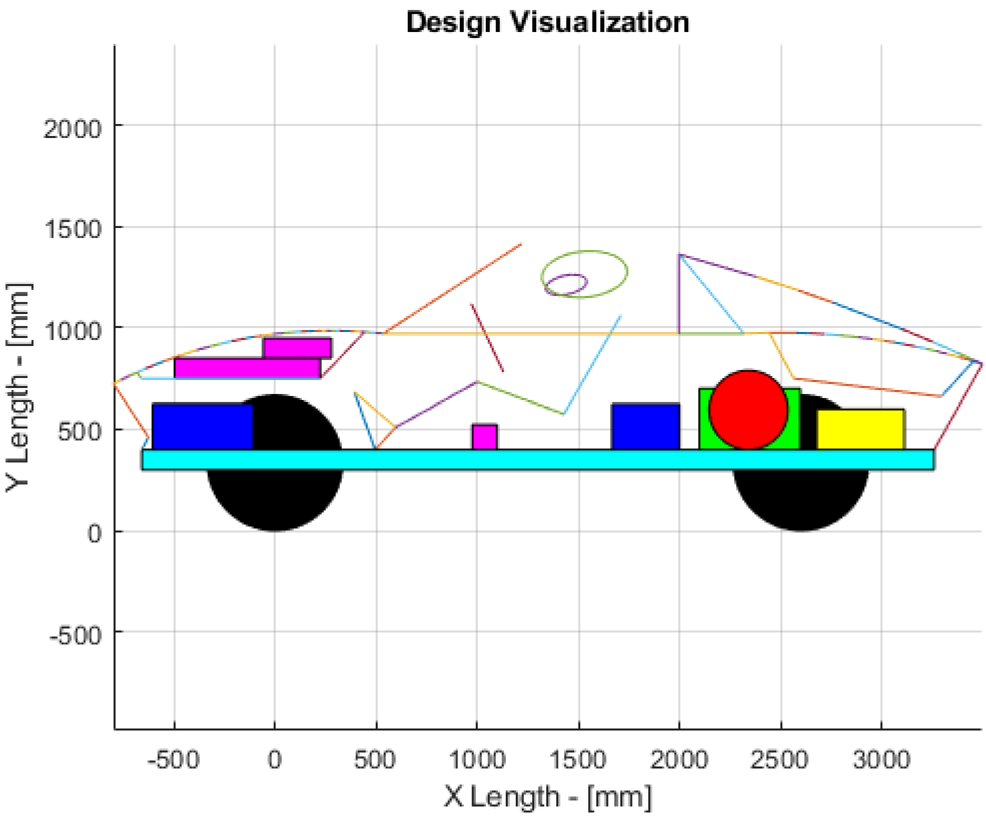
## Design Visualization

```
%% be careful using these commands... they really delete everything.
clear all;
close all;


%% Setting all the variable values -- per category
% requirements
tireRadius = 334.05; % in [mm] as per requirement
wheelBase = 2600;% in [mm] as per requirement
motor_radius = 195; %MotorA Dia


% design choices -- variables that can be chosen by the designer
frameRailLength = 3917.1; % in [mm]
frameRailHeight = 100; % in [mm]
H156 = 300; % in [mm]
```

```
% intermediate variable -- computed from design variables or requirements
frontAxle = [0,tireRadius]; % in [mm]
rearAxle = frontAxle + [wheelBase,0]; % follows from definition of wheelBase
middle = wheelBase/2;
front = middle - frameRailLength/2;
rear = front +  frameRailLength;


%% Plot the vehicle packaging
figure(1)
grid on
hold on
axis equal
plotCircle(frontAxle,tireRadius,'k');
plotCircle(rearAxle,tireRadius,'k')
%packaging of all components
plotRectangle([-605.15,626.1],[-105.15,400],'b')% Battery (60%)
plotRectangle([-496.75,748.5],[223.25,849.72],'m')% Luggage(72%)
plotRectangle([-57.05,849.72],[278.7,949.72],'m')% Luggage (28%)
plotRectangle([975.35,400],[1099.05,523.7],'m')% Cargo
plotRectangle([1664.55,400],[1998.75,625.6],'b')% Battery (40%)
plotRectangle([2097.15,400],[2597.15,700],'g')% Gearbox
plotCircle([2340.15,595],motor_radius,'r')% Motor
plotRectangle([2677.25,400],[3111.25,600.7],'y')% Inverter
% legend({'Front wheel' 'Rear wheel' 'Battery 60%' 'Luggage 72%' 'Luggage 28%'
'Cargo' 'Battery 40%' 'Gearbox' 'Motor' 'Inverter' })
xlabel('X Length - [mm]');
ylabel('Y Length - [mm]');
title('Design Visualization');
%95th percentile driver and passenger
plot([495.35,392.45],[400,683.7])% Acclerator
plot([495.35,596.75],[400,510.4])% shoe
plot([596.75,392.45],[510.4,683.7])% shoe
plot([596.75,1000.45],[510.4,733.4])% Shin
plot([1000.45,1426.95],[733.4,575])% Thigh
plot([1426.95,1708.45],[575,1062.6])% Torso
plot([1130.35,970.55],[781.2,1121])% Steering Wheel
%Body panels and otline of vehicle body.
plot([-658.55,-626.15],[400,461.5])
plot([-626.15,-798.65],[461.51,723.8])
plot([-798.65,-761.95],[723.8,742.5])
plot([-761.95,-717.65],[742.5,764.6])
plot([-717.6,-682.05],[764.6,780.4])
plot([-682.05,-639.45],[780.4,799.1])
plot([-639.45,-596.35],[799.1,817.1])
plot([-596.35,-549.04],[817.1,835.4])
plot([-549.04,-472.45],[835.4,863.1])
plot([-472.45,-388.05],[863.1,889.9])
plot([-388.05,-323.45],[889.9,908.2])
```

```
plot([-323.45,-265.15],[908.2,922.9])
plot([-265.15,-213.05],[922.9,934.7])
plot([-213.05,-150.05],[934.7,947.3])
plot([-150.05,-104.25],[947.3,956])
plot([-104.25,-62.55],[956,961.5])
plot([-62.55,-11.85],[961.5,969.1])
plot([-11.85,38.72],[969.1,974.5])
plot([38.72,94.35],[974.5,979.5])
plot([94.35,128.05],[979.5,981.9])
plot([128.05,159.85],[981.9,983.7])
plot([159.85,194.05],[983.7,985.1])
plot([194.05,229.75],[985.1,986.1])
plot([229.75,254.85],[986.1,986.5])
plot([254.85,276.25],[986.5,986.6])
plot([276.25,299.05],[986.6,986.4])
plot([299.05,318.45],[986.4,986.2])
plot([318.45,335.65],[986.2,985.7])
plot([335.65,369.05],[985.7,984.7])
plot([369.05,392.75],[984.7,983.6])
plot([392.75,425.55],[983.6,981.7])
plot([425.55,449.25],[981.7,980])
plot([449.25,474.35],[980,978])
plot([474.35,500.65],[978,975.6])
plot([500.65,523.15],[975.6,973.3])
plot([523.15,535.45],[973.3,972])
plot([535.45,1218.75],[972,1415.7])
plot([535.45,1995.45],[972,972])
plot([1995.45,1995.45],[972,1364.6])
plot([1995.45,2317.45],[972,972])
plot([1995.45,2317.45],[1364.6,972])
plot([3258.55,3497.45],[400,822])
plot([3497.45,3433.05],[822,841.4])
plot([3433.05,3383.45],[841.41,855.6])
plot([3383.45,3324.15],[855.6,871.6])
plot([3324.15,3270.15],[871.6,885.2])
plot([3270.15,3219.05],[885.2,897.1])
plot([3219.05,3166.25],[897.1,908.5])
plot([3166.05,3105.05],[908.5,920.6])
plot([3105.05,3057.85],[920.6,929.1])
plot([3057.85,3009.65],[929.1,937.1])
plot([3009.65,2951.85],[937.1,945.6])
plot([2951.85,2883.85],[945.6,954.4])
plot([2883.85,2789.45],[954.4,964.1])
plot([2789.45,2715.85],[964.1,969.8])
plot([2715.45,2640.95],[969.8,973.9])
plot([2640.95,2575.05],[973.9,976.1])
plot([2575.05,2442.75],[976.1,976.5])
plot([2442.75,2316.05],[976.5,972])
plot([3497.45,3447.75],[822,844.2])
plot([3447.75,3382.65],[844.2,873.6])
plot([3382.65,3251.15],[873.6,931.1])
```

```
plot([3251.15,3103.25],[931.1,992.9])
plot([3103.25,2757.25],[992.9,1126])
plot([2757.25,2586.1],[1126,1185.9])
plot([2586.1,2381.35],[1185.9,1252.7])
plot([2381.35,1994.75],[1252.7,1364.6])
plot([-682.05,-661.75],[780.4,749.7])
plot([-661.75,223.25],[749.7,749.7])
plot([223.25,440.65],[749.7,977.9])
plot([3452.65,3293.25],[835.5,661])
plot([3293.25,2562.45],[661,750.7])
plot([2562.45,2442.75],[750.7,976.5])


% Eye Ellipse
t = linspace(0,2*pi,100);
 theta = deg2rad(12);
 Major_radius=206.4/2;
 Minor_radius=93.4/2;
 x_coordinate = 1438.25;
 y_coordinate = 1213;
 x = x_coordinate + Major_radius*cos(t)*cos(theta) - Minor_radius*sin(t)*sin(theta);
 y = y_coordinate + Minor_radius*sin(t)*cos(theta) + Major_radius*cos(t)*sin(theta);
 plot(x,y)


%Head Ellipse
t = linspace(0,2*pi,100);
 theta = deg2rad(5.4);
 Major_radius=211.25;
 Minor_radius=113.50;
 x_coordinate = 1528.85;
 y_coordinate = 1265.6;
 x = x_coordinate + Major_radius*cos(t)*cos(theta) - Minor_radius*sin(t)*sin(theta);
 y = y_coordinate + Minor_radius*sin(t)*cos(theta) + Major_radius*cos(t)*sin(theta);
 plot(x,y)


% plot the frame rails
plotRectangle([front,H156+frameRailHeight],[rear,H156],'c')
```




## Powertrain Selection

**Powertrain selection code/model:**

**MATLAB code for calculating powertrain design variables:**

The code is well commented in order to explain how it works

```
load US06_Drive_Cycle.mat; %Drive cycle is loaded
%Declarig initial parameters which will be updated after the simulation.
alpha=0;
wd=0.46665; % from the front
L_wheelbase=2.600;
a=L_wheelbase*wd;
b=L_wheelbase-a;
mewf=1;
g=9.81;
R=0.668/2; %[m]
cd= 0.27;
fA= 1.530*1.1985; %[m2]
L=2.5;
mew_rr= 0.009; % rolling resistance
rho_air=1.26; %[kg/m3]
fd= 5; %Final drive ratio
eff_motor=1;
```

```
eff_transmission= 1;
eff_batt=0.9*0.9; %battery efficiency
%Cost of components, cost for few parts are calculated later as they depend
%upon changing factors
cost_of_axle=2*350;
cost_of_motor=7500;
cost_of_GB=7500;
cost_of_electronics=3000;
cost_of_wheels=(450*4);
cost_of_diff=600;
l=4.296;
h=1.198;
w=2.020;
%Cost of body calculation
bounding_box=(l*h*w);
cost_of_body=(bounding_box*1000);
disp(['cost of body is: ',  num2str(cost_of_body), '$']);
```

```
cost of body is: 10396.1482$
```

```
%mass , distance from start of the frame rail and height from ground for the
components, mass for few parts are calculated later as they depend
%upon changing factors
mass_of_frame_rail=38.2054; % mass for 2 frame rails
cost_of_frame_rail=(mass_of_frame_rail*2);
L_framerail=1.9592;
H_framerail=0.18253;
mass_of_body=(l*h*50);
L_body=2.008;
H_body=0.5516259304;
mass_of_frame=(mass_of_frame_rail*3);
L_frame=2.008;
H_frame=H_body;
mass_of_motor=85;
L_motor=2.9987;
H_motor=0.431;
mass_of_inverter=35;
L_inverter=3.5528;
H_inverter=0.3368;
mass_of_passenger=(101*2);
L_passenger=2.0855;
H_passenger=0.4078;
mass_of_cargo=(7*2);
L_cargo=1.6971;
H_cargo=0.2939;
mass_of_info=15;
L_info=1.2747;
H_info=0.803;
mass_of_sunroof=10;
L_sunroof=2.2260;
H_sunroof=1.1985;
mass_of_unsprung=60*4;
L_unsprung_front=0.6487;
```

```
L_unsprung_back=3.2487;
H_unsprung=0.334;
mass_of_GB=45;
L_gearbox=3.0057;
H_gearbox=0.386;
L_axle=3.2487;
H_axle=0.334;
mass_of_batt=285.6;
L_batt_60per=0.3034;
H_batt_60per=0.2432;
L_batt_40per=2.4902;
H_batt_40per=0.2429;
mass_of_load=100;
L_of_load_72per=0.5218;
L_of_load_28per=0.7205;
H_of_Load_72per=0.3997;
H_of_load_28per=0.4997;
mass_of_diff=0; % not used
%total mass of the vehicle
mass_of_vehicle=(mass_of_load+mass_of_batt+
mass_of_frame_rail+mass_of_body+mass_of_frame+mass_of_motor+mass_of_inverter+mass_of
_passenger+mass_of_cargo+mass_of_info+mass_of_sunroof+mass_of_unsprung+mass_of_GB);
disp(['mass of the vehicle is: ', num2str(mass_of_vehicle), ' kg']);
```

```
mass of the vehicle is: 1441.752 kg
```

```
%Distance of vehicle CG from the start of the frame rail
L_CG=(((mass_of_load*0.72)*L_of_load_72per)+((mass_of_load*0.28)*L_of_load_28per)+((
mass_of_batt*0.40)*L_batt_40per)+((mass_of_batt*0.60)*L_batt_60per)+((mass_of_unspru
ng*0.5)*L_unsprung_front)+((mass_of_unsprung*0.5)*L_unsprung_back)+(mass_of_cargo*L_
cargo)+(mass_of_motor*L_motor)+(mass_of_GB*L_gearbox)+(mass_of_inverter*L_inverter)+
(mass_of_body*L_body)+(mass_of_frame_rail*L_framerail)+(mass_of_frame*L_frame)+(mass
_of_sunroof*L_sunroof)+(mass_of_info*L_info)+(mass_of_passenger*L_passenger))/mass_o
f_vehicle;
disp(['Length of CG from the point start from the frame rail is: ',num2str(L_CG),'
m']);
```

```
Length of CG from the point start from the frame rail is: 1.862 m
```

```
%Distance of vehicle CG from the front wheel axle
L_CG_from_faxle=(L_CG-0.6487)
```

```
L_CG_from_faxle = 1.2133
```

```
%Weight distribution
weight_distribution=L_CG_from_faxle/L_wheelbase;
disp(['Wieght distribution is : ',num2str(weight_distribution)])
```

```
Wieght distribution is : 0.46665
```

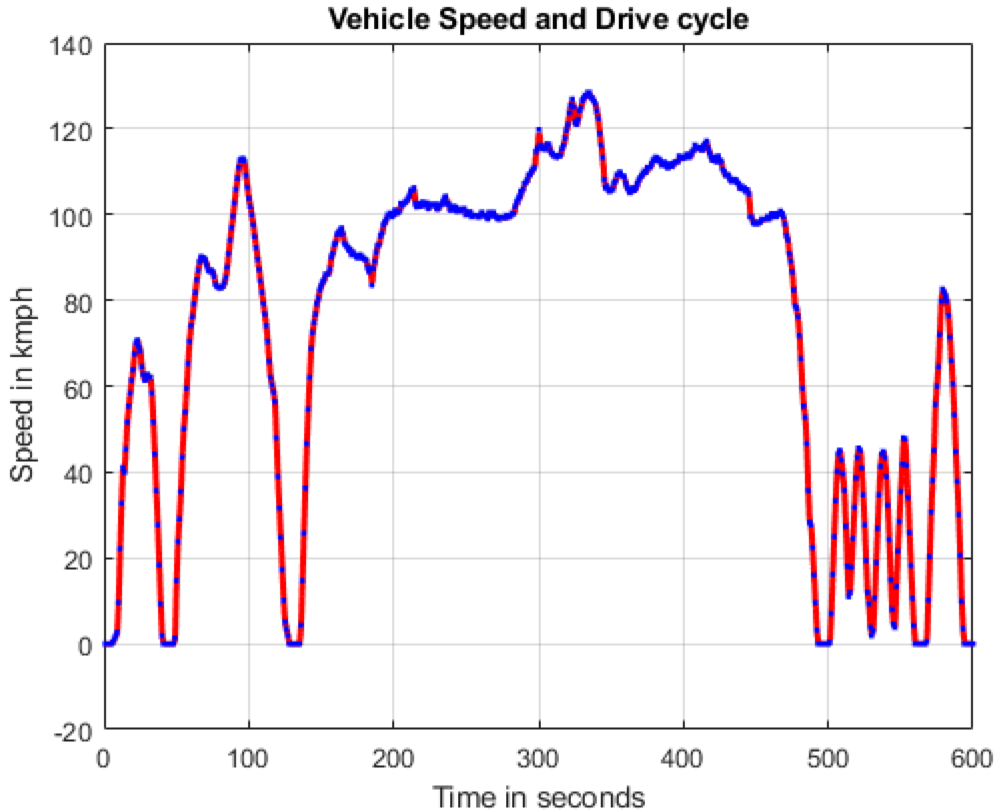
```
%height of the cg
H_CG=(((mass_of_load*0.72)*H_of_Load_72per)+((mass_of_load*0.28)*H_of_load_28per)+((
mass_of_batt*0.40)*H_batt_40per)+((mass_of_batt*0.60)*H_batt_60per)+((mass_of_unspru
ng)*H_unsprung)+(mass_of_cargo*H_cargo)+(mass_of_motor*H_motor)+(mass_of_GB*H_gearbo
x)+(mass_of_inverter*H_inverter)+(mass_of_body*H_body)+(mass_of_frame_rail*H_framera
il)+(mass_of_frame*H_frame)+(mass_of_sunroof*H_sunroof)+(mass_of_info*H_info)+(mass_
of_passenger*H_passenger))/mass_of_vehicle;
disp(['Hieght of CG is: ', num2str(H_CG), ' m'])
```

```
Hieght of CG is: 0.40285 m
```

```
%Calculating the max traction force for longitudnal load transfer
```

```
max_acc=(mewf*a*g)/(L_wheelbase-H_CG*mewf);
Max_trac=(mass_of_vehicle*((g*a)+(H_CG*max_acc))*mewf)/L_wheelbase;
disp(['Max traction force for the RWD vehicle is : ', num2str(Max_trac), ' N'])
Max traction force for the RWD vehicle is : 7810.2568 N
load MotorA_Data.mat
mass=mass_of_vehicle;
num_m=1; % Number of motors used
time=(0:0.1:50);
load US06_Drive_Cycle.mat;
figure;
 %kWh per motor
drive_cycle =  [t_cyc,v_cyc./3.6];
tsim=600;
task3= sim("task3_torque_model.slx",tsim); %Simulating Drive cycle
total_battery_energy=num_m*(task3.energy.signals.values(end,end))*(2.7778*10^-7); %
energy requirement for drive cycle, watt seconds to kWh.
disp(['Energy consumed over drive cycle US06: ', num2str(total_battery_energy), '
kWh']);
Energy consumed over drive cycle US06: 1.8745 kWh
Energy_consumed=(total_battery_energy*1000)/((task3.distance_cyc(end,end))/1000);
disp(['Energy consumed per km: ', num2str(Energy_consumed), ' Wh/km']);
Energy consumed per km: 146.2963 Wh/km
number_stacks=14;
per_cell_wh=17;
num_cell_module=192;
capacity=per_cell_wh*num_cell_module*number_stacks/1000; % calculating capacity of
the battery
cost_of_batt=(capacity*145);
disp(['cost of battery is: ',  num2str(cost_of_batt), '$']);
cost of battery is: 6625.92$
cost_of_vehicle=(cost_of_batt+cost_of_body+cost_of_wheels+cost_of_frame_rail+cost_of
_motor+cost_of_electronics+cost_of_axle+cost_of_GB);
disp(['cost of Vehicle is: ',  num2str(cost_of_vehicle), '$']);
cost of Vehicle is: 37598.479$
range=(num_m*capacity*1000/Energy_consumed);
disp(['Range: ', num2str(range), ' km']);
Range: 312.3523 km
mass_of_batt=(capacity*1000/160);
disp(['mass of the battery is: ', num2str(mass_of_batt), ' kg']);
mass of the battery is: 285.6 kg
batt_vol=(capacity*1000/200);
disp(['Volume of the battery is: ', num2str(batt_vol), ' l or ',
num2str(batt_vol/1000), ' m^3']);
Volume of the battery is: 228.48 l or 0.22848 m^3
figure;
plot(task3.tout,task3.velocityT3.signals.values, 'r','LineWidth',2); % Plotting
vehicle speed vs drive cycle profile
hold on
plot(t_cyc,v_cyc, 'b.','LineWidth',2);
ylabel('Speed in kmph');
xlabel('Time in seconds');
title('Vehicle Speed and Drive cycle');
```

```
grid on
hold off
```



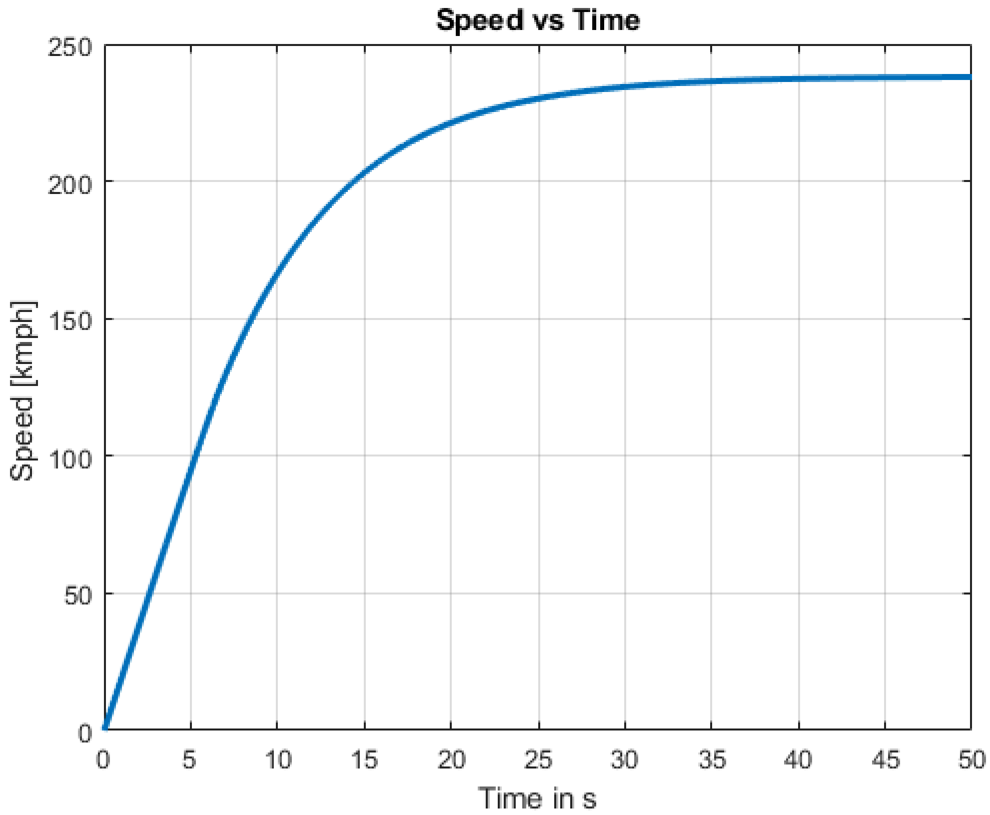

```
load MotorA_Data.mat
task2 =sim("Task22.slx",time); % Calculating the top speed and 0-100kmph time
for i=1:length(task2.tout)
    if task2.velocity(i)<=27.7778
        T0_100=task2.tout(i);
    end
end
top_speed=max(task2.velocity)*3.6;
disp(['Top speed of the vehicle is ', num2str(max(task2.velocity)), ' m/s', ' or ',
num2str(max(top_speed)), ' kmph']);
```

```
Top speed of the vehicle is 66.1061 m/s or 237.9819 kmph
```

```
disp(['Time for 0 to 100 kmph is ', num2str(T0_100), ' s']);
```

```
Time for 0 to 100 kmph is 5.2 s
```

```
figure(100)
plot(task2.tout,task2.velocity*3.6,'LineWidth',2);
ylabel('Speed [kmph] ')
xlabel('Time in s')
title('Speed vs Time')
ylim([0 250]);
grid on;
```



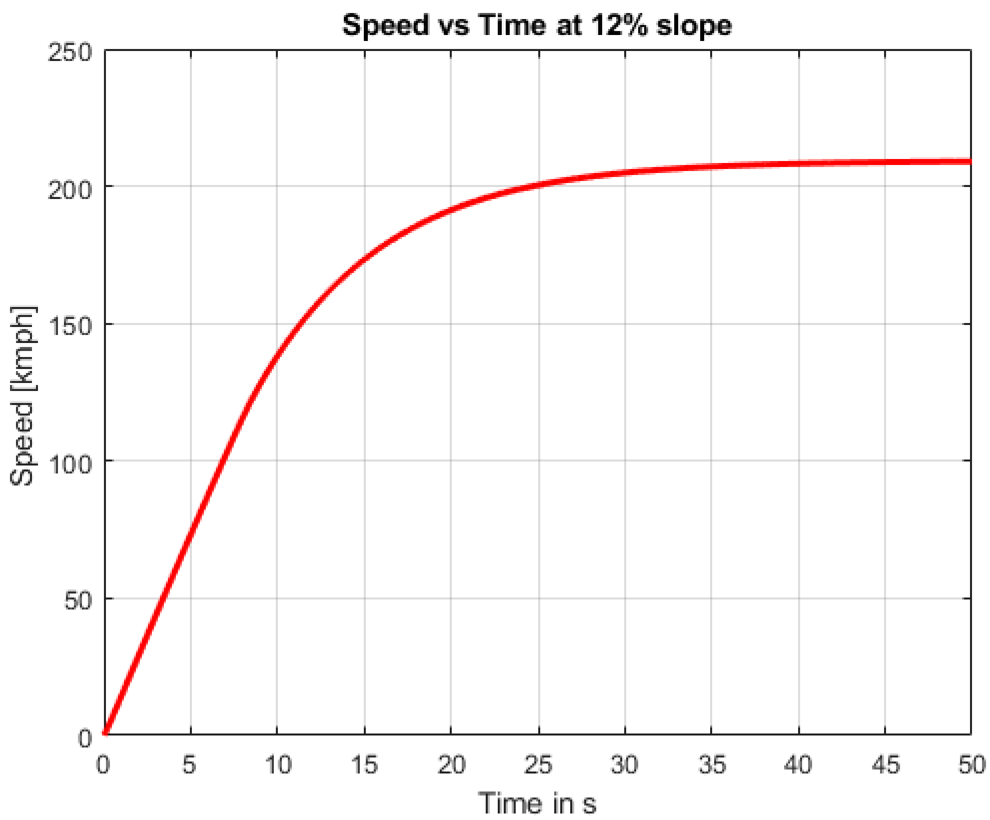

```
alpha=(6.84*pi)/180; %%% 12 percent slope
task2 =sim("Task22.slx",time); % Calculating 0-100kmph time and top speed for
vehicle at 12 % slope
for i=1:length(task2.tout)
    if task2.velocity(i)<=27.7778
        T0_100_slope=task2.tout(i);
    end
end
top_speed_slope=max(task2.velocity)*3.6;
disp(['Top speed of the vehicle on a slope is ', num2str(max(task2.velocity)), '
m/s', ' or ', num2str(max(top_speed_slope)), ' kmph']);
```

```
Top speed of the vehicle on a slope is 58.065 m/s or 209.0341 kmph
```

```
disp(['Time for 0 to 100 kmph on a slope is ', num2str(T0_100), ' s']);
```

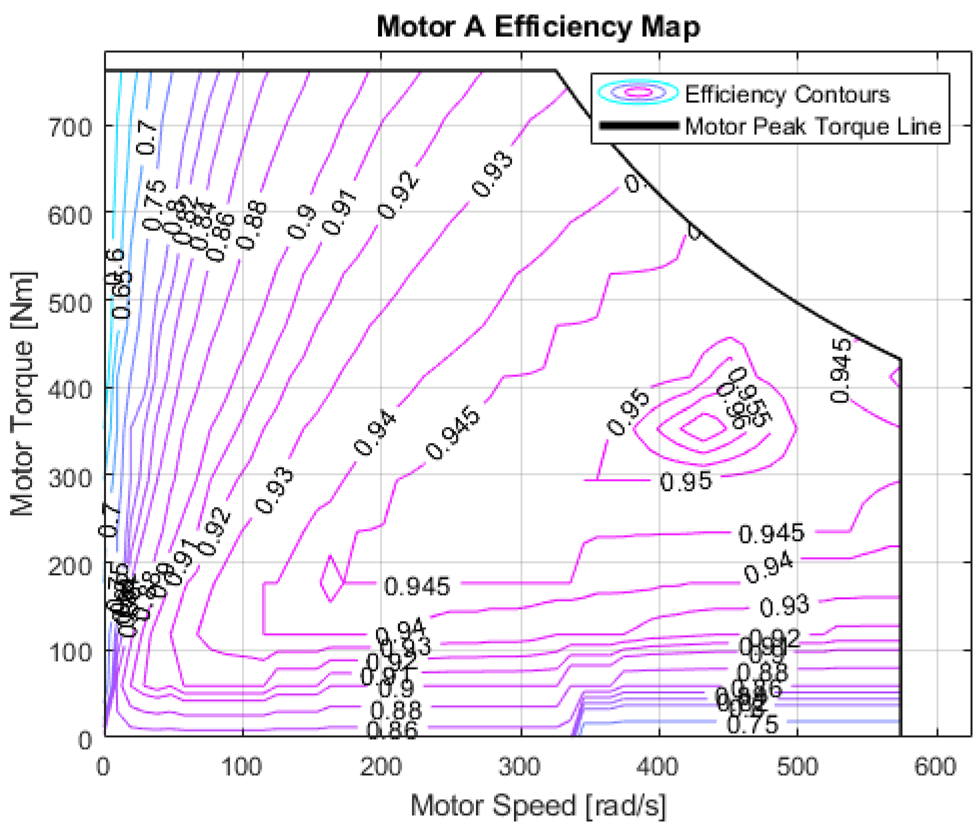
```
Time for 0 to 100 kmph on a slope is 5.2 s
```

```
figure(2)
plot(task2.tout,task2.velocity*3.6,'r','LineWidth',2);
ylabel('Speed [kmph] ')
xlabel('Time in s')
title('Speed vs Time at 12% slope')
ylim([0 250]);
grid on;
```




```
% the efficiency values for which to create contour lines
eta = [0.97 0.965 0.96 0.955 0.95 0.945 0.94 0.93 0.92 0.91 0.9 0.88 0.86 0.84 0.82
0.8 0.75 0.7 0.65 0.6 0.55 0.5];


figure (3)


% Create a contour plot of the efficiency map:
%   EM_Efficiency_Map is a 2D lookup table with efficiency values for given
%      rotational speed and torques corresponding to the scales below
%   EM_Omega_Map represents the speed in rad/s
%   EM_Toreque_Map represents the torque in Nm
[cc,h] = contour(EM_Omega_Map,EM_Torque_Map,EM_Efficiency,eta);
clabel(cc,h);
hold on
grid on


% The map extends beyond the rated range for the motor.  We therefore plot
% the maximum torque values and white out the portion of the map that is
% invalid
plot([EM_Omega_Max EM_Omega_Max(end)],[EM_Torque_Max 0],'k','linewidth',3)
colormap(cool)
hold off
large = 1e6;
```

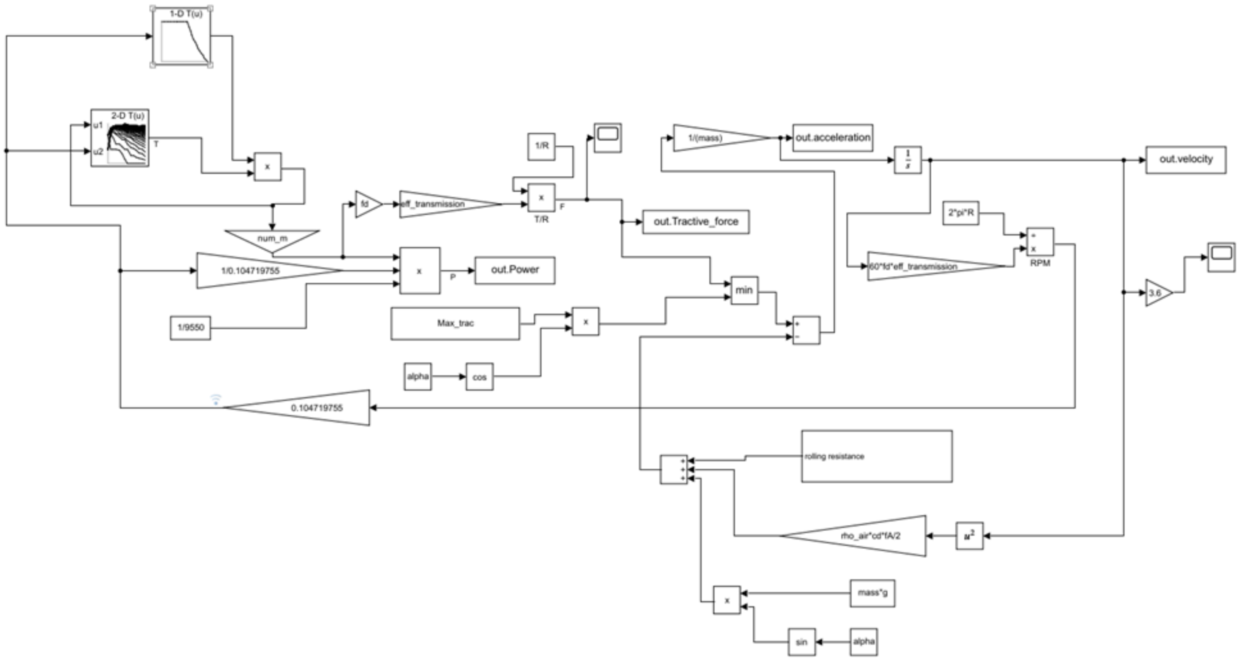
```
xp = [EM_Omega_Max EM_Omega_Max(end) large large 0];
yp = [EM_Torque_Max 0 0 large large];
patch(xp,yp,'white','Edgecolor','white');


% make it look good...
axis([0 max(EM_Omega_Map)+50 0 max(EM_Torque_Map)+20])
xlabel('Motor Speed [rad/s]')
ylabel('Motor Torque [Nm]')
title('Motor A Efficiency Map')
legend('Efficiency Contours','Motor Peak Torque Line','location','northeast')
```



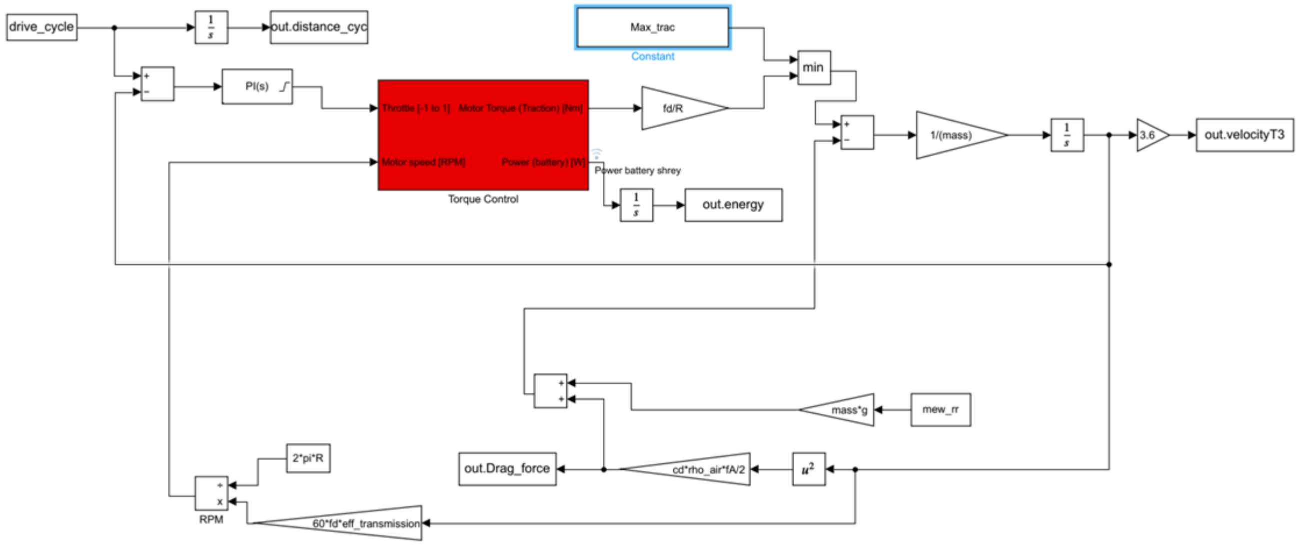

**Simulink Models:**

**Task22 (for calculating 0 – 100 time and top speed)**

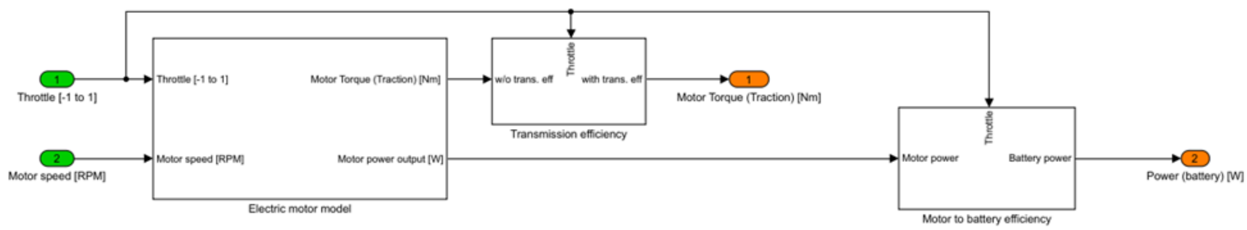
Task22.slx is used to calculate the top speed and 0-100kmph time of the vehicle for the given torque map and torque efficiency. The torque map inputs the rad/s value and gives out the torque value in Nm, the torque value is multiplied by the efficiency using a 2-D lookup table and is limited by the max torque value for that motor, the torque value is multiplied by the number of motors used (one) to get the total torque from the powertrain and is used to calculate the power output of the powertrain. The torque is amplified by the final drive ratio of 5.0 and transmission efficiency, assumed to be 1.0. The traction force is limited by the minimum function, comparing it with the maximum traction force of the system. The traction force is divided by the mass to give us the acceleration value, which is further integrated to give us the velocity, time is noted for when velocity reaches 100 kmph. The velocity is used to calculate the aerodynamics forces along with other components of road load. These losses are subtracted from the traction force. The Velocity is also used to calculate the rad/s of the motor at the point of operation, the same rad/s is used to calculate the motor torque and motor efficiency for the next time step, and the loop goes on for time duration of the drive cycle. The time duration could be reduced but is set to 600s to match with the drive cycle time just to keep everything consistent. PID is not used for throttle control as the simulation is operating at wide open throttle.

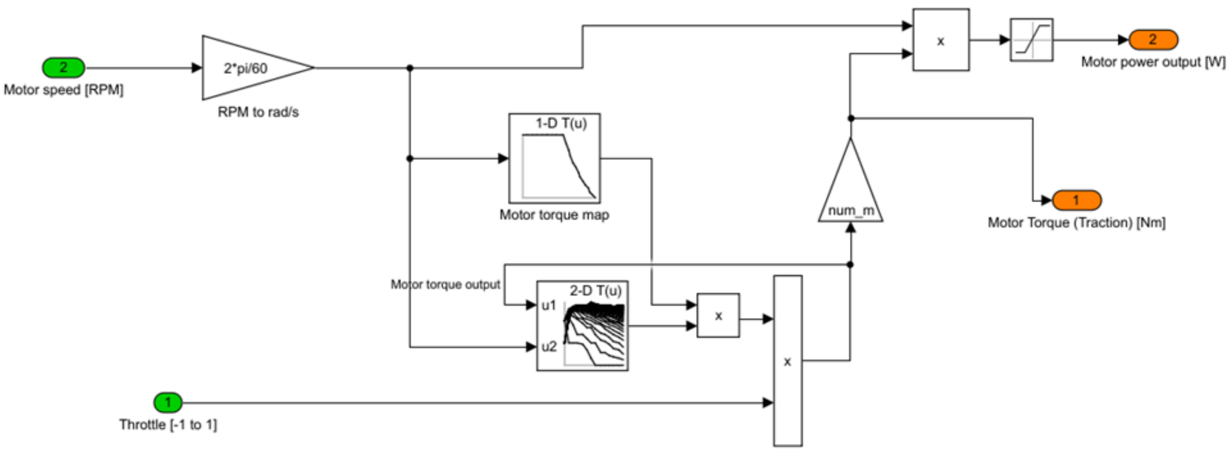



**Task3_torque_model.slx**simulates the vehicle designed over the drive cycle US06, the [time, velocity] double is used as an input for the drive cycle values, the velocity of drive cycle is integrated to give us the drive cycle distance covered, which will eventually help us give the energy required per km. A PID block is implemented and tuned because throttle is required for the simulation of drive cycle. Throttle value and motor speed (RPM) are the inputs of electric motor model in torque control subsystem where RPM is converted to rad/s and using 1-D lookup table and 2-D lookup table to calculate the torque value and efficiency of the motor respectively. The torque is multiplied by the efficiency and further by the number of motors used, Here the torque value is multiplied with the rad/s value to give us the power of the motor at that operating point, the Max power generated by the motor is limited by 5 times the capacity of the battery to ensure that discharge requirement is never above 5C. then the torque value is multiplied by the transmission efficiency to keep the transmission losses in account. The motor power is multiplied with the battery efficiency and switch function is used with input value as throttle to account for charging and discharging of the battery. Now, from motor power the energy required for the drive cycle is calculated by integration which will be used to calculate the energy per km which will further be a decider for the battery sizing of our vehicle. After this point the rest of the model is pretty similar to task22.slx where tractive force is used to calculate acceleration and velocity (used as input for PID) , further calculating the road load and RPM of the motor at the operating point and is used as the input for torque and efficiency table.



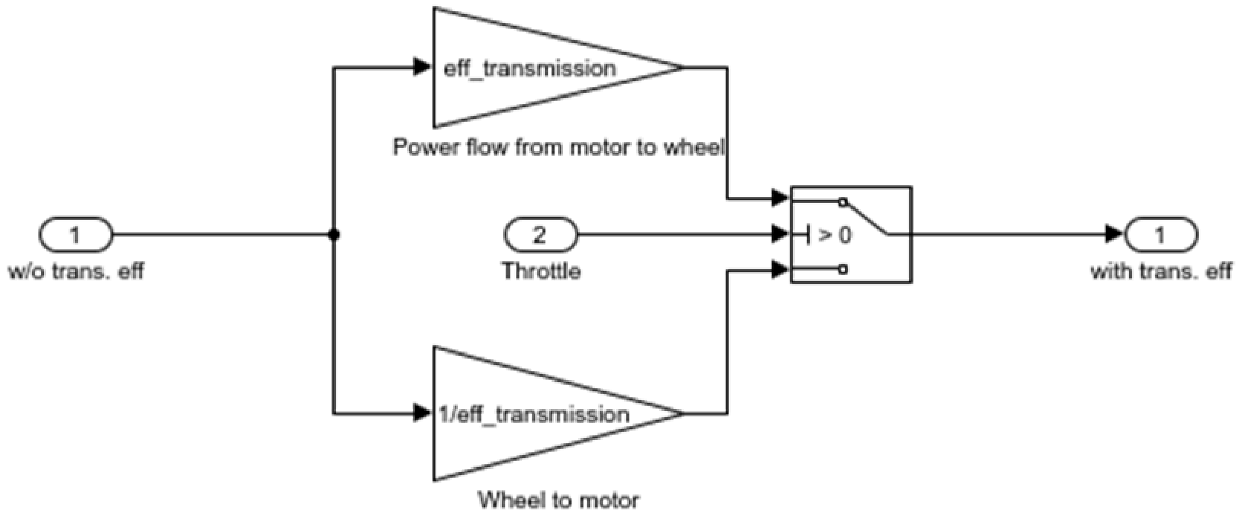

Electric Motor Model:



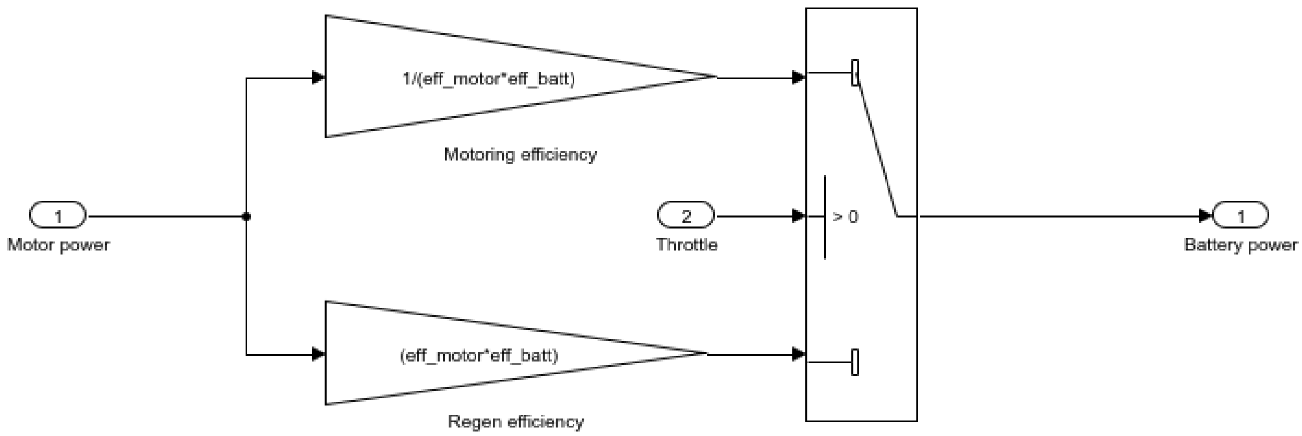



```
Transmission Efficiency
```

Ideally, this logic would be present in the transmission subsytem. To make it easier for you we have included the efficiency computation here. You only need to include the gear ratio calcualations in the transmission subsystem




```
Motor to Battery Efficiency
```




## SFBM Calculations

We selected the beam dimensions by comparing the roadster vehicles available in the market and got the range of dimension of beams and used them to design our own beam and reached at the set of dimensions of the beam which were acceptable while satisfying the stress, bending moment and deflection criteria given to us. We considered all the loads as point loads and applied on the beam at the specified locations on the frame rail and ran the SFBM code given to us. In which we get the FBD of the beam and Shear force and Bending moment diagram of the beam. To calculate deflection, we used the double integration method, in which we integrated the equation of maximum bending stress available to us at that specific point and after integration we apply the limits and divide the whole equation by the product of Young's Modulus (E) and Moment of inertia of the beam (I). By doing that, we got the feasible solution and got the deflection under 10mm.

**Vehicle Structures**: Standard specifications of the beams were analyzed from the vehicles of same class (roadster). By comparing the available data, we took the best suitable dimensions for the frame rail as below.

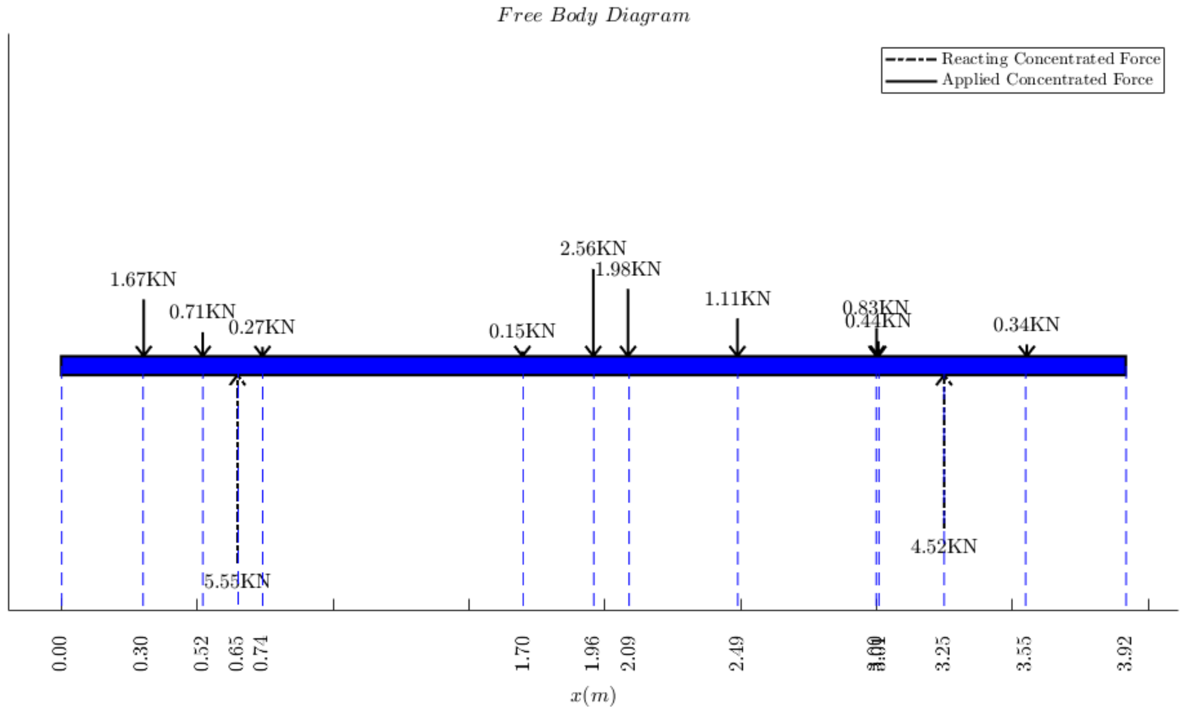
| Frame Dimensions | Value | Units |
|---|---|---|
| Height | 100 | mm |
| Width | 35 | mm |
| Thickness | 5 | mm |

```
Name1 = 'SFBM_Calcuations';
Cross_Area_Beam = 1500; % Square mm
E = 210000; % MPa
FrameRailLength = 3917.1 % mm
FrameRailLength = 3.9171e+03
% Length and Supports
LengthSupport1 = [3.9171,0.6487,3.2487];% length  = 3.9171m, supports at 0.6487m and 
3.2487m;


% Concetrated Loads
F1 = {'CF',-1.67,0.3034}; % Battery weight(60%)
F2 = {'CF',-0.706,0.5205}; % Luggage (72%)
F3 = {'CF',-0.2746,0.7405}; % Luggage (28%)
F4 = {'CF',-0.1471,1.6971}; % cargo
F5 = {'CF',-2.56,1.958} % Body
F5 = 1×3 cell
```

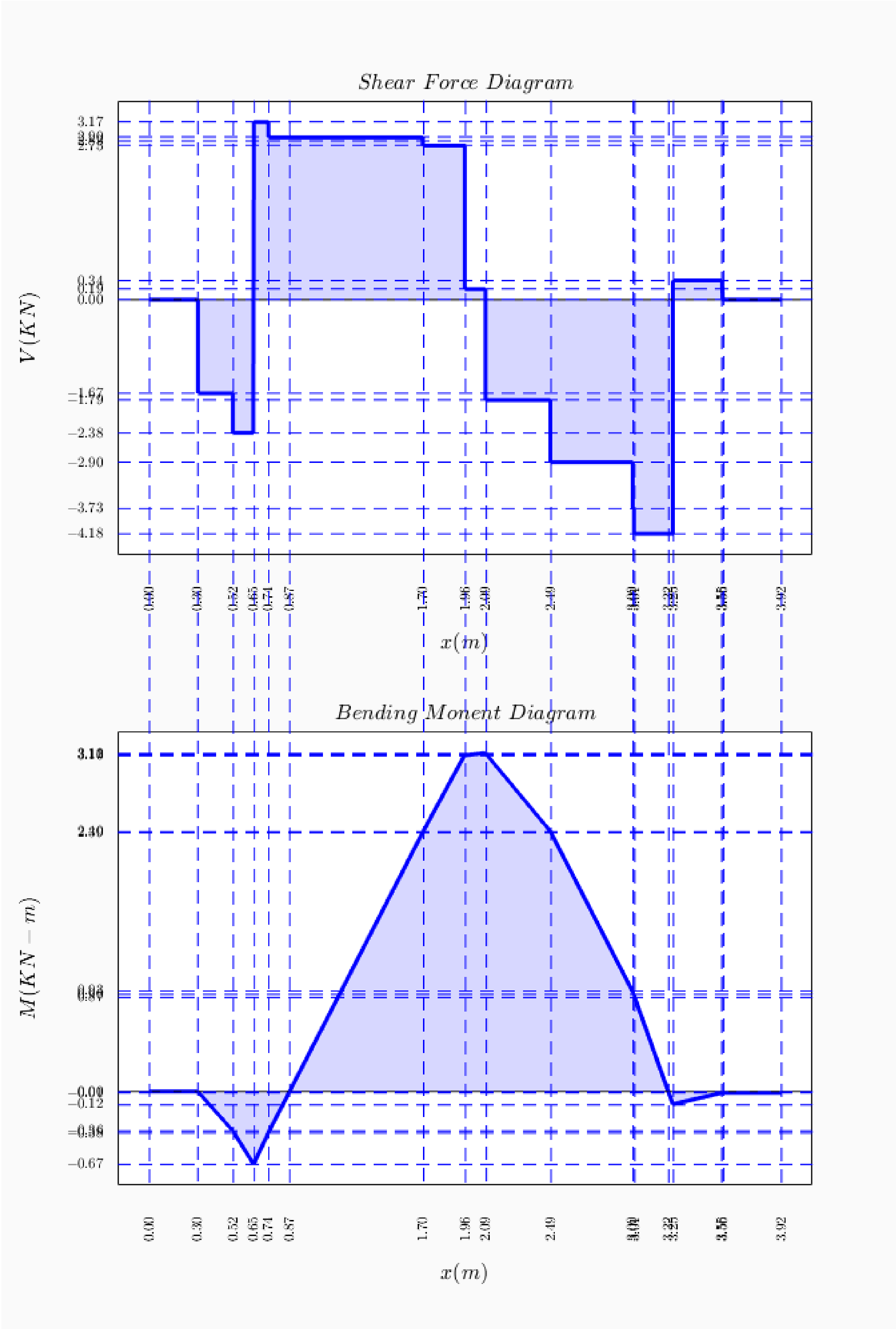
| | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 'CF' | -2.5600 | 1.9580 |

```
F6 = {'CF',-1.98,2.0855}; % Passenger & Driver
F7 = {'CF',-1.11,2.488}; % Battery (40%)
F8 = {'CF',-0.8338,3}; % Motor
F9 = {'CF',-0.4414,3.0067}; % Gearbox
F10 = {'CF',-0.3433,3.553}; %Inverter
```

```
SFBM(Name1,LengthSupport1,F1,F2,F3,F4,F5,F6,F7,F8,F9,F10);
```




| *Range* | *Equations of Shear Force :* | *Equations of Bending Moment :* |
|---|---|---|
| 0 *to* 0.3034 | 0 | 0 |
| 0.3034 *to* 0.5205 | $-1.67$ | $-1.67x + 0.504$ |
| 0.5205 *to* 0.6487 | $-2.376$ | $-2.376x + 0.869$ |
| 0.6487 *to* 0.7405 | 3.171 | $3.171x - 2.728$ |
| 0.7405 *to* 1.6971 | 2.897 | $2.897x - 2.524$ |
| 1.6971 *to* 1.958 | 2.75 | $2.75x - 2.275$ |
| 1.958 *to* 2.0855 | 0.19 | $0.19x + 2.729$ |
| 2.0855 *to* 2.488 | $-1.79$ | $-1.79x + 6.855$ |
| 2.488 *to* 3 | $-2.9$ | $-2.9x + 9.616$ |
| 3 *to* 3.0067 | $-3.734$ | $-3.734x + 12.115$ |
| 3.0067 *to* 3.2487 | $-4.176$ | $-4.176x + 13.441$ |
| 3.2487 *to* 3.553 | 0.343 | $0.343x - 1.233$ |
| 3.553 *to* 3.9171 | 0 | $-0.013$ |



Shear Force Diagram
V(KN)
3.17
0.00
−2.38
−2.90
−3.73
−4.18
x(m)
Bending Moment Diagram
M(KN − m)
2.30
−0.12
−0.67
0.00
0.30
0.52
0.65
0.71
0.87
1.70
1.96
2.09
2.49
3.92
x(m)

```
%Maximum Bending stress calculations
Maximum_Bending_moment = 3.13*10.^6;%Maximum Bending moment (KN-m)
B = 35;
D = 100;
b = 25;
d = 90;
I = ((B*D.^3)/12)-((b*d.^3)/12);
y = D/2;


Max_Bend_Stress = (Maximum_Bending_moment*y)/I;
fprintf('The Maximum Bending Stress is %f MPa',Max_Bend_Stress)
```

```
The Maximum Bending Stress is 111.952310 MPa
```

```
syms dist1
equation(dist1) = .221*dist1 + 2.695;
show(dist1) = int(int(equation));
deflection = double((show(1.96)-show(2.085))/E*I);


fprintf('The Deflection of the Beam is %f mm',deflection)
```

```
The Deflection of the Beam is -4.911665 mm
```

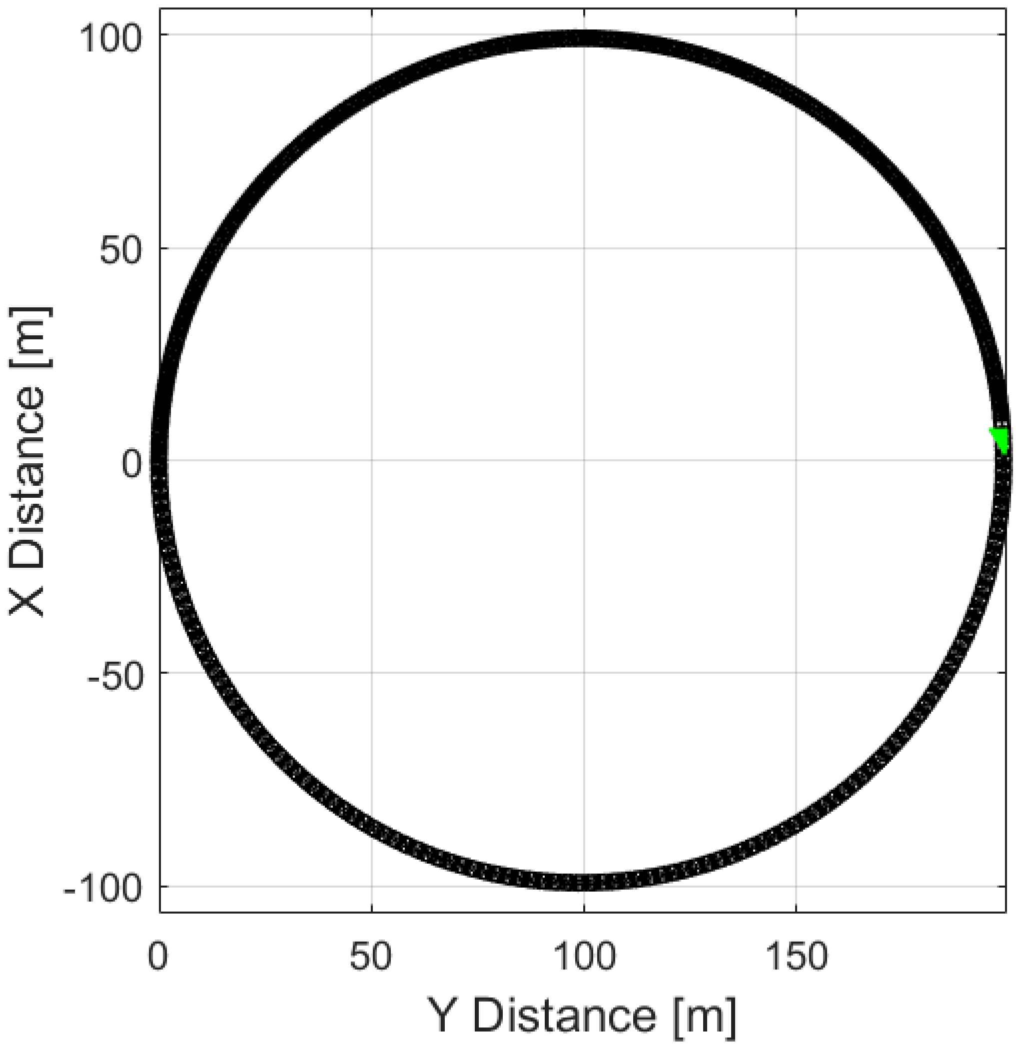
```
Fra_mass = (((B*D)-(b*d))*FrameRailLength) * 7800*10.^-9;


fprintf('The Mass of the Frame Rail is %f kg',Fra_mass)
```

```
The Mass of the Frame Rail is 38.191725 kg
```

## Vehicle Dynamics

```
%Simulink parameter definitions
% Define the vehicle parameters used in the Simulink model
mass = 1441.752;             % total vehicle mass [kg]
mass_unsprung=60*4;
mass_sprung=mass-mass_unsprung;
WD = 0.46665;              % weight distribution [%] -- proportion on rear axle
L = 2.600;              % wheelbase [m]
W= 1.530;
Jz = (1/12)*mass*((L^2)+(W^2));              % Moment of inertia of the vehicle
[kg.m^2]
g=9.81;
yaw_rate=0.282;
velocity=((0.8*9.81*100)^(1/2));
damping_ratio=0.3;
resonance_freq_front=1.2; %Hz
resonance_freq_rear=1.44;
```

```
nat_freq_front=resonance_freq_front/((1-2*damping_ratio^2));
disp(['Natural Frequency in front is: ', num2str(nat_freq_front),' Hz'])
```
```
Natural Frequency in front is: 1.4634 Hz
```
```
nat_freq_rear=resonance_freq_rear/((1-2*damping_ratio^2));
disp(['Natural Frequency in rear is: ', num2str(nat_freq_rear),' Hz'])
```
```
Natural Frequency in rear is: 1.7561 Hz
```
```
k_front=(((nat_freq_front*2*pi)^2)*mass_sprung*WD);
disp(['Spring Stiffness in front is: ', num2str(k_front),' N/m'])
```
```
Spring Stiffness in front is: 47413.3507 N/m
```
```
k_rear=(((nat_freq_rear*2*pi)^2)*mass_sprung*(1-WD));
disp(['Spring Stiffness in rear is: ', num2str(k_rear),' N/m'])
```
```
Spring Stiffness in rear is: 78034.0539 N/m
```
```
crit_damping_front=2*((mass_sprung*WD)*k_front)^(1/2);
crit_damping_rear=2*((mass_sprung*(1-WD))*k_rear)^(1/2);
damping_coefficient_front=damping_ratio*crit_damping_front;
damping_coefficient_rear=damping_ratio*crit_damping_rear;
disp(['Damping coefficient of front suspension is: ',
num2str(damping_coefficient_front),' Ns/m'])
```
```
Damping coefficient of front suspension is: 3093.8883 Ns/m
```
```
disp(['Damping coefficient of rear suspension is: ',
num2str(damping_coefficient_rear),' Ns/m'])
```
```
Damping coefficient of rear suspension is: 4243.331 Ns/m
```
```


% Derived quantities using the weight distribution of the vehicle
a = (1-WD)*L;           % front axle distance from CG [m]
b = WD*L ;              % rear axle distance from CG [m]


t1 = sim("AUE_881_HW4_2DOF_task1.slx", 100);
```

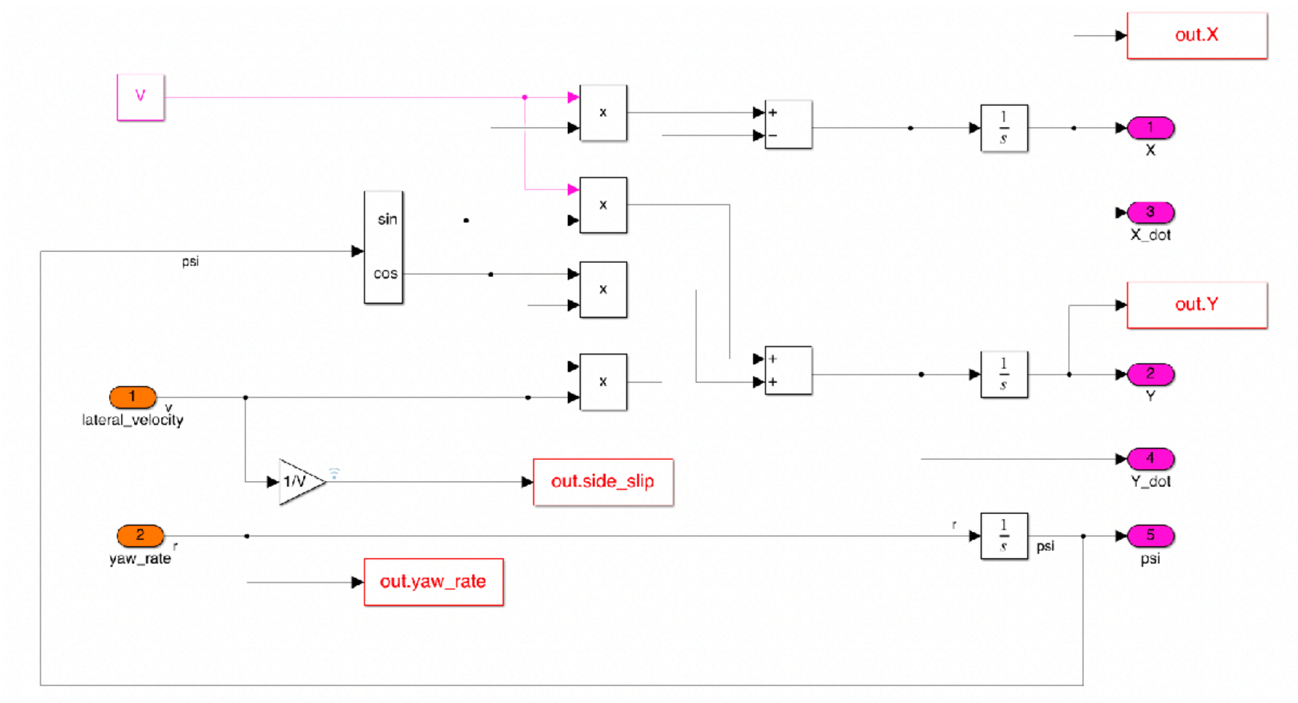



```
R_sim = max(t1.X.Data);

fprintf('R = %.2f [m]', R_sim)
```

```
R = 99.34 [m]
```

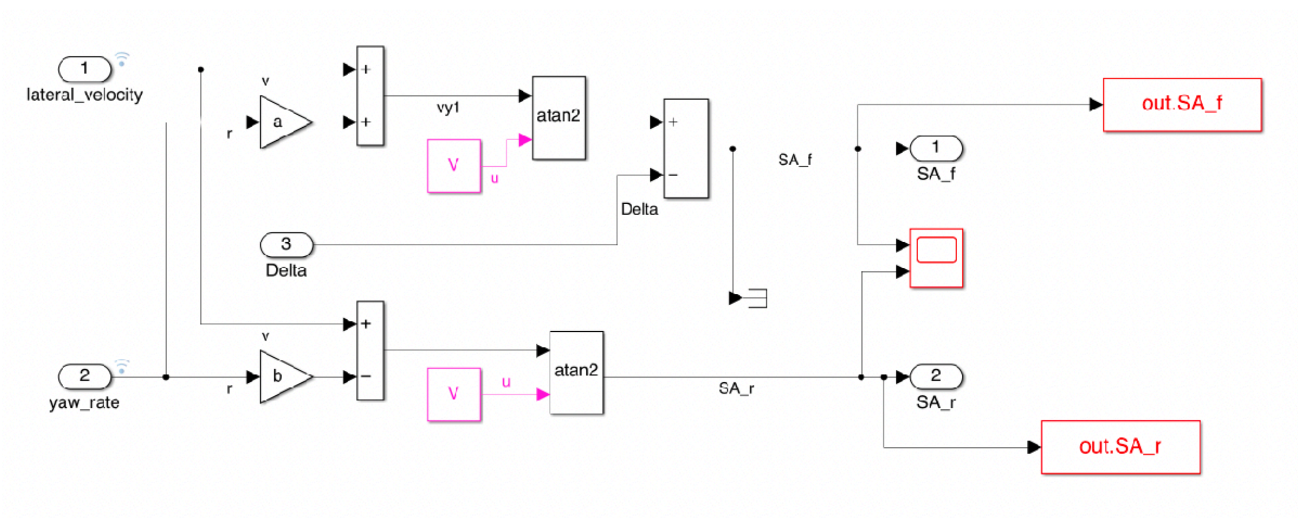
## Task 1 - Simulink

Here, the lateral velocity, yaw rate and yaw is used to calculate the global co-ordinates and x dot for the bicycle model.



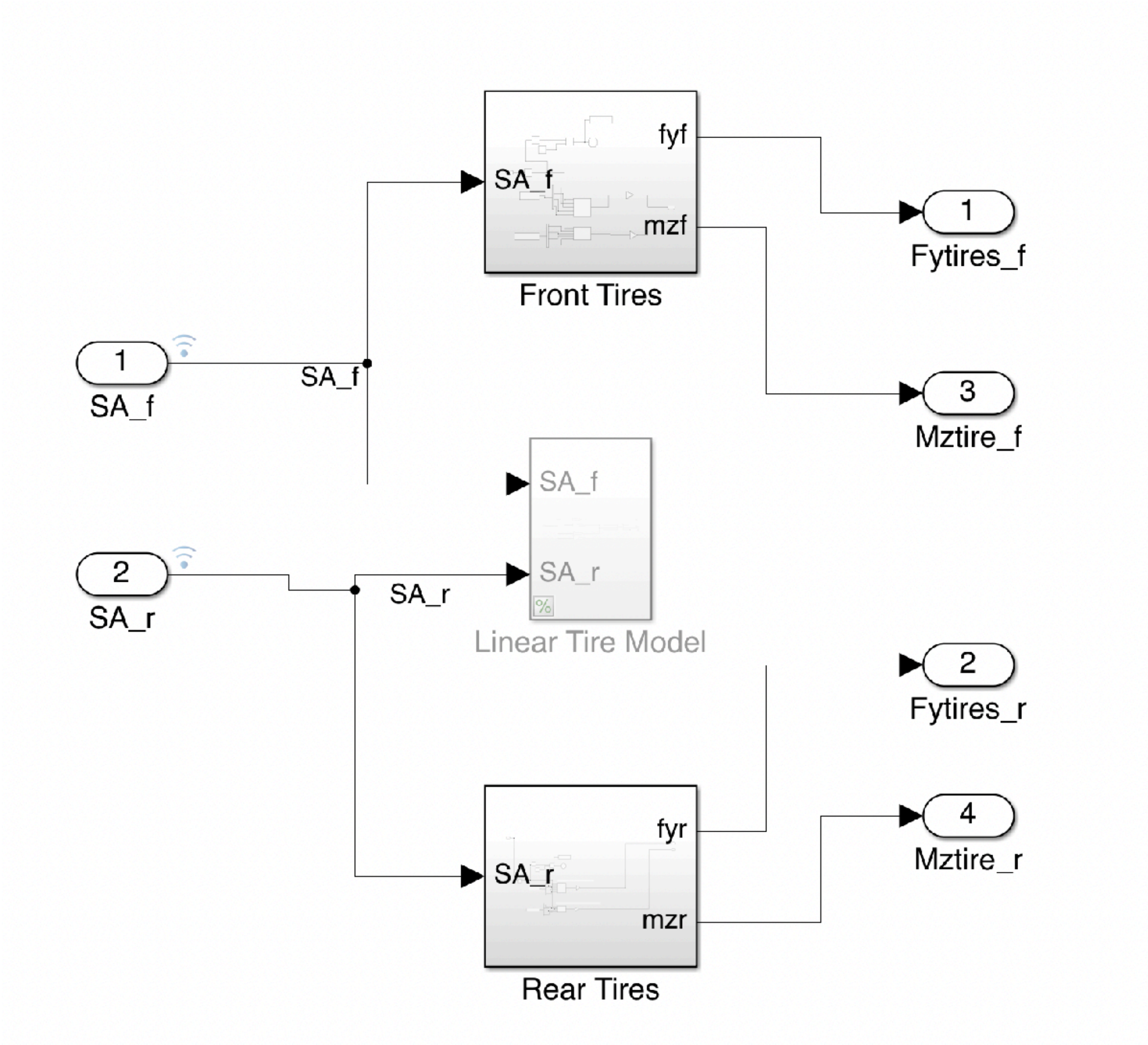

## Slip Angle - Simulink

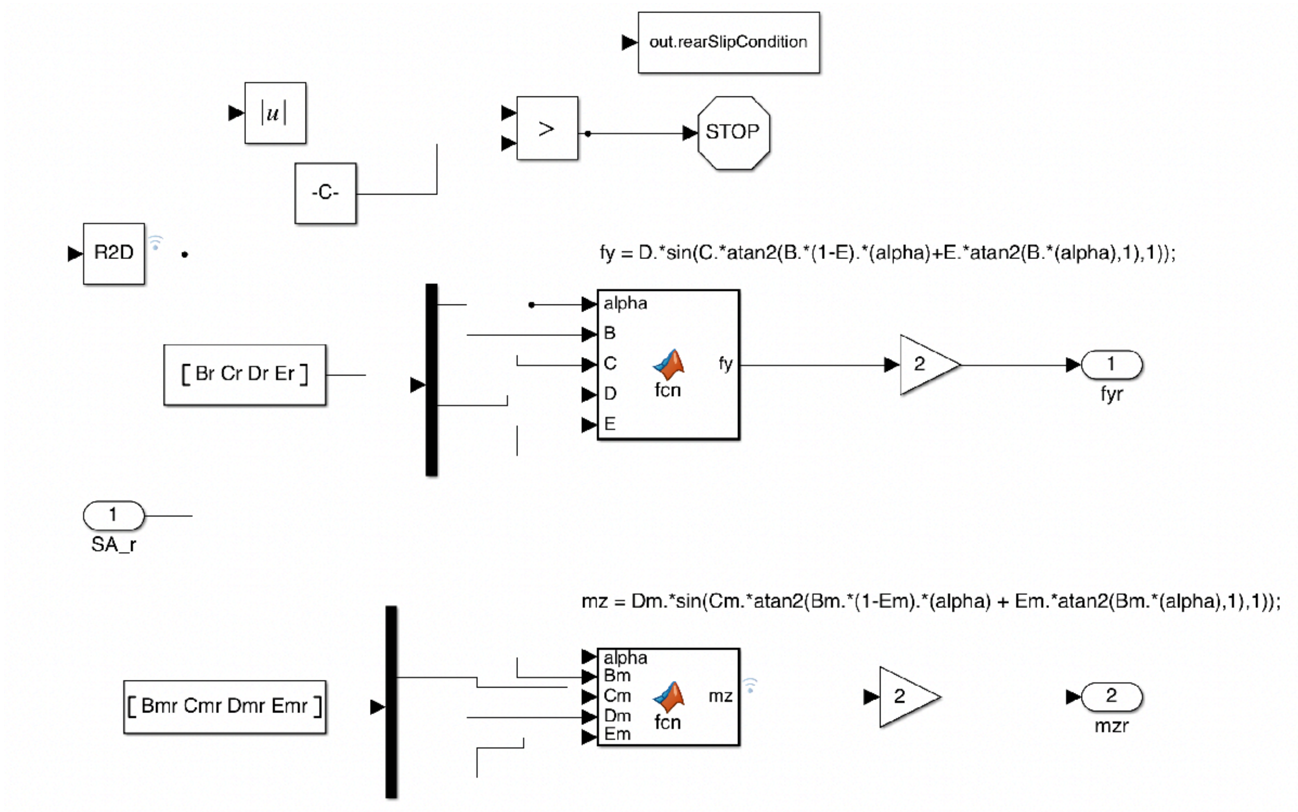
Here, using the lateral velocity and steering angle of the bicycle model, slip angle is calculated for front and rear tire.




## Tire Model - Slides

here, using slip angle for front tire and rear tire lateral forces and moment in Z - direction is calculated for front and rear tire respectively.


fyf
SA_f
mzf
Front Tires
1
Fytires_f
1
SA_f
SA_f
3
Mztire_f
SA_f
2
SA_r
SA_r
SA_r
Linear Tire Model
2
Fytires_r
fyr
SA_r
mzr
Rear Tires
4
Mztire_r



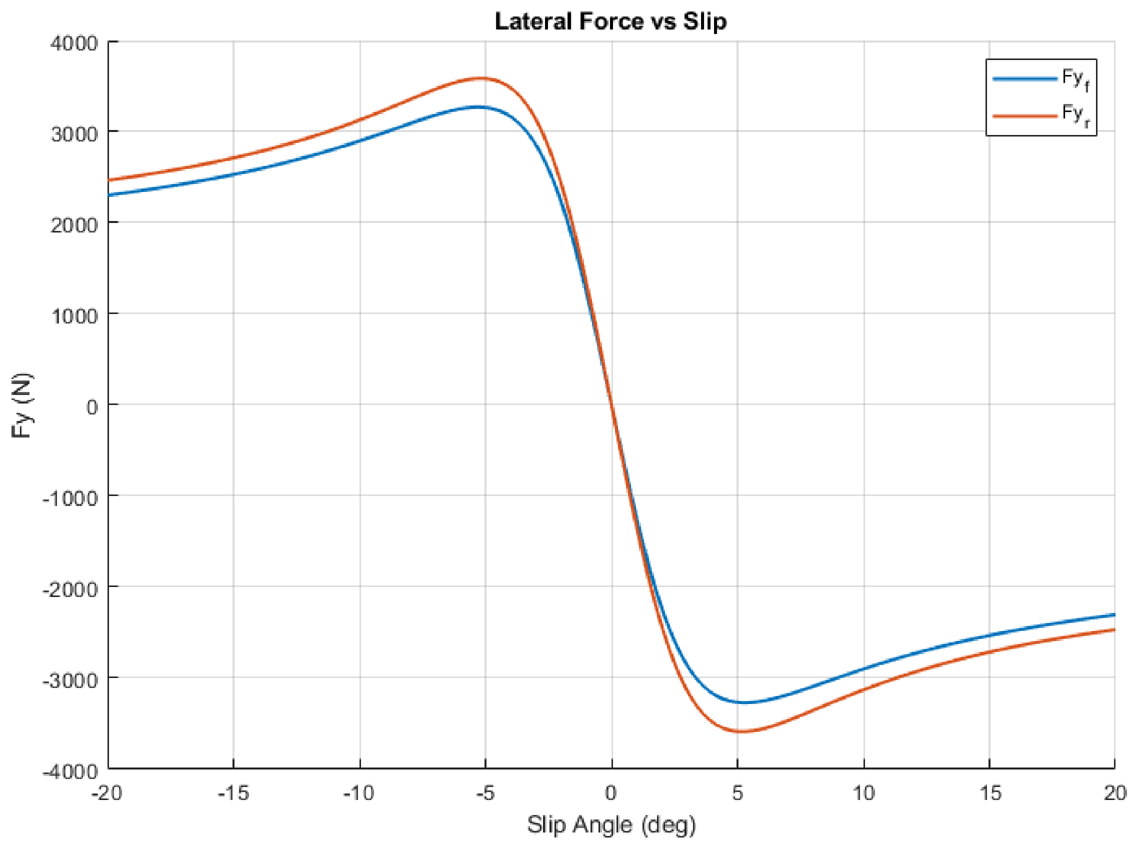

**Where the coefficients $(B, C, D, E)$ where derived from the Pajecka model using the provided matlab function. Fz was input as tire weight in [kN].**

```
% Define constants used in the Simulink model
g = 9.81;              % gravity [m/s^2]


Fzf = WD*mass*g/1000;   % Front axle weight kN
fzf = Fzf/2;            % Front tire weight kN
Fzr =(1-WD)*mass*g/1000; % Rear axle weight kN
fzr = Fzr/2;            % Rear tire weight kN


[Bf,Cf,Df,Ef,Bmf,Cmf,Dmf,Emf] = pacejka_coefficients(fzf);
[Br,Cr,Dr,Er,Bmr,Cmr,Dmr,Emr] = pacejka_coefficients(fzr);


Caf = abs(Bf*Cf*Df)*rad2deg(1);
Car = abs(Br*Cr*Dr)*rad2deg(1);



enablePlotting = false; % turn off the plotting features (makes simulation quicker)
maxYawRate = 3;        % (rad/s) -- stop the simulation if faster
```

```
maxYawRate_deg = rad2deg(maxYawRate); % maximum yaw rate in degrees yaw rate in
[deg/s]
maxSlipAngle = 30;  %max slip in degrees
maxSteeringAngle = atan2(L,6); % max steering angle radians


% Default values of steering scenerio
steeringScenario = 1;         % controls switch in steering signal generator
steerAngle = 1;


% Circle and circumfrence of circle
R=100; % Radius of circle [m]
circumf=2*pi*R; % Circumfrence of the circle [m]


% Default value of velocity and yaw reference to run model
V = velocity;             % velocity [m/s] (40 kmph is 40/3.6 m/s)
time_circle=circumf/V; % The time to complete the circle at the given velocity [s]
yaw_reference=(2*pi)/time_circle; %constant angular velocity for turn and the
reference yaw [rad/s]


%% Plot Tire Behavior
alpha = linspace(-20, 20, 500);


fyf = Df.*sin(Cf.*atan2(Bf.*(1-Ef).*(alpha)+Ef.*atan2(Bf.*(alpha),1),1));


fyr = Dr.*sin(Cr.*atan2(Br.*(1-Er).*(alpha)+Er.*atan2(Br.*(alpha),1),1));


mzf = Dmf.*sin(Cmf.*atan2(Bmf.*(1-Emf).*(alpha) + Emf.*atan2(Bmf.*(alpha),1),1));


mzr = Dmr.*sin(Cmr.*atan2(Bmr.*(1-Emr).*(alpha) + Emr.*atan2(Bmr.*(alpha),1),1));
```

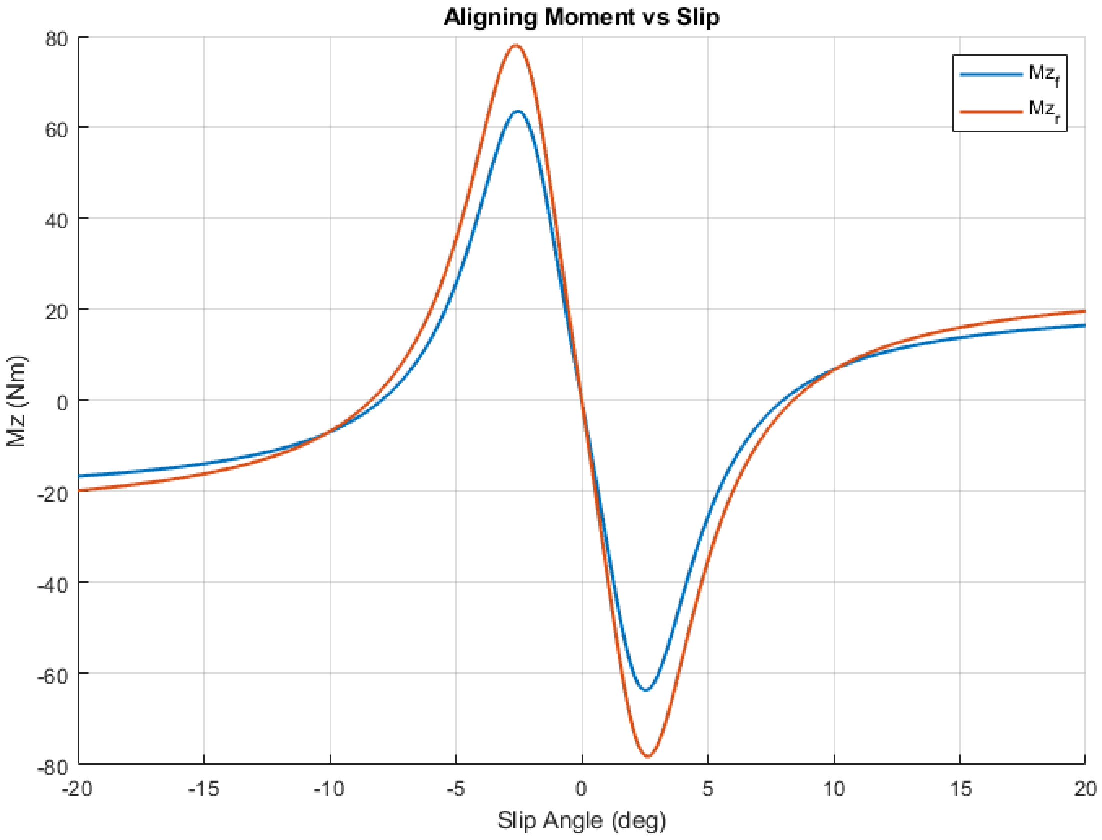
```
% Fy plot
fig = figure('Position', [0, 0, 800, 550]);


hold on
grid on


plot(alpha, fyf, 'linewidth', 1.5)
plot(alpha, fyr, 'linewidth', 1.5)


xlabel('Slip Angle (deg)')
```

```
ylabel('Fy (N)')
title('Lateral Force vs Slip')


legend('Fy_f', 'Fy_r')


saveas(gcf, 'fyvsalpha.png')
```



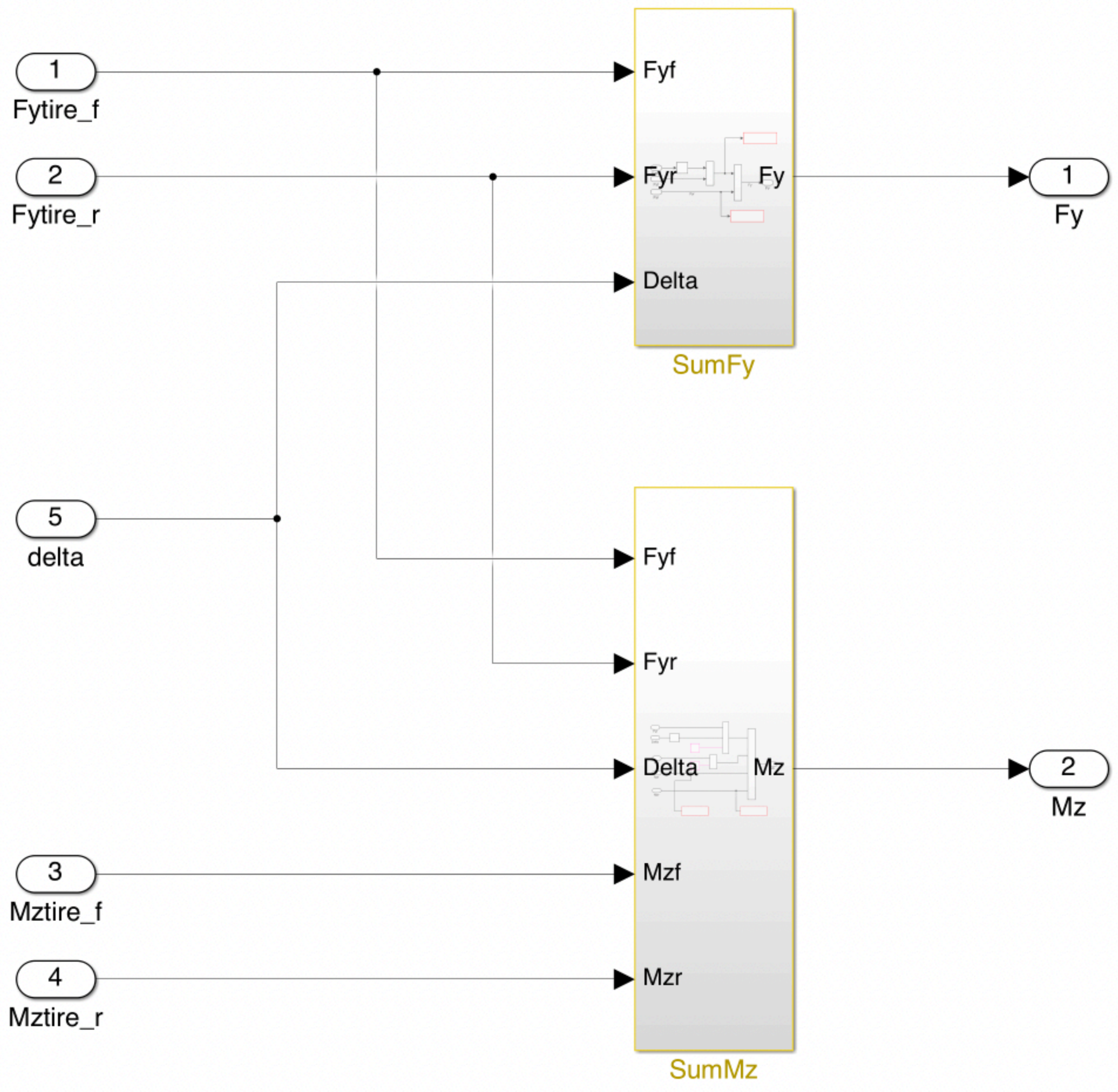

```
% Aligning Moment Plot
fig = figure('Position', [0, 0, 800, 550]);


hold on
grid on


plot(alpha, mzf, 'linewidth', 1.5)
plot(alpha, mzr, 'linewidth', 1.5)


xlabel('Slip Angle (deg)')
ylabel('Mz (Nm)')
title('Aligning Moment vs Slip')
```

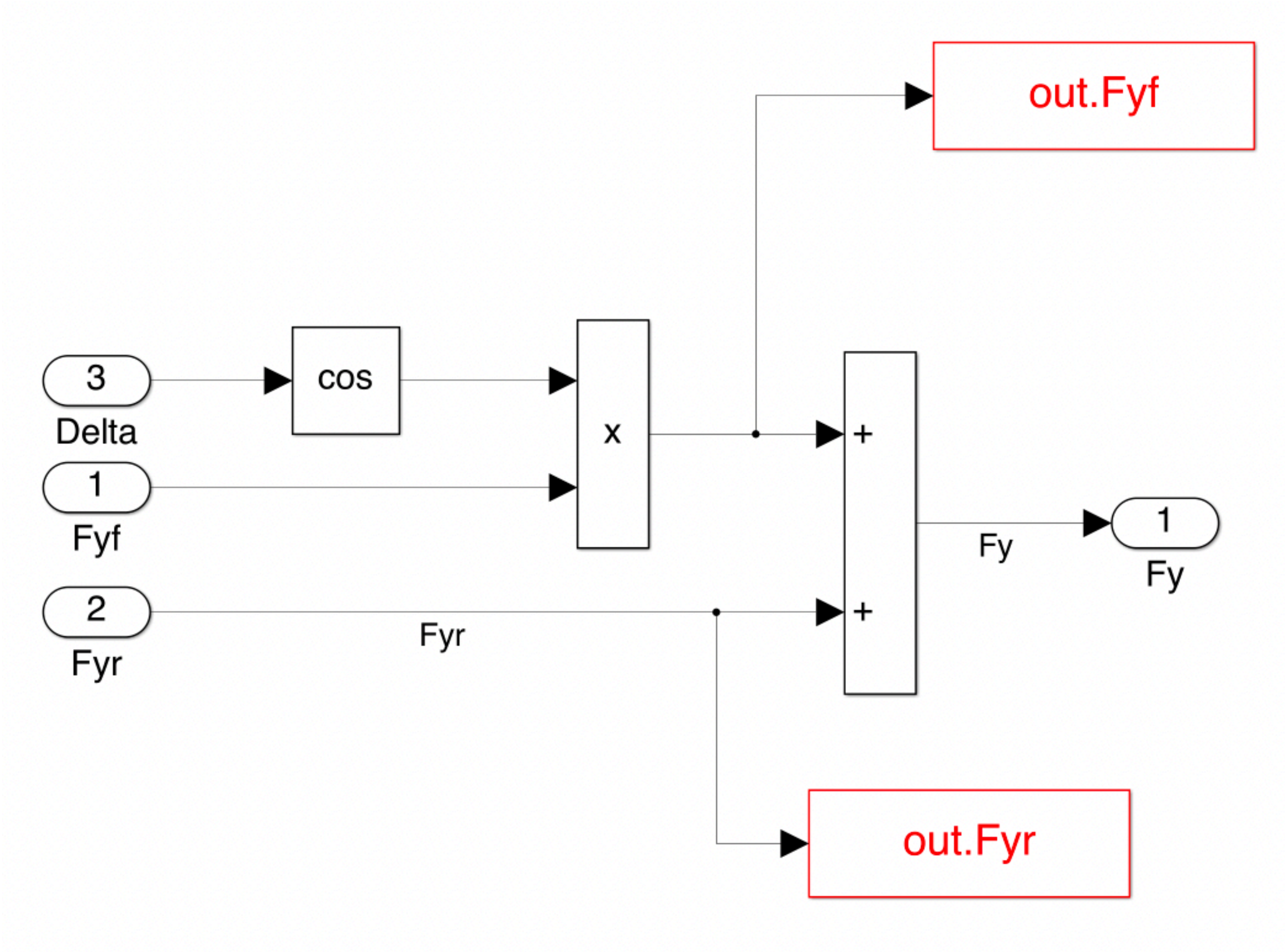
```
legend('Mz_f', 'Mz_r')
```

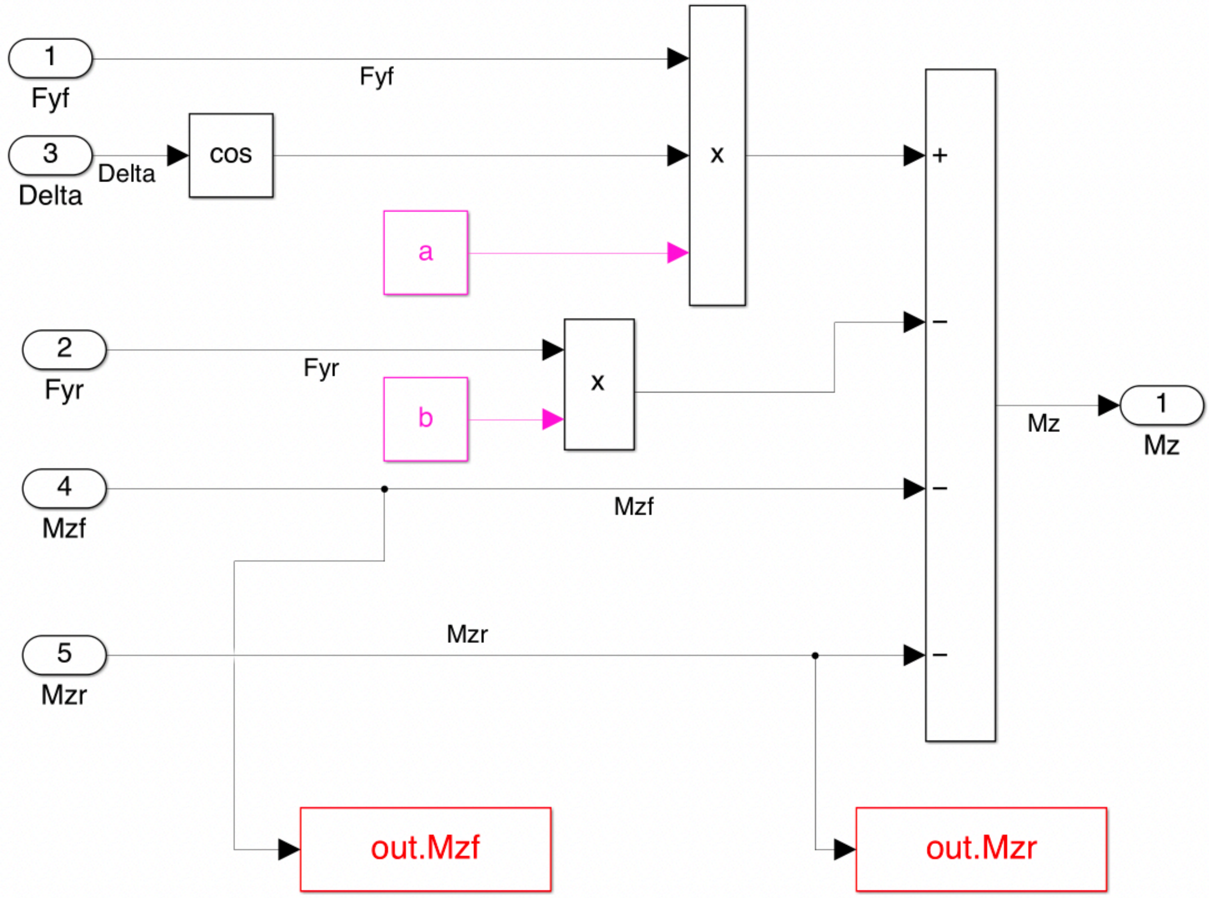



### Local Forces and Moments - Simulink

Here, from the local forces from the front tire and rear tire moment and lateral force on bicycle model is calculated.

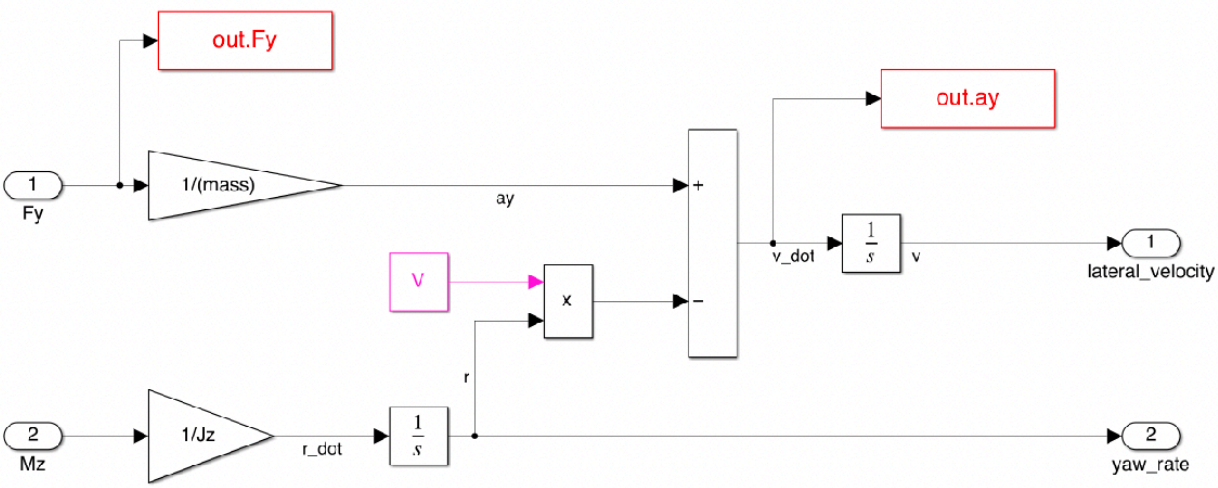

1
Fytire_f
2
Fytire_r
5
delta
3
Mztire_f
4
Mztire_r
Fyf
Fyr
Delta
Fy
SumFy
Fyf
Fyr
Delta
Mz
Mzf
Mzr
SumMz
1
Fy
2
Mz

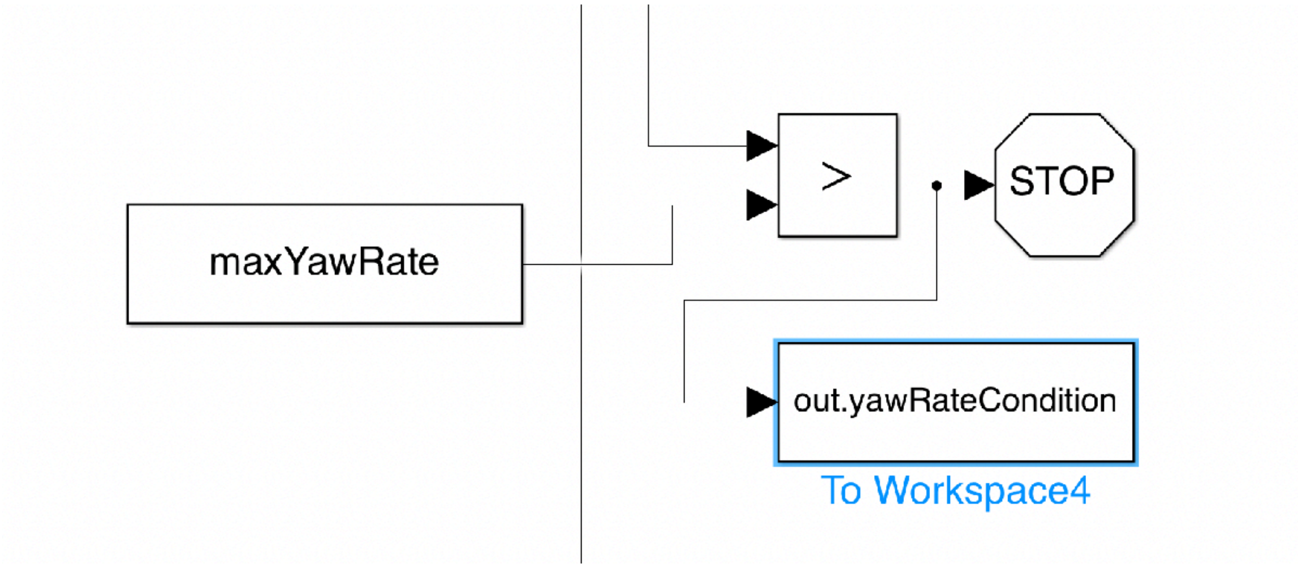

out.Fyf
3
Delta
cos
x
1
Fyf
+
+
Fy
1
Fy
2
Fyr
Fyr
out.Fyr




### EOM - Simulink

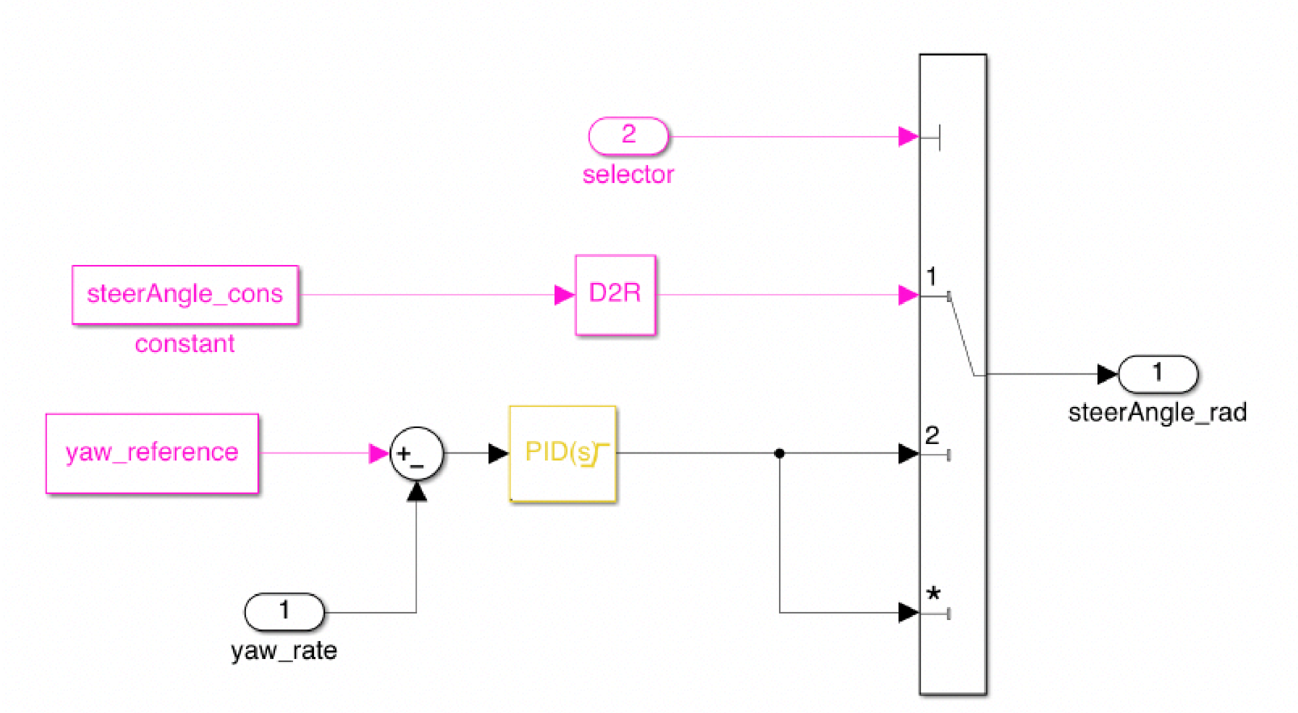
here, the lateral force and moment on the model is used to calculate thelateral velocity and yaw rate






**Steering Model**

for the smaller yaw rate is the input for the PID controller to calculate the adjusted steerign angle




# Sim

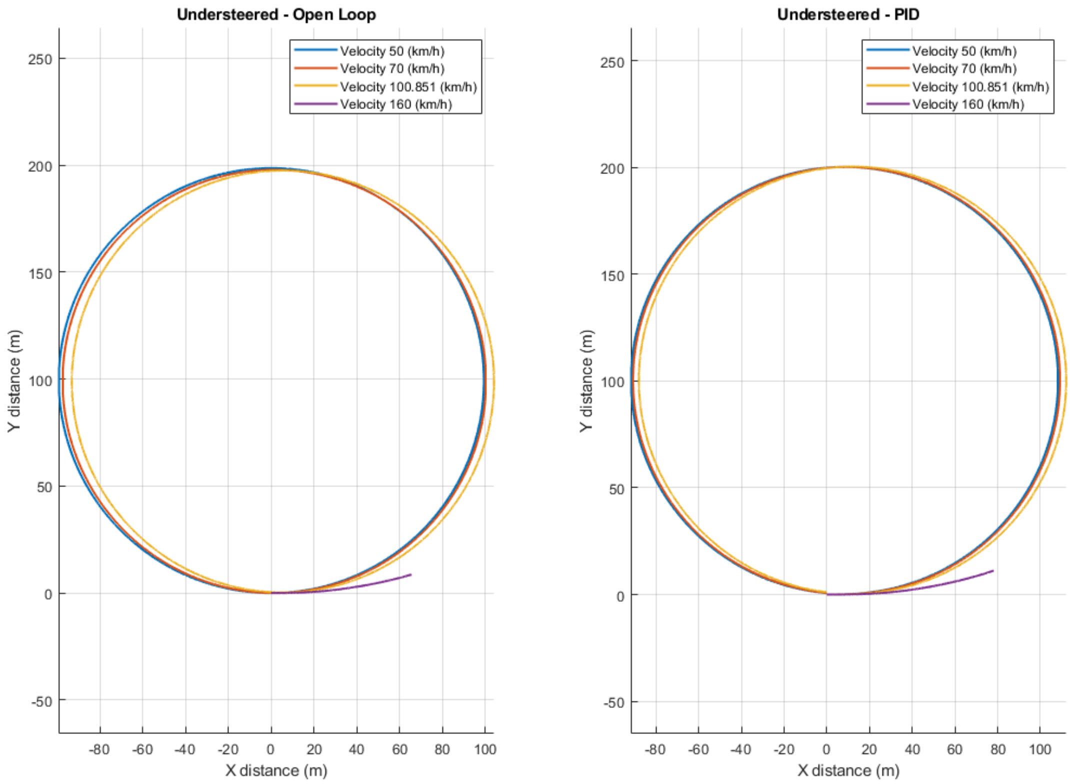
```
%% Test velocities
Vels=[50, 70, velocity*3.6, 160]./3.6; % Vehicle velocities to run the model [km/h]
nv = length(Vels);
```

```
%% PID Controller gains
integral_gain = .3;%85*pi/180;  % Value of integral gain in the PID controller
proportional_gain = 0.005;%45*pi/180; % Value of properotional gain in the PID
controller
derivative_gain = 0.01;%.01*pi/180;


%diff_steering=zeros(1,nv); %Initialalize array with differences of steering angles
[deg]
leg=cell(1,nv); %Initialize the legend cell


steerAngle_cons = atand(L/R); % Steering Angle if constant input [deg]


sims = cell(2, nv, 2);
for k=1:2
    u = 1;
    steeringScenario = k; % Set the steering sceneio to k,
    % steeringScenario 1 is constant steering while scenario 2 is PID
    for i=1:nv
        V=Vels(i);
        time_circle= min(300,circumf/V); % The time to complete the circle at the
given velocity [s]
        yaw_reference=(2*pi)/time_circle; % constant angular velocity for turn and
the reference yaw [rad/s]
        sims{k, i, u} =
sim('bicycle_2DOF_PID_controller.slx','StopTime',num2str(time_circle)); % Increase
the simulation time for 1km/h

        if any(nnz(sims{k,i,u}.yawRateCondition.Data))
            warning('Simulation terminated: Maxmimum Yaw rate %g (rad/s) exceeded at
%g (km/h) in steering scenario %g at simulation time %g (s)',maxYawRate, V*3.6,
steeringScenario, task.yaw_rate.Time(end))
        end

        if steeringScenario == 1
            leg{i}=sprintf('Velocity %g (km/h)', Vels(i)*3.6); %Cell with legend
names of velocities
        end
    end
end
```

## Plots

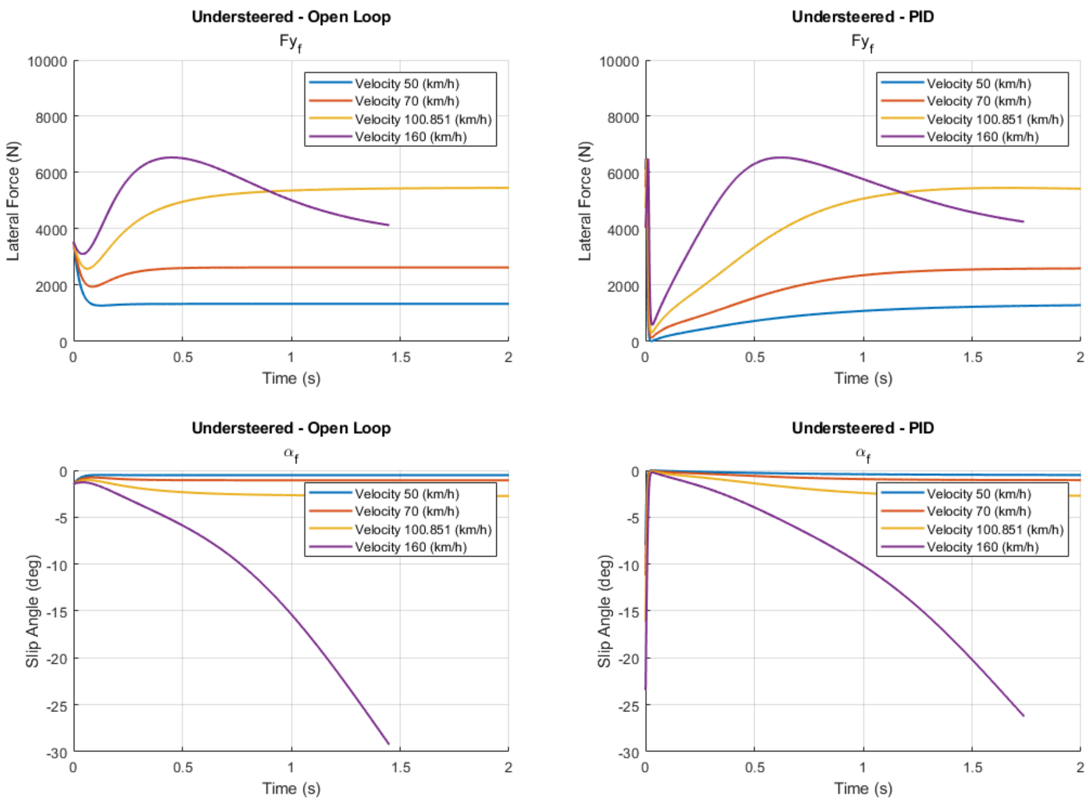
```
%% Generate Plots
% different simulatin cases
cases = cell(1,4);
```

```
cases{1} = 'Understeered - Open Loop';
cases{2} = 'Understeered - PID';
cases{3} = 'Oversteered - Open Loop';
cases{4} = 'Oversteered - PID';


% Plot understeer trajectories
fig = figure('Position', [0, 0, 1200, 800]);


p = 1;
u = 1;
for k = 1:2
    subplot(1,2,p)
    axis equal
    grid on
    hold on
    for i = 1:nv
        plot(sims{k,i,u}.X.data, sims{k,i,u}.Y.data, 'linewidth', 1.5)
    end
    title(cases{p})
    xlabel('X distance (m)')
    ylabel('Y distance (m)')
    legend(leg{:})
    p = p+1;
end
```




```
% plot understeer Fy and alpha
fig = figure('Position', [0, 0, 1200, 800]);


p = 1;
u = 1;
for k = 1:2
    subplot(2,2,p)
    grid on
    hold on
    for i = 1:nv
        plot(sims{k,i,u}.Fyf.Time, sims{k,i,u}.Fyf.Data, 'linewidth', 1.5)
    end
    xlim([0,2])
    ylim([0 10^4])
    title(cases{p}, 'Fy_f')
    xlabel('Time (s)')
    ylabel('Lateral Force (N)')
    legend(leg{:})

    subplot(2,2,p+2)
    grid on
    hold on
    for i = 1:nv
```

```
        plot(sims{k,i,u}.SA_f.Time, sims{k,i,u}.SA_f.data*rad2deg(1), 'linewidth',
1.5)
    end
    xlim([0,2])
    title(cases{p}, '\alpha_f')
    xlabel('Time (s)')
    ylabel('Slip Angle (deg)')
    legend(leg{:})
    p = p+1;
end
```




# Simulation

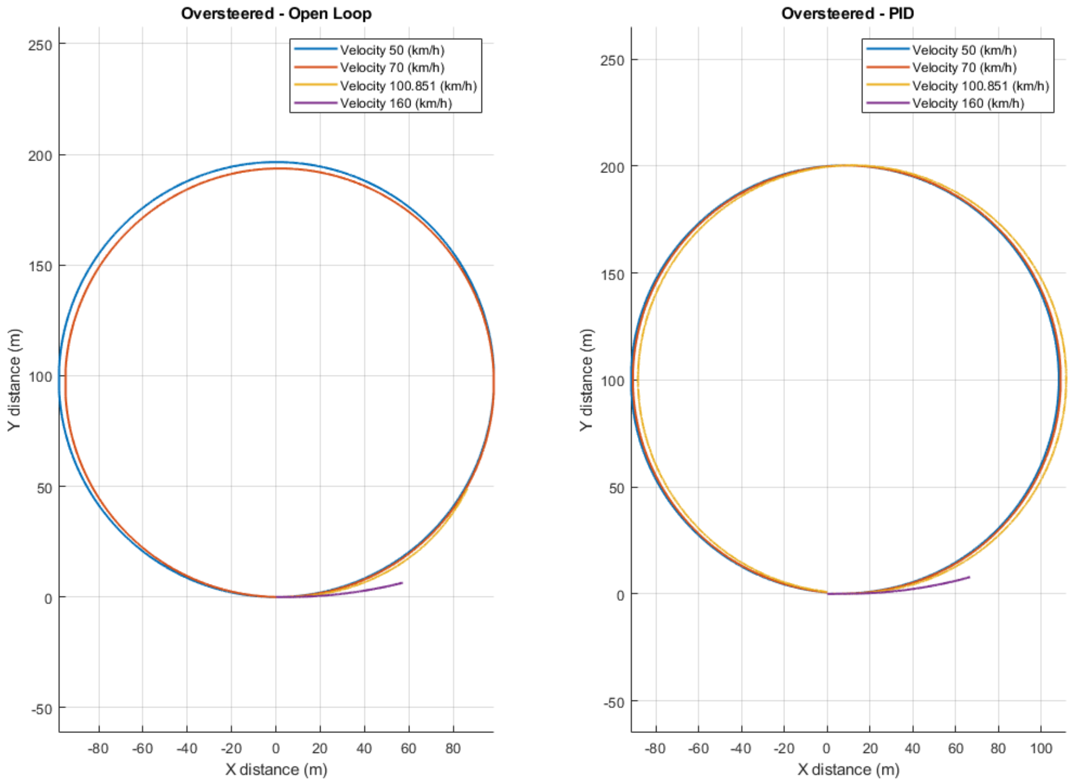
```
%% PID Controller - oversteered vehicle at different speeds
%Caf = 5000*rad2deg(1); % Increase tire cornering stiffness front to make vehicle
oversteered [N/rad]
WD = 0.45;


% Derived quantities using the weight distribution of the vehicle
a = (1-WD)*L;           % front axle distance from CG [m]
b = WD*L ;              % rear axle distance from CG [m]
```

```
Fzf = WD*mass*g/1000;   % Front axle weight kN
fzf = Fzf/2;
Fzr =(1-WD)*mass*g/1000; % Rear axle weight kN
fzr = Fzr/2;


% redefine tire coefficients for new weight distribution
[Bf,Cf,Df,Ef,Bmf,Cmf,Dmf,Emf] = pacejka_coefficients(fzf);
[Br,Cr,Dr,Er,Bmr,Cmr,Dmr,Emr] = pacejka_coefficients(fzr);


Caf = abs(Bf*Cf*Df)*rad2deg(1);
Car = abs(Br*Cr*Dr)*rad2deg(1);


steerAngle_cons=atand(L/R); % Steering Angle if constant input [deg]
for k=1:2
    u = 2;
    steeringScenario = k; % Set the steering sceneio to k,
    % steeringScenario 1 is consant steering while scenario 2 is PID
    for i=1:nv
        V=Vels(i);
        time_circle= min(300, circumf/V); % The time to complete the circle at the
given velocity [s]
        yaw_reference=(2*pi)/time_circle; % constant angular velocity for turn and
the reference yaw [rad/s]

        sims{k,i,u} =
sim('bicycle_2DOF_PID_controller.slx','StopTime',num2str(time_circle)); % Increase
the simulation time for 1km/h

        if any(nnz(sims{k,i,u}.yawRateCondition.Data))
            warning('Simulation terminated: Maxmimum Yaw rate %g (rad/s) exceeded at
%g (km/h) in steering scenario %g at simulation time %g (s)',maxYawRate, V*3.6,
steeringScenario, task.yaw_rate.Time(end))
        end
    end
end
```

# Plots

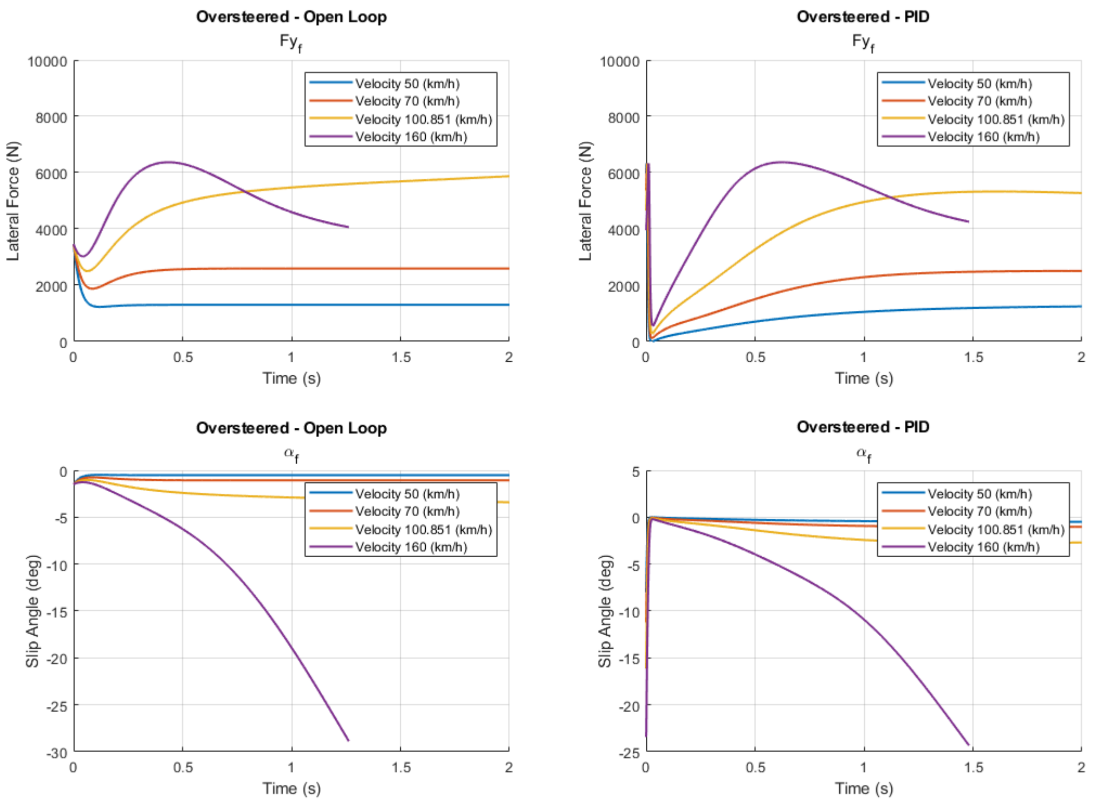
```
% plot oversteer trajectories
fig = figure('Position', [0, 0, 1200, 800]);


p = 1;
u = 2;
for k = 1:2
    subplot(1,2,p)
    axis equal
    grid on
    hold on
```

```
    for i = 1:nv
        plot(sims{k,i,u}.X.data, sims{k,i,u}.Y.data, 'linewidth', 1.5)
    end
    title(cases{p+2})
    xlabel('X distance (m)')
    ylabel('Y distance (m)')
    legend(leg{:})
    p = p+1;
end
```




```
% plot oversteer Fy and alpha
fig = figure('Position', [0, 0, 1200, 800]);


p = 1;
u = 2;
for k = 1:2
    subplot(2,2,p)
    grid on
    hold on
    for i = 1:nv
        plot(sims{k,i,u}.Fyf.Time, sims{k,i,u}.Fyf.Data, 'linewidth', 1.5)
    end
    xlim([0,2])
    ylim([0 10^4])
    title(cases{p+2}, 'Fy_f')
```

```
    xlabel('Time (s)')
    ylabel('Lateral Force (N)')
    legend(leg{:})

    subplot(2,2,p+2)
    grid on
    hold on
    for i = 1:nv
        plot(sims{k,i,u}.SA_f.Time, sims{k,i,u}.SA_f.data*rad2deg(1), 'linewidth',
1.5)
    end
    xlim([0,2])
    title(cases{p+2}, '\alpha_f')
    xlabel('Time (s)')
    ylabel('Slip Angle (deg)')
    legend(leg{:})
    p = p+1;
end
```




## Profit Model

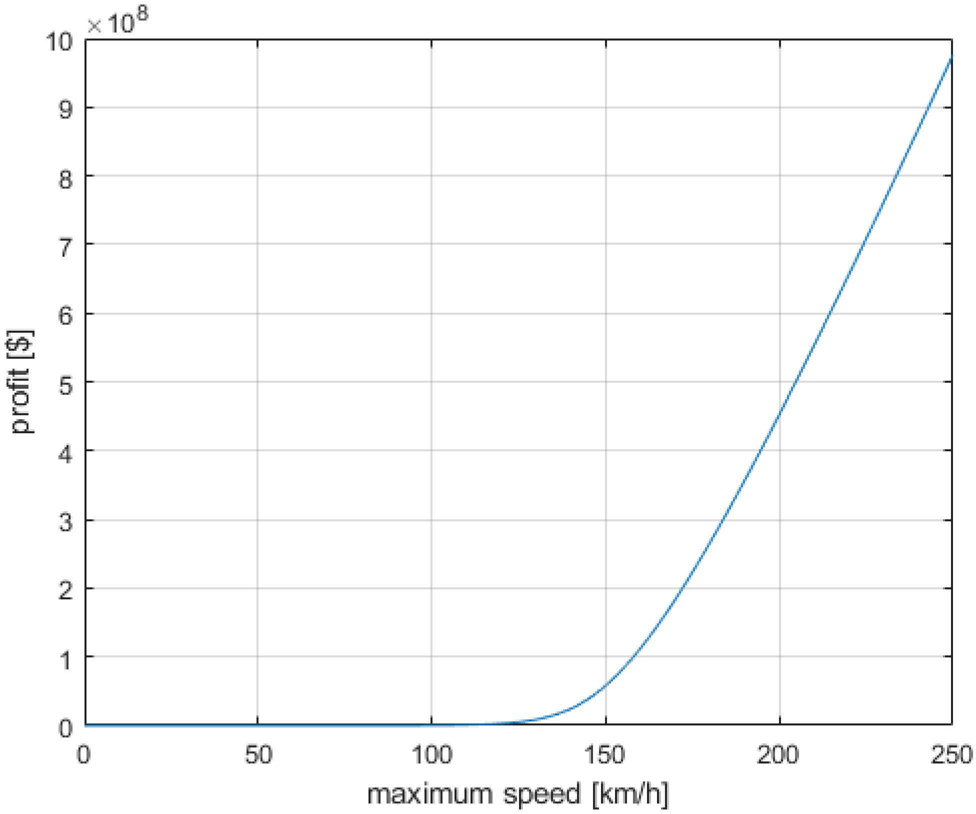
**Profit Model:**We used the profit model given to us which has the data of competitor's car and upper limits of the profit function which we should not exceed.

We successfully designed our vehicle in such a way that it would give us the maximum performance for that setup while keeping the cost at minimum,

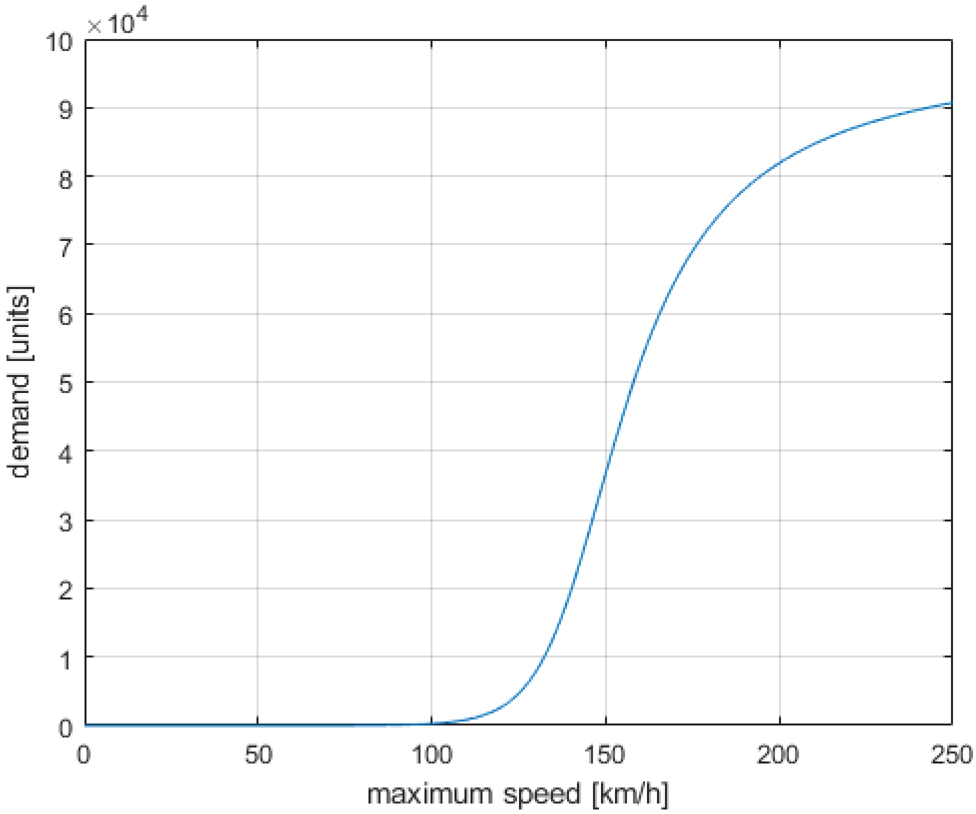
this helped us to increase profit.

```
%%
% Your competition is selling a BEV with the following attributes:
%   price = $40,000
%   range = 200 km
%   0-100km/h = 6 seconds
%   maximum speed = 220 km/h
competitionAttributeValues = [40000;200;6;220];
myAttributeValues = [37598.3536;312;5.2;237.9819];


% We use the default model for the BEV
%    marketSize = 1 million
%    betaCoeff = [-1; 35; -2800; 150]/1000;
% The meaning of the coefficients is that the customers are willing to pay
%    $1 to get a $1 reduction in price (pretty obvious...)
%    $40 to get an increase in range of 1 mile
%    $3,200 to get a reduction of 1 second in the 0-100km/h-time
%    $120 to get an increase of 1km/h in maximum speed
% (note the division by 1000 is a scaling factor that determines how
% quickly the demand changes when you do better/worse than the competition)


% We can now explore how the profit, demand and optimal price change as a
% function of each of the attributes.  For instance, start with cost:


%%%%%%
attNum = 4; % change this to  2, 3, or 4 to sweep the other attributes
%%%%%%
attName = {'cost [$]',
           'range [km]',
           '0-100km/h acceleration time [s]',
           'maximum speed [km/h]'};


upperBound = [90000; 300; 15; 250];
sweep = linspace(0,upperBound(attNum),1000);
f = zeros(size(sweep));
d = zeros(size(sweep));
p = zeros(size(sweep));
for i=1:length(sweep)
    att = myAttributeValues;
    att(attNum) = sweep(i);
    [profit,demand,price] = profitPredict(att, competitionAttributeValues);
    f(i) = profit;
    d(i) = demand;
    p(i) = price;
end
```

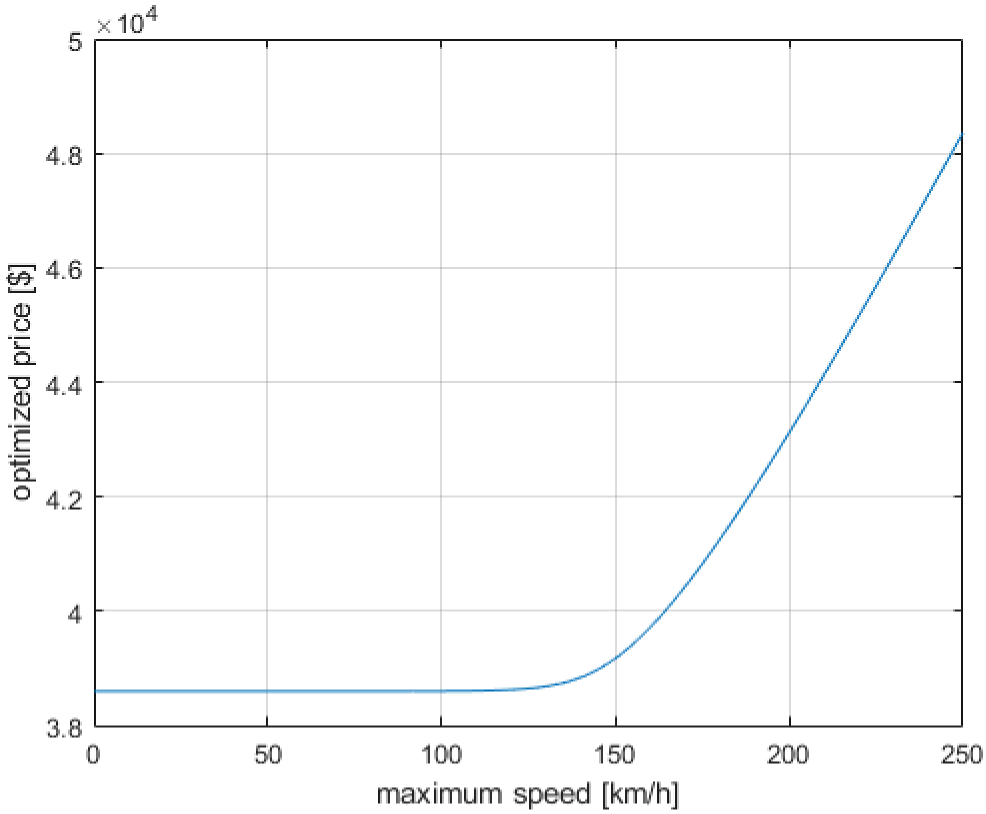
```
figure(10)
plot(sweep,f)
grid on
hold on
xlabel(attName{attNum});
ylabel('profit [$]')
```




```
figure(20)
plot(sweep,d)
grid on
hold on
xlabel(attName{attNum});
ylabel('demand [units]')
```



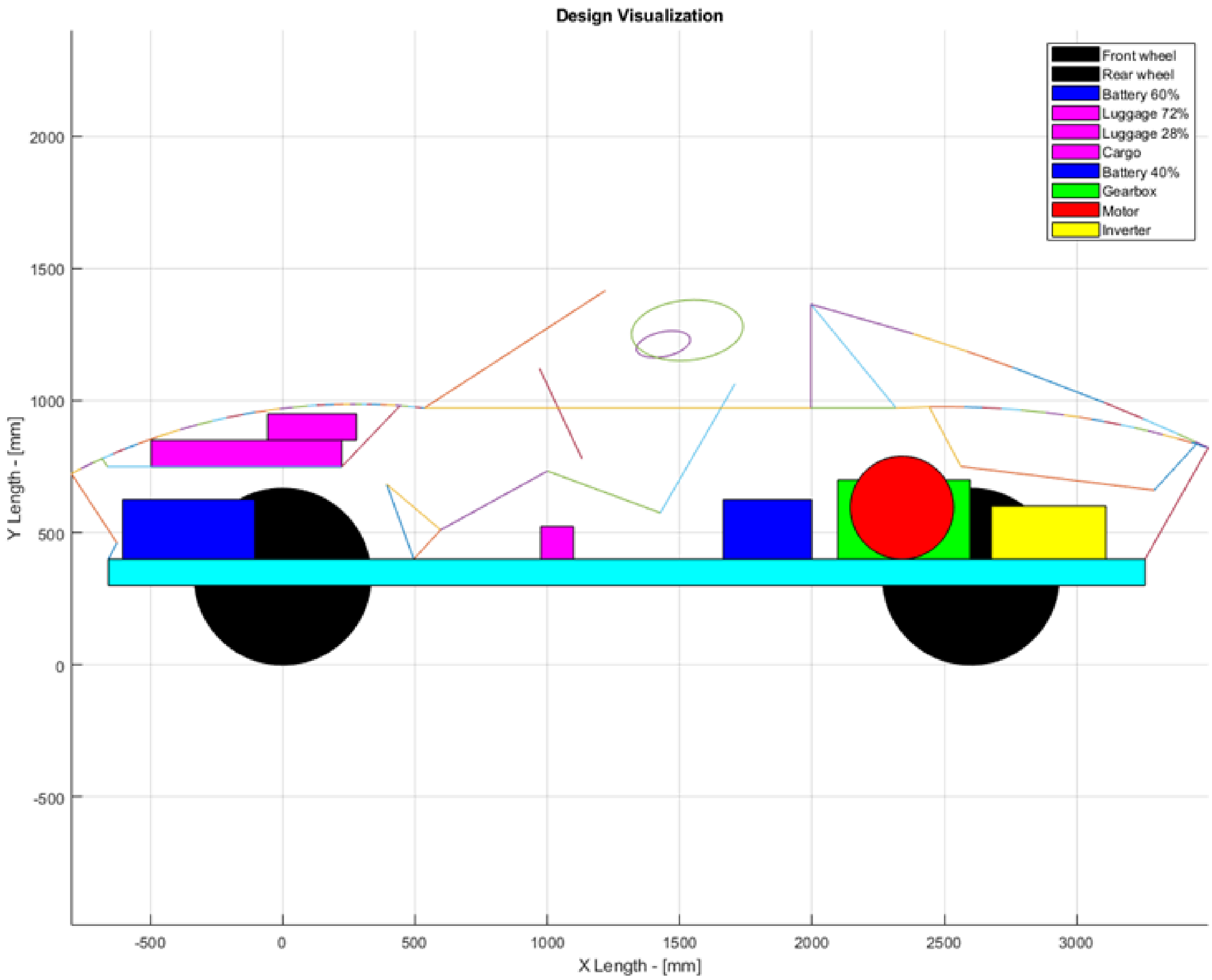

```
figure(30)
plot(sweep,p)
grid on
hold on
xlabel(attName{attNum});
ylabel('optimized price [$]')
```



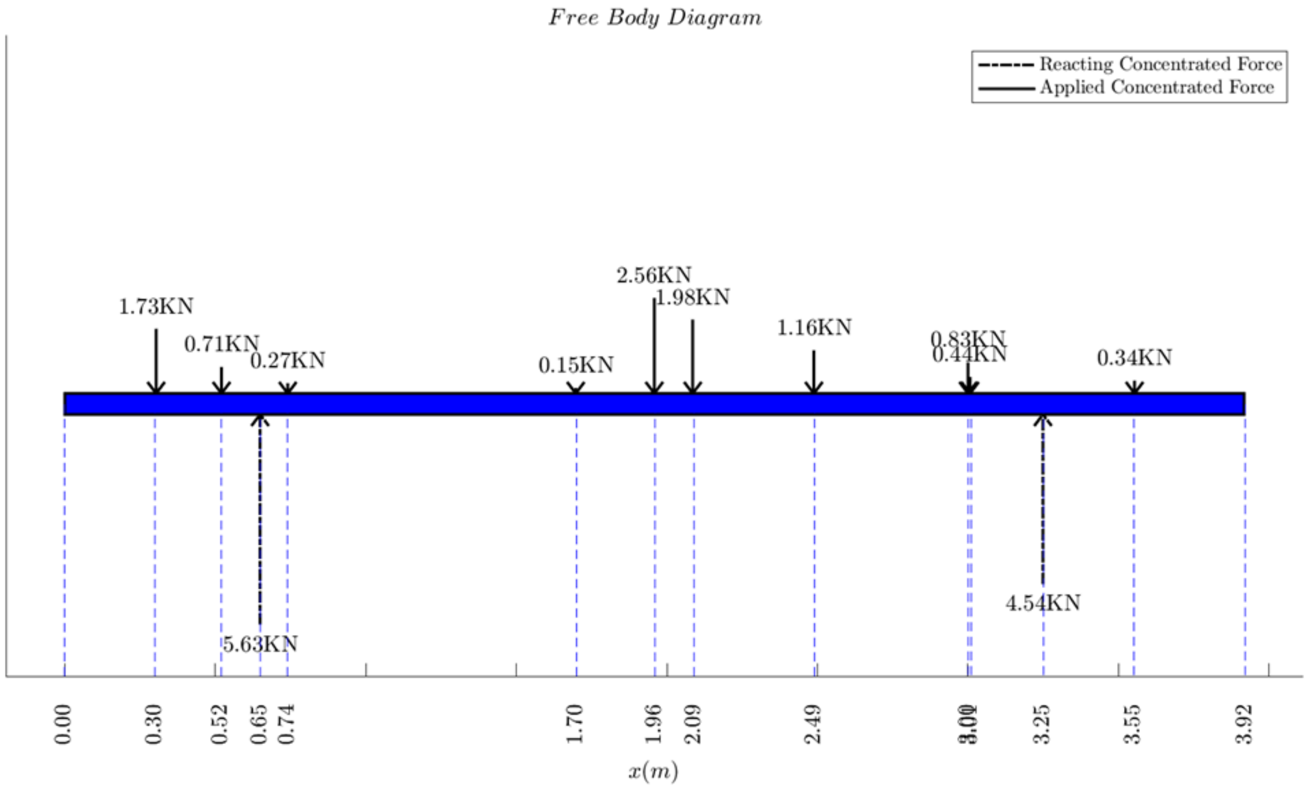

```
disp([num2str(profit/10^6), ' Million $']);
976.3042 Million $
```

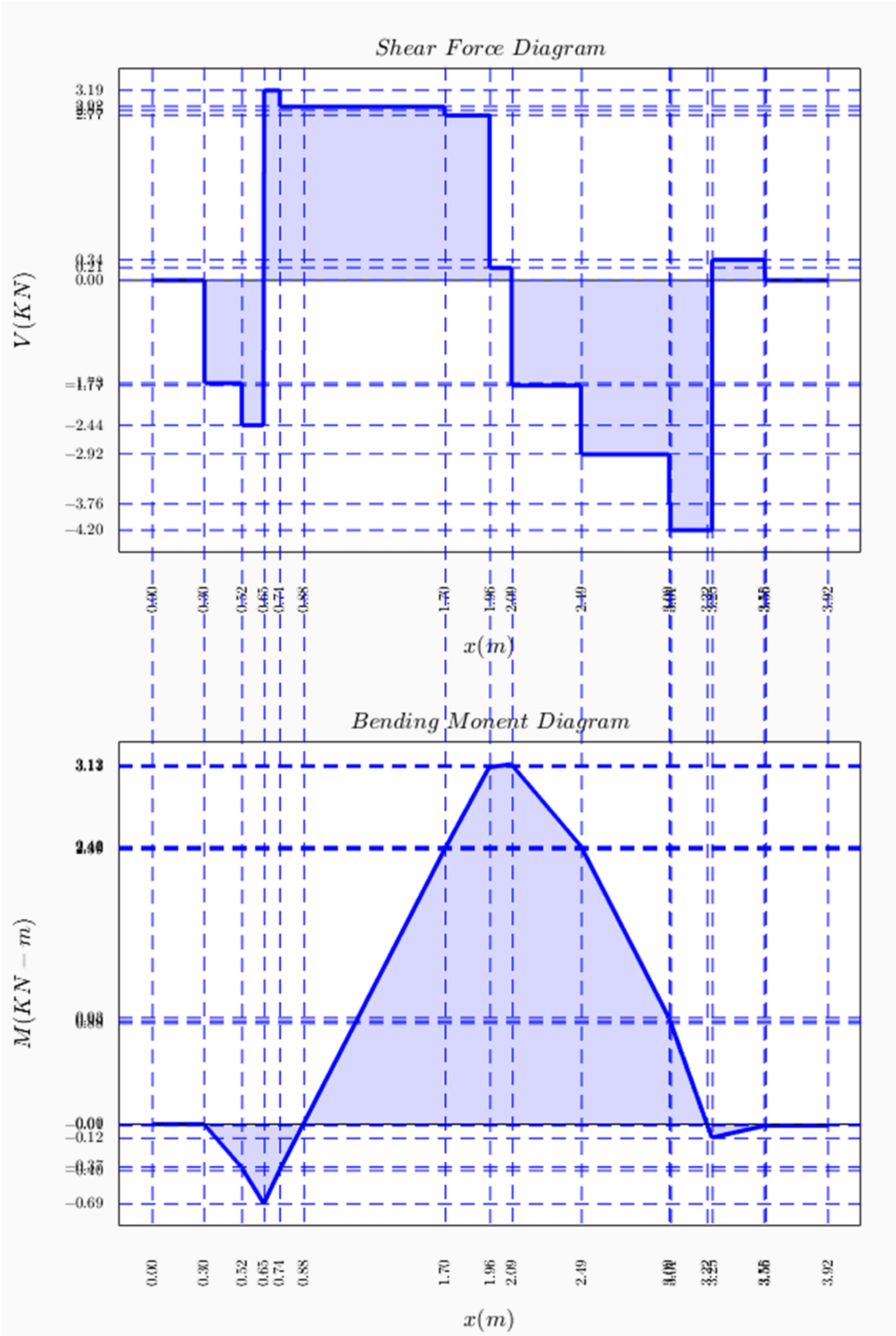
**3) Describe each test-case and include the results (if possible/appropriate in a graph). Describe how the results show your models are functioning as expected.**

**Design Visualization Result:**




**SFBM Calculations Results:**


Free Body Diagram
Reacting Concentrated Force
Applied Concentrated Force
1.73KN
0.71KN
0.27KN
0.15KN
2.56KN
1.98KN
1.16KN
0.83KN
0.44KN
0.34KN
5.63KN
4.54KN
0.00
0.30
0.52
0.65
0.74
1.70
1.96
2.09
2.49
3.25
3.55
3.92
x(m)

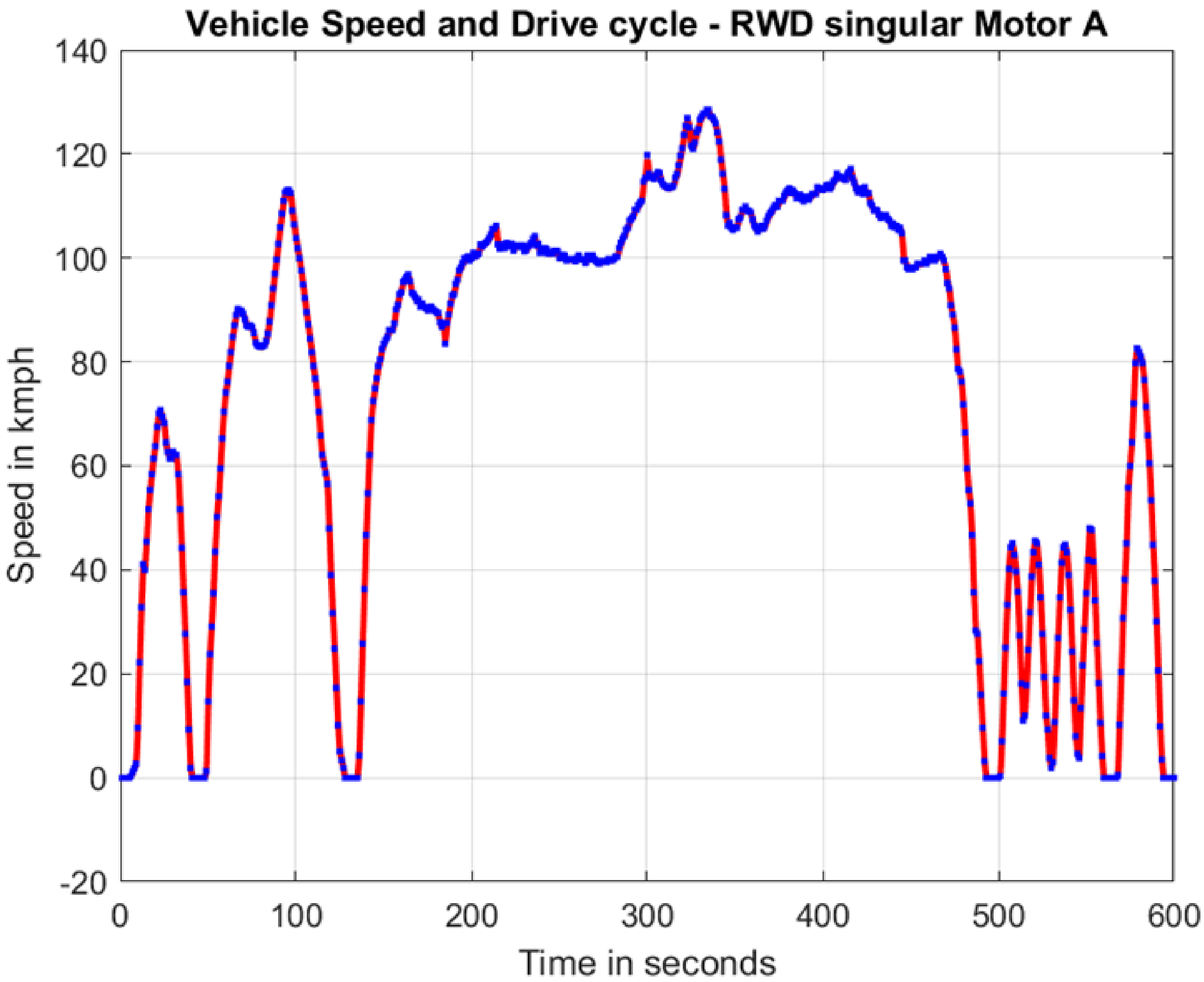

Shear Force Diagram
V(KN)
3.19
0.00
−2.44
−2.92
−3.76
−4.20
x(m)
Bending Moment Diagram
M(KN − m)
3.13
2.40
−0.00
−0.12
−0.69
0.00
0.30
0.52
0.65
0.74
0.88
1.70
1.96
2.09
2.49
3.92
x(m)

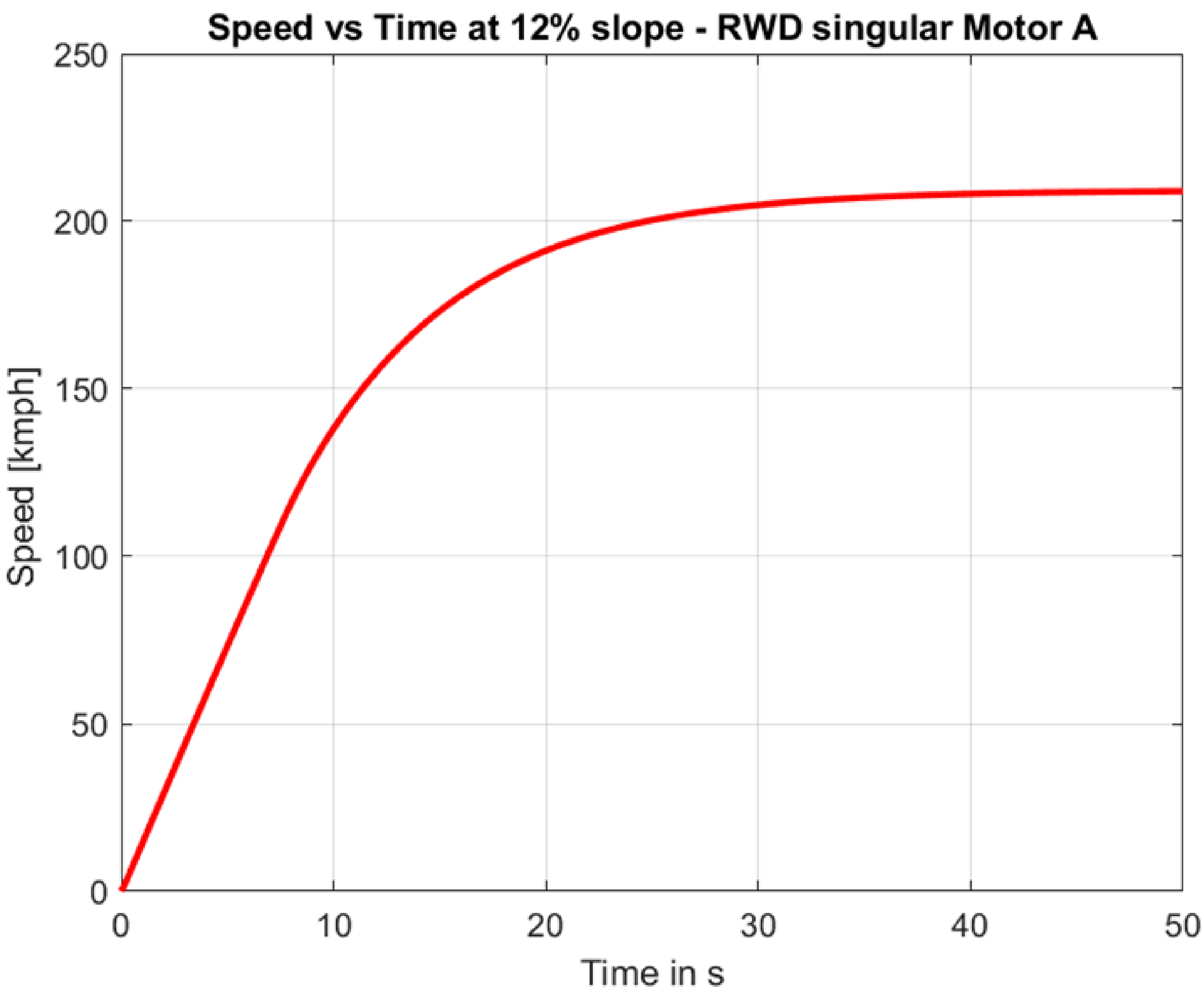
`The Maximum Bending Stress is 111.952310 MPa.`

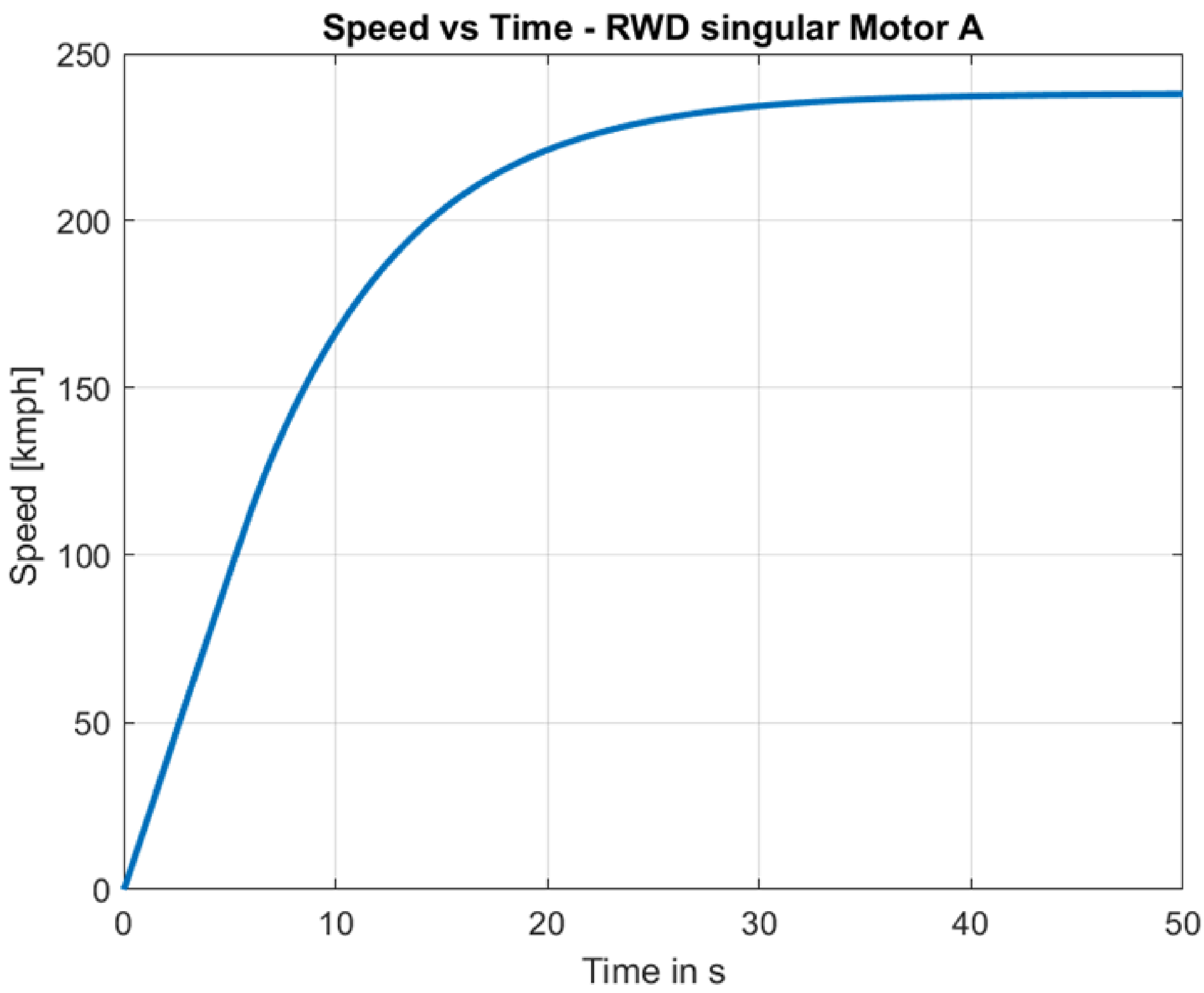
`Ø The Deflection of the Beam is -4.911665 mm.`

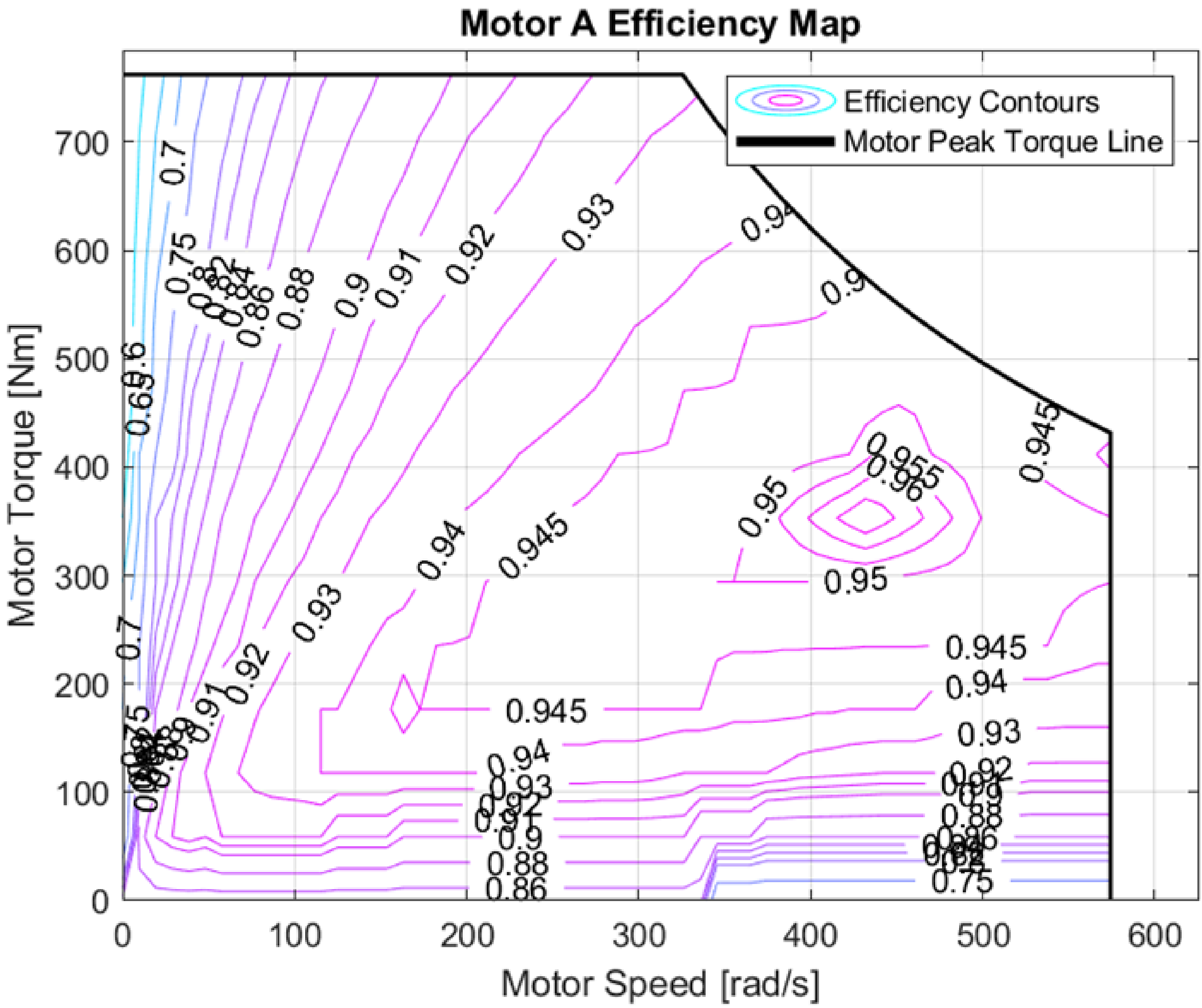
`Ø The Mass of the Frame Rail is 38.191725 kg.`

**Powertrain Selection Results:**




Speed vs Time at 12% slope - RWD singular Motor A
Speed [kmph]
250
200
150
100
50
0
0
10
20
30
40
50
Time in s


Speed vs Time - RWD singular Motor A
Speed [kmph]
250
200
150
100
50
0
0
10
20
30
40
50
Time in s

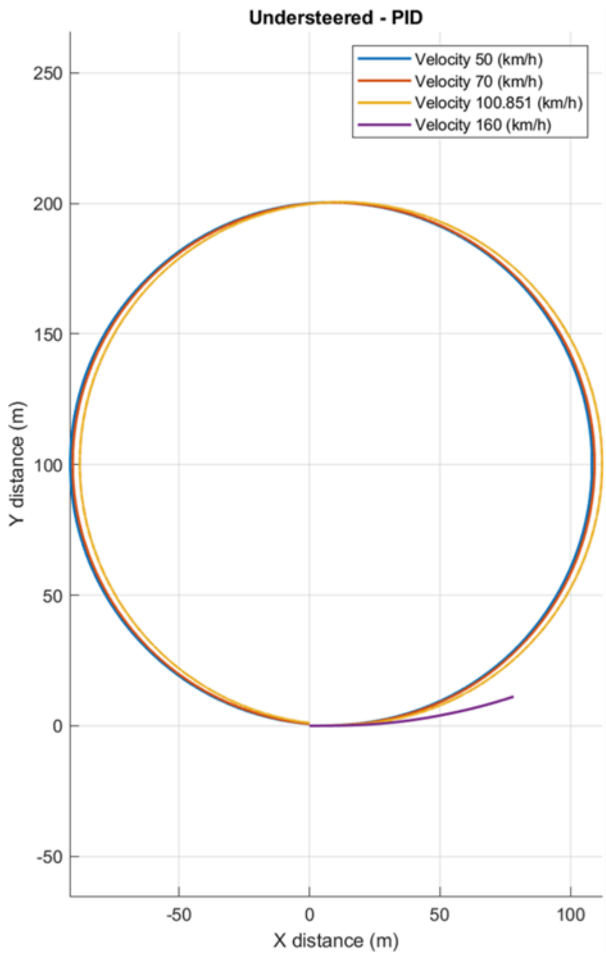


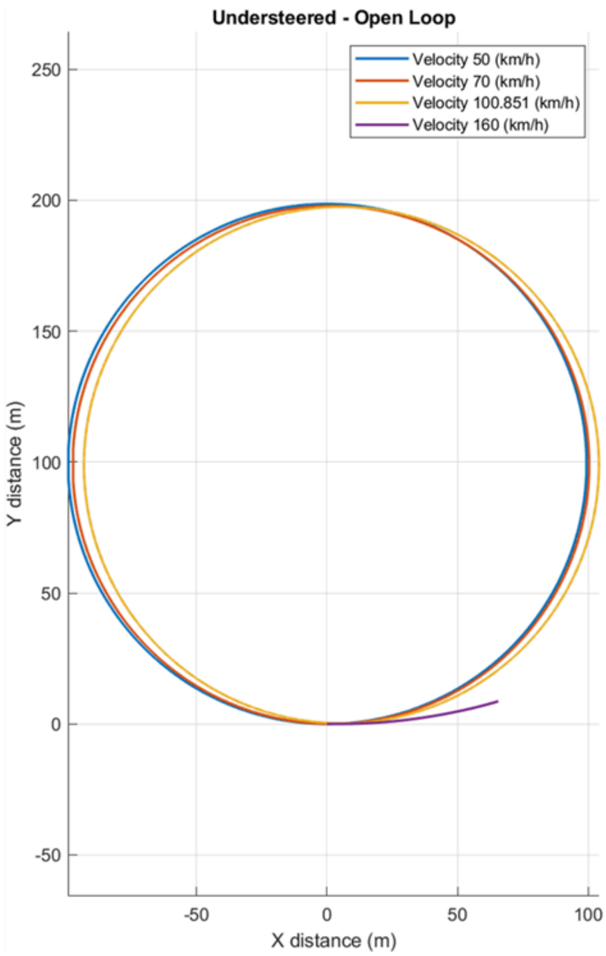

**Results:**

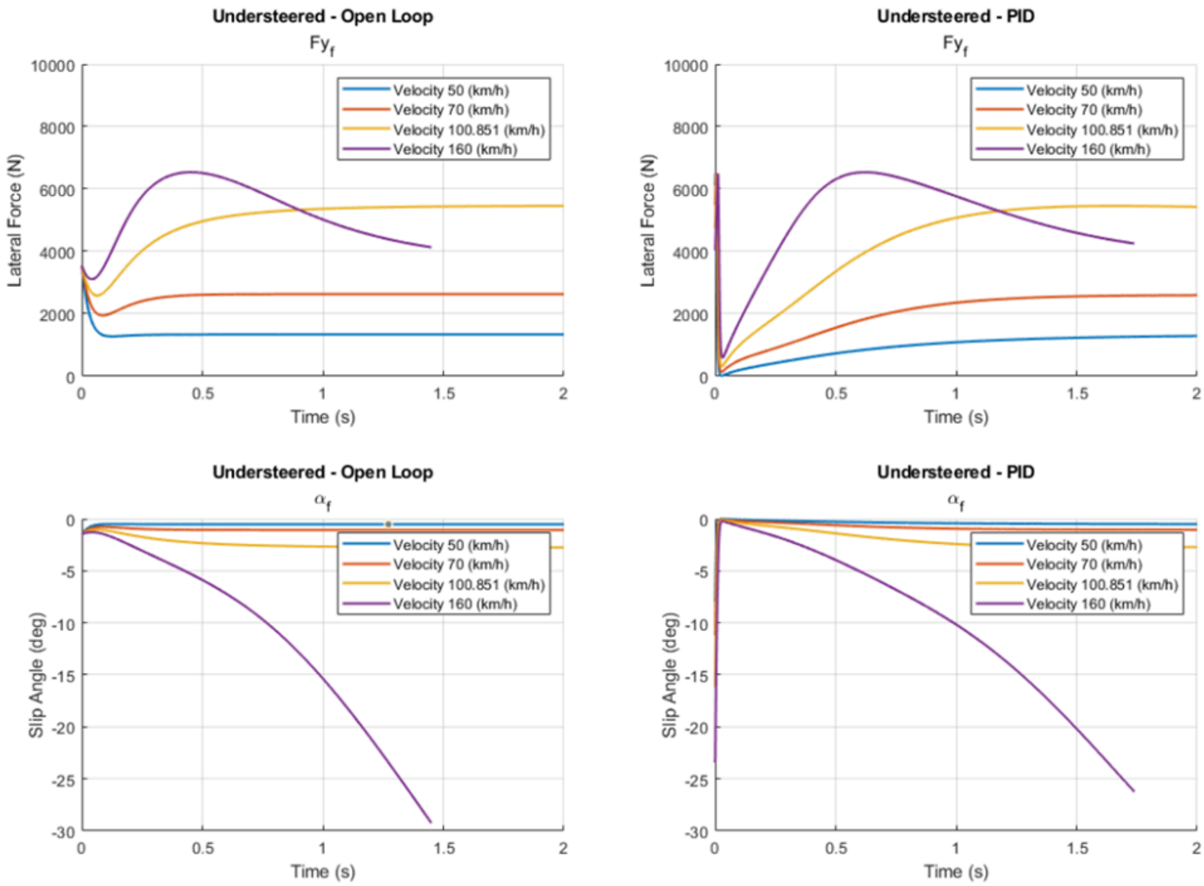
- Ø mass of the vehicle is: 1441.752 kg
- Ø Length of CG from the point start from the frame rail is: 1.862 m
- Ø L_CG_from_faxle = 1.2133
- Ø Weight distribution is : 0.46665
- Ø Height of CG is: 0.40285 m
- Ø Max traction force for the RWD vehicle is : 7810.2568 N
- Ø cost of body is: 10396.1482$
- Ø cost of battery is: 6625.92$
- Ø cost of Vehicle is: 37598.479$
- Ø Energy consumed over drive cycle US06: 1.8745 kWh
- Ø Energy consumed per km: 146.2963 Wh/km
- Ø Range: 312.3523 km

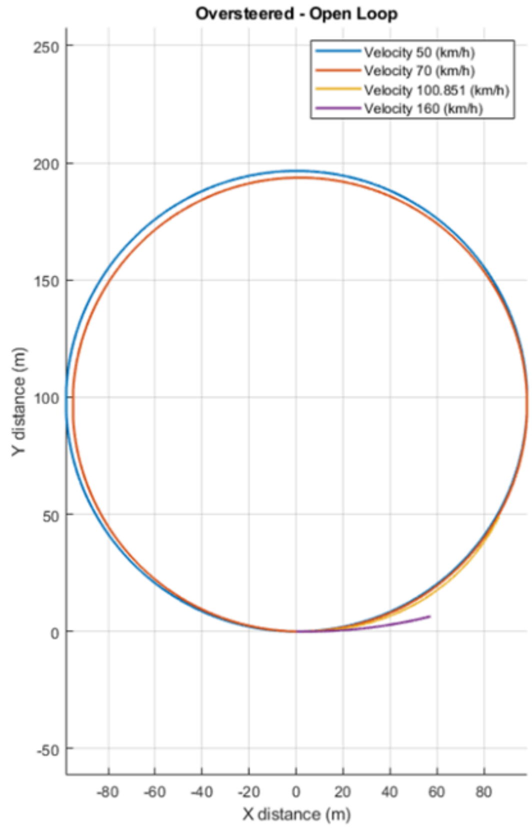
```
Ø mass of the battery is: 285.6 kg
Ø Volume of the battery is: 228.48 l or 0.22848 m^3
Ø Top speed of the vehicle is 66.1061 m/s or 237.9819 kmph
Ø Time for 0 to 100 kmph is 5.2 s
Ø Top speed of the vehicle on a slope is 58.065 m/s or 209.0341 kmph
Ø Time for 0 to 100 kmph on a slope is 5.2 s
```

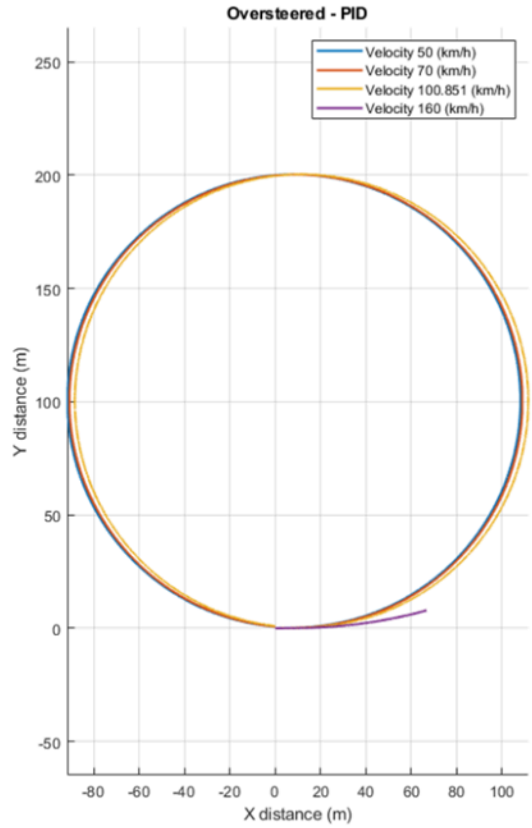
**Vehicle Dynamics Results:**

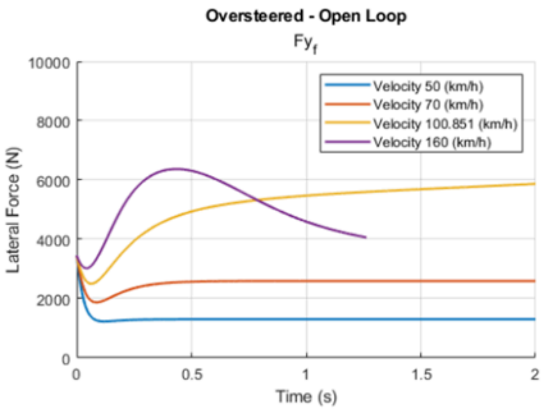


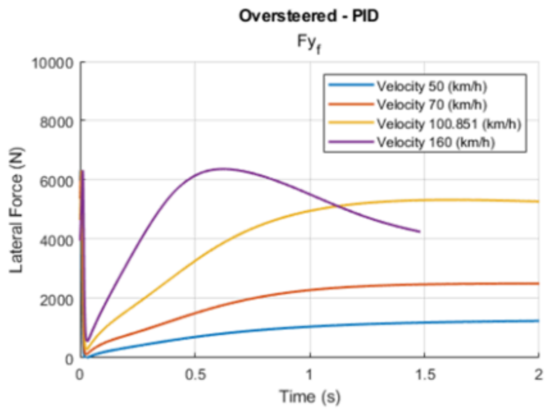



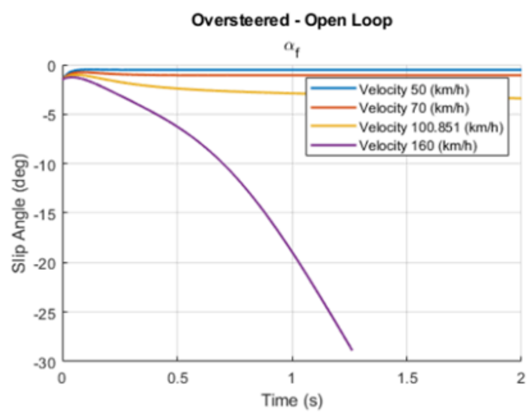

Understeered - Open Loop
Fy_f
Lateral Force (N)
Time (s)
Velocity 50 (km/h)
Velocity 70 (km/h)
Velocity 100.851 (km/h)
Velocity 160 (km/h)
Understeered - PID
Fy_f
Lateral Force (N)
Time (s)
Velocity 50 (km/h)
Velocity 70 (km/h)
Velocity 100.851 (km/h)
Velocity 160 (km/h)
Understeered - Open Loop
α_f
Slip Angle (deg)
Time (s)
Velocity 50 (km/h)
Velocity 70 (km/h)
Velocity 100.851 (km/h)
Velocity 160 (km/h)
Understeered - PID
α_f
Slip Angle (deg)
Time (s)
Velocity 50 (km/h)
Velocity 70 (km/h)
Velocity 100.851 (km/h)
Velocity 160 (km/h)

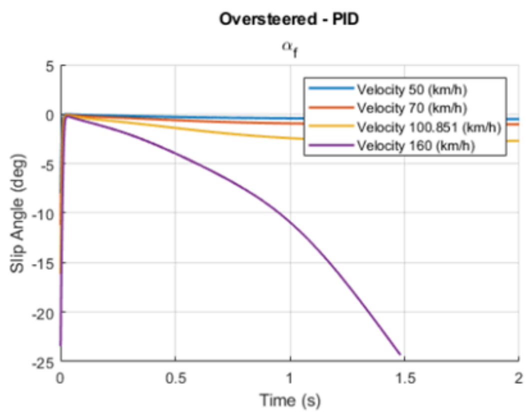

Oversteered - Open Loop
Velocity 50 (km/h)
Velocity 70 (km/h)
Velocity 100.851 (km/h)
Velocity 160 (km/h)
Y distance (m)
X distance (m)


Oversteered - PID
Velocity 50 (km/h)
Velocity 70 (km/h)
Velocity 100.851 (km/h)
Velocity 160 (km/h)
Y distance (m)
X distance (m)


Oversteered - Open Loop
Fy_f
Velocity 50 (km/h)
Velocity 70 (km/h)
Velocity 100.851 (km/h)
Velocity 160 (km/h)
Lateral Force (N)
Time (s)


Oversteered - PID
Fy_f
Velocity 50 (km/h)
Velocity 70 (km/h)
Velocity 100.851 (km/h)
Velocity 160 (km/h)
Lateral Force (N)
Time (s)

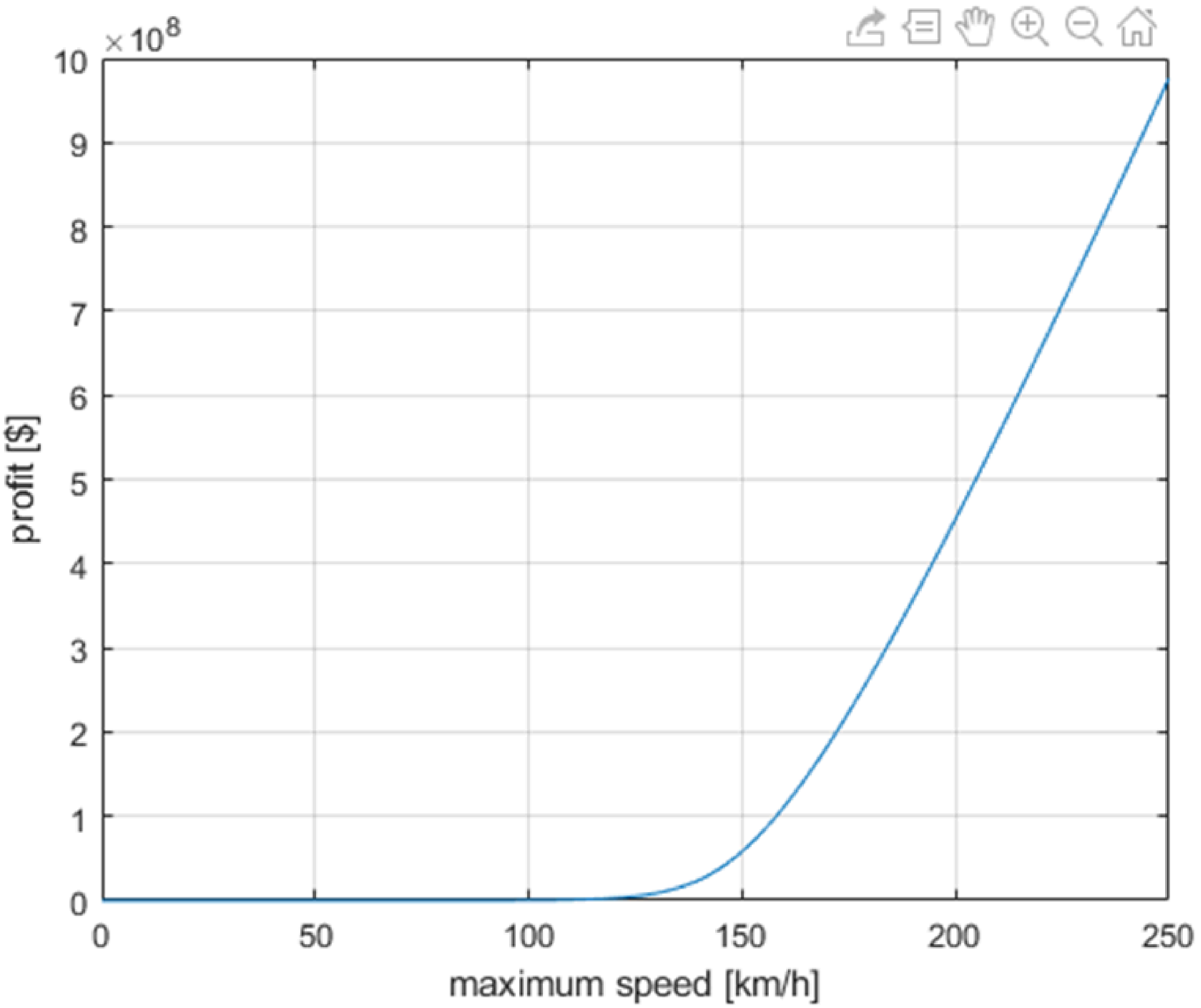

Oversteered - Open Loop
α_f
Velocity 50 (km/h)
Velocity 70 (km/h)
Velocity 100.851 (km/h)
Velocity 160 (km/h)
Slip Angle (deg)
Time (s)

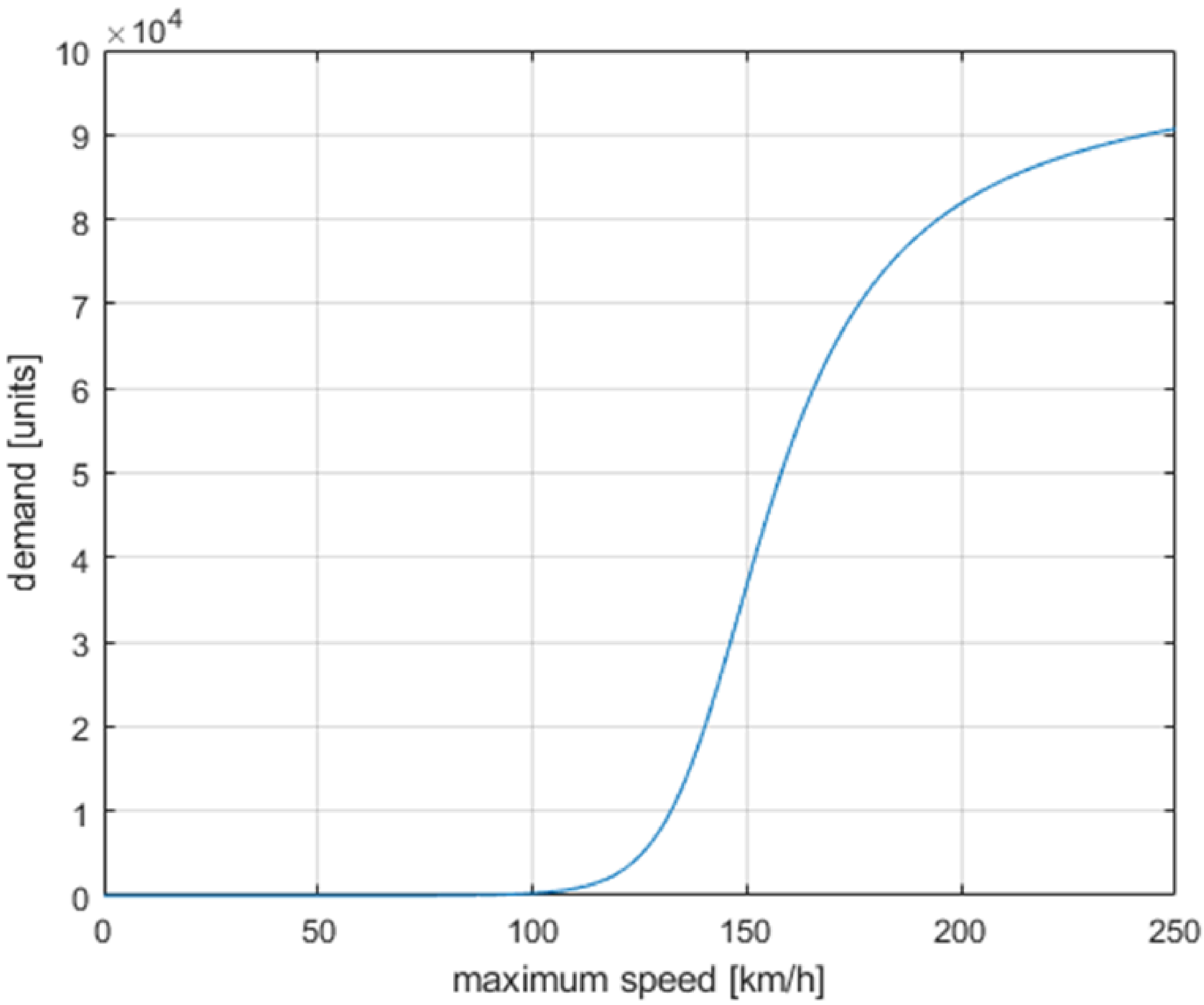

Oversteered - PID
α_f
Velocity 50 (km/h)
Velocity 70 (km/h)
Velocity 100.851 (km/h)
Velocity 160 (km/h)
Slip Angle (deg)
Time (s)

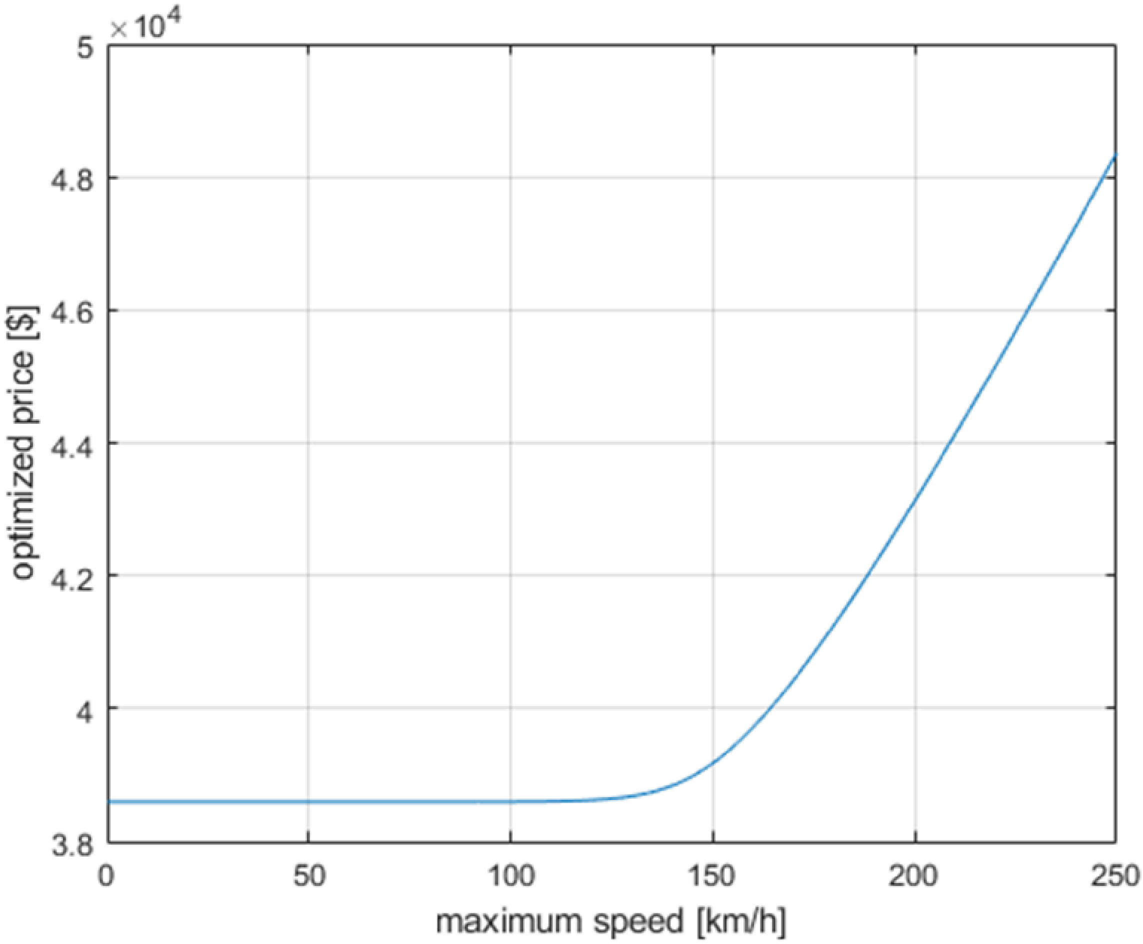
**Results:**

- Natural Frequency in front is: 1.4634 Hz
- Natural Frequency in rear is: 1.7561 Hz
- Spring Stiffness in front is: 47413.3507 N/m
- Spring Stiffness in rear is: 78034.0539 N/m
- Damping coefficient of front suspension is: 3093.8883 Ns/m
- Damping coefficient of rear suspension is: 4243.331 Ns/m
- R = 99.34 [m]

**Profit Model Results:**




×10^4
10
9
8
7
6
5
4
3
2
1
0
demand [units]
0
50
100
150
200
250
maximum speed [km/h]




**Result:**

**Ø 976.3042 million $**

## Task 3: Explore the design space to find a feasible solution

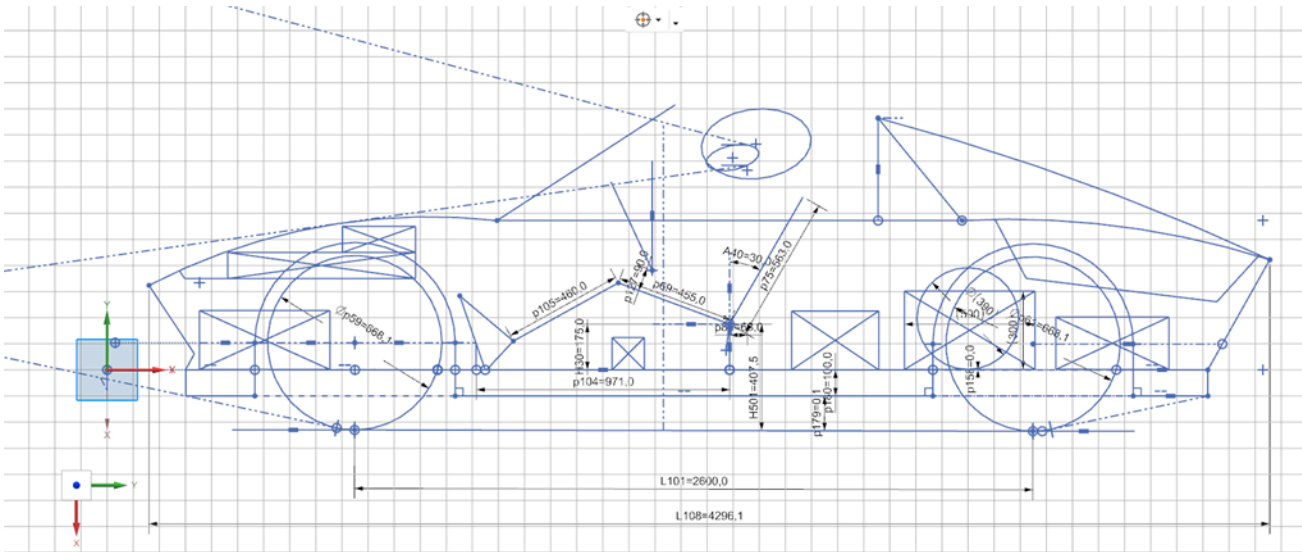
The goal of this exploration task is two-fold:

- To build up your experience in vehicle systems thinking: What will be the system-level consequences if we were to increase or decrease the value of a particular design variable?

- To build up your experience in vehicle systems engineering processes: In which order and how should we take the different requirements and design choices into account?

1. Provide a 2D sideview of your vehicle frame that illustrates the passengers, the selected components and layout requirements (i.e. vision cone, ground clearance and angle of approach/departure).
2.

2D sideview design of the vehicle with grid from Siemens NX software.




1. Describe the (partially computational) process you have used to explore the find a feasible solution.

**Design visualization:**

We had taken 1992 Honda NSX as a benchmarking vehicle and used the basic dimensions of that vehicle for designing our vehicle as the vehicle layout of NSX matches with our project vehicle.

We made the vehicle outline by using the limits of dimensions given to us in the problem statement and the packaging of the vehicle was done by using the dimensions and the data given in the occupant packaging pdf given to us.

We were able to fit all the components of the vehicle successfully in CAD (Siemens NX) by following the guidelines given to us and later finalized model was plotted in MATLAB by using Design Visualization code.

**Powertrain Selection:**

We did battery selection and motor selection based on our requirements by doing the benchmarking and undercutting the competitors model given to us in the profit model. The vehicle was able to achieve the desired performance while keeping the cost as low as possible.

The energy requirement for US06 is 2.0438kW over a distance of 12812.7 m or 12.812 km, i.e., **159.5171Wh/km.**

Now that we have the energy requirement for on a per kilometer basis we can set a desired range, in our case we chose to go as soon as the top limit keeping the battery stacks of 192 intact. We finally got the number of stacks in series and number of stacks (192 cells) in parallel,

Number of stacks in series: 1

Number of stacks in parallel: 14

Capacity for 14 stack was calculated based on per cell capacity and number of cells in 14 stacks of battery, **192*14*17Wh= 45.696kWh**. which is giving us a range of approx. 312 km. For the capacity rating of 2C and 5C, since the motor will be requiring the discharge of 5C during 0-100km time test or during heavy acceleration, the max power battery can give is saturated by the top value of 5*45.696kW, if the power was saturated at 2C the motor could not have extracted the discharge current required to operate at 250kW peak power.

**SFBM Calculations:**

We selected the beam dimensions by comparing the roadster vehicles available in the market and got the range of dimension of beams and used them to design our own beam and reached at the set of dimensions of the beam which were acceptable while satisfying the stress, bending moment and deflection criteria given to us.

**Vehicle dynamics:**

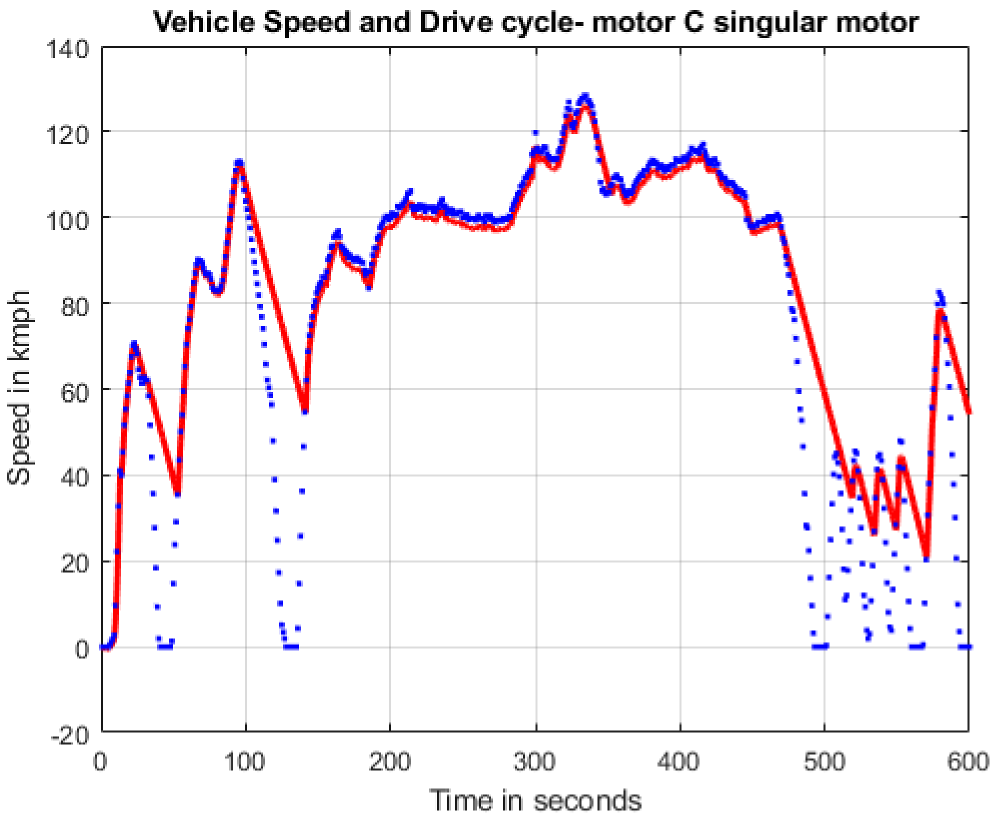
We build upon the model that we did in vehicle dynamics assignment and used the value of resonant frequency given to us and calculated all the parameters successfully like spring coefficient, Damping coefficient, turning radius of the vehicle while meeting all the requirements like the weight distribution which is in the acceptable and range and designed our car to be understeering.

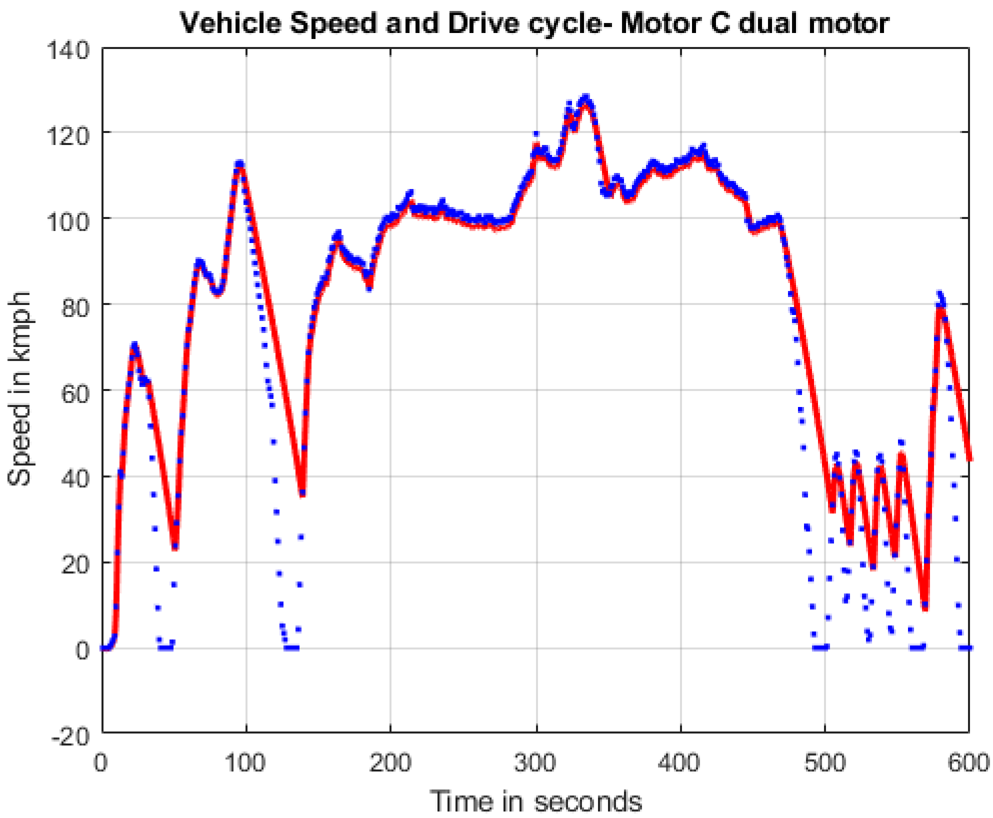
**Profit model:**

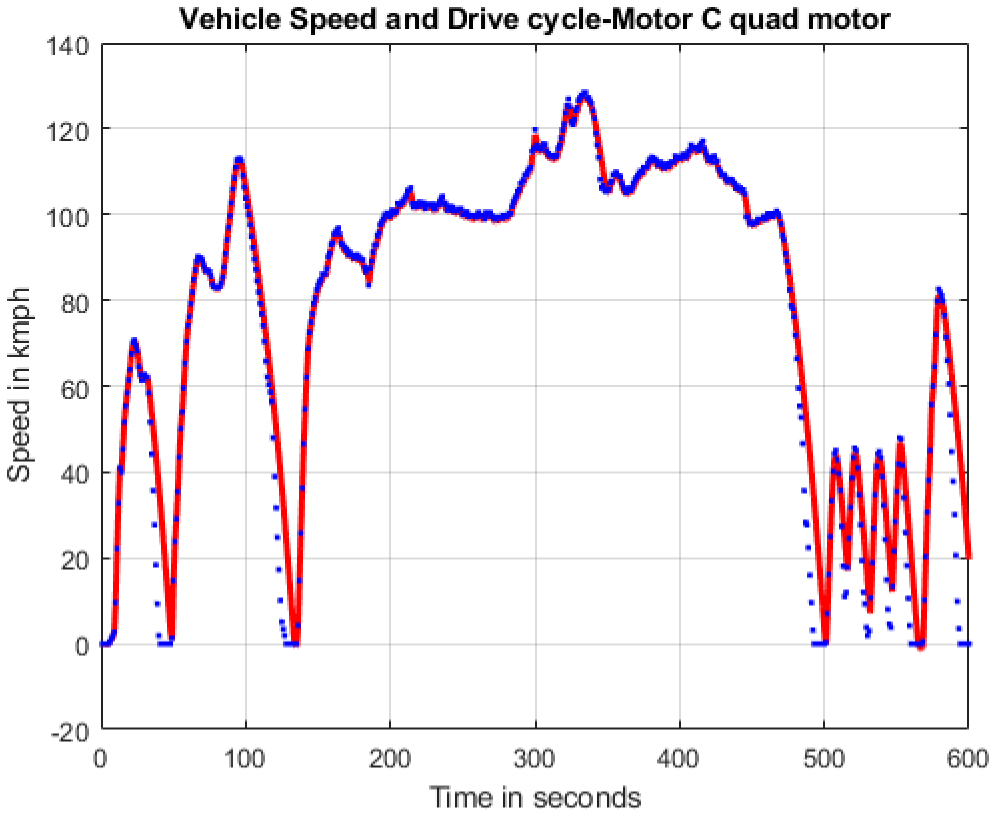
We used the profit model given to us which has the data of competitor's car and upper limits of the profit function which we should not exceed. We successfully designed our vehicle in such a way that it would give us the maximum performance for that particular setup while keeping the cost at minimum, this helped us to increase profit.

3. Describe the process you have used to improve on or optimize the feasible solution you started with. Use results from different design variable choices (i.e. motor selections, battery capacity, final drive ratio, etc.) to justify the feasible solution you selected.

Motor C lacked in area of low-speed high torque requirement. We have tried multiple iterations by tuning the PID controller and changing the values for the final drive ratio to match and fulfill the high torque requirement at the lower speed. But changing the final drive ratio did not fulfill our requirement and after an extent the speed requirement of the drive cycle was not being fulfilled as the final drive ratio was too high, as the result shown below none of the motor were able to meet the torque requirement for the drive cycle expect the quad configuration of motor C, which barely makes the requirement. but the price of the quad motor configuration will drive the demand to zero as the price will be too high.


Vehicle Speed and Drive cycle- motor C singular motor
140
120
100
80
60
40
20
0
-20
Speed in kmph
0
100
200
300
400
500
600
Time in seconds


Vehicle Speed and Drive cycle- Motor C dual motor
140
120
100
80
60
40
20
0
-20
Speed in kmph
0
100
200
300
400
500
600
Time in seconds




For sizing of the battery, the vehicle is equipped with one motor ‘A’ and its inverter and is run over the drive cycle US06, the energy requirement of the drive cycle is calculated by integrating the power required for the vehicle with motor ‘A’ to run over drive cycle US06.

The energy requirement for US06 is 2.0438kW over a distance of 12812.7 m or 12.812 km, i.e., **159.5171Wh/km.**

Now that we have the energy requirement for on a per kilometer basis we can set a desired range, in our case we chose to go as soon as the top limit keeping the battery stacks of 192 intact. We finally got the number of stacks in series and number of stacks (192 cells) in parallel,

Number of stacks in series: 1

Number of stacks in parallel: 14

Capacity for 14 stack was calculated based on per cell capacity and number of cells in 14 stacks of battery, **192*14*17Wh= 45.696kWh**. which is giving us a range of approx. 312 km. For the capacity rating of 2C and 5C, since the motor will be requiring the discharge of 5C during 0-100km time test or during heavy acceleration, the max power battery can give is saturated by the top value of 5*45.696kW, if the power was saturated at 2C the motor could not have extracted the discharge current required to operate at 250kW peak power.

For the Final drive ratio, we firstly prioritized the factor that torque demand of the drive cycle should be met, so that the vehicle speed and desired speed are same till the 2 decimal places. For this, we found out the range of the final ratio in which we can play around, trading off top speed for the better 0-100kmph time while our torque demand is still being met. For Motor A this range was “fd=3.8” to “fd=10.5”, after this value the torque demand for the drive cycle was not being fulfilled for the lower end speed operation or the higher end speed operation for the lower ratio of final drive ratio and higher ratio of final drive respectively. We shortlisted the narrow range of final drive ratio of 5.0 to 5.7 trying to keep the specifications of our vehicle better than the competitor and also maximizing the profit as profit is directly affected from the 0-100kmph acceleration time and top speed. We were getting the highest profit for the "fd=5.0".

4. Include the final (optimized) design you have obtained.

| **PRELIMINARY Variables Values for the Optimized design** | |
|---|---|
| Configuration | **Rear Wheel Drive Single Motor A** |
| Motor_inverter_select | **Motor-Inverter A (UQM PowerPhase HD250)** |
| Battery_capacity | **0.22848 m^3** |
| Drive_ratio | **5** |
| Profit | **$ 976.3042 Million USD** |
| Cost of the vehicle | **$ 37598.479 USD** |
| Acceleration Time | **5.2 seconds for 0-100 kmph** |
| Max_speed | **237.9819 kmph** |
| Range_battery | **312.3523 km** |
| Max_Speed on Slope_result | **209.0341 kmph** |
| Max. Bending Stress | **111.952310 MPa** |
| Deflection_max | **-4.911665 mm** |
| Lateral_acceleration_max | **7.8 m/s^2** |
| Radius of curvature, R | **99.34 m** |
| Turning _result | **Pass, as the R is less than 100m** |
| Spring Stiffness_rate_front | **47413.3507 N/m** |
| Spring_Stiffness_rate_rear | **78034.0539 N/m** |
| Spring_damping_coefficient_front | **3093.8883 Ns/m** |
| Spring_damping_coefficient _rear | **4243.331 Ns/m** |