AuE 881

AUTOMOTIVE SYSTEMS OVERVIEW

FINAL PROJECT

DESIGNING A BATTERY-ELECTRIC VEHICLE

CLEMSON UNIVERSITY INTERNATIONAL CENTER FOR AUTOMOTIVE RESEARCH

Report Guide and Content

The report is divided into 4 major sections based on the tasks we have in this project, each section head, and subsection will have a different color code.

Section 1: **Task 1: ORIENTED AND GET ORGANIZED**

1. List of objectives, performance measures, design variables, intermediate variables, and parameters.
2. Variable computation
3. Computational order

SECTION 2: **TASK 2: MODEL CREATION**

SECTION 3: **TASK 3: EXPLORE THE DESIGN SPACE AND THE OPTIMIZED DESIGN**

1. Frame Rail Preliminary Design
2. Design Space Exploration
   FWD1 | FWD2 | RWD1 | RWD2 | AWD2 | AWD4
3. Summary, the optimized design, and the 2D layout

SECTION 4: **Appendix: MATLAB code and the Iterations**

1. Sample of the iterations' solutions
2. The MATLAB codes

In case any of the codes is not clear, please refer to the code at the appendix.

Thanks.

TASK

1

GET ORIENTED AND GET ORGANIZED

List of objectives, performance measures, design variables, intermediate variables, and parameters

Objective: maximize profit for the enterprise

Performance measures:

#	Performance Measures
1	0-100km/h acceleration time
2	range on one battery charge
3	top speed
4	the cost
5	the speed at 12% slope
6	material stress in the frame rail under normal loading conditions
7	maximum deflection of the frame rail
8	lateral acceleration on 100m diameter skidpad
9	The max torque and motor speed (power) at any point of operations
10	the ride frequency in the front
11	the ride frequency in the rear

Design variables:

#	Design Variables
1	Battery capacity in [kWh]
2	Choice of electric motor & inverter pair: A [Y or N], B [Y or N], C [Y or N]
3	Motor and driveline configuration: FWD 1 motor [Y or N], FWD 2 motor [Y or N], RWD 1 motor [Y or N], RWD 2 motors [Y or N], AWD 2 motors [Y or N], AWD 4 motors [Y or N]
4	Final drive ratio
5	2-D vehicle layout [mm]: L104, L105, L108, H101, H5-1, and H5-2

6	Frame rail
7	Suspension spring constant in [N/m] front: 3412 N/m rear: 4913 N/m
8	Suspension damping coefficient in [Ns/m]

Intermediate variables:

#	Intermediate Variables
1	Mass
2	Volume
3	Drag force
4	Friction force
5	Rolling force
6	Weight distribution
7	Energy consumption per 100 km

Parameters:

#	Parameter type	Parameters
1	Mass:	4 AM95 passengers (101kg/passenger) = 404
2	Mass:	Cargo per EPA (7kg/passenger)=7*4=28
3	Mass:	Infotainment =15kg
4	Mass:	Sunroof =10kg
5	Mass:	Unsprung masses = 60kg/corner = 60*4 = 240 kg
6	Mass:	Gearbox (single-ratio): for low-torque gearbox (torque<300Nm) = 28kg
7	Mass:	Gearbox (single-ratio): for high-torque gearbox (torque>300Nm) = 45kg
8	Mass:	Differential = 10kg
9	Mass:	Axles (5kg per half shaft) = 5*4 = 20 kg
10	Mass:	Body (Length X height x 50kg/m2)
11	Mass:	Frame rail : 3*volume of the frame rail in the side view*7800
12	Mass:	If selected, prop shaft = 10kg/m
13	Mass:	Battery = 250Wh/kg
14	Mass:	Motor-Inverter A = 85 kg + 35 kg

15	Mass:	Motor-Inverter B = 41 kg + 16 kg
16	Mass:	Motor-Inverter C = 55 kg + 12 kg
17	Cost:	Misc. Electronics: $3000
18	Cost:	Wheel Assemblies: $450/corner
19	Cost:	Gearbox (single-ratio): $4000 for low-torque gearbox (torque<300Nm)
20	Cost:	Gearbox (single-ratio): $7500 for high-torque gearbox (torque>300Nm)
21	Cost:	Differential: $600
22	Cost:	Axles: $350
23	Cost:	Body: cost is proportional to bounding-box volume: 1000$/m3
24	Cost:	Frame rail: cost is proportional to mass: $2/kg
25	Cost:	Battery: cost is proportional to capacity: $145/kWh
26	Cost:	Prop shaft: cost is proportional to length: $600/m
27	Cost:	Motor-Inverter A: $7500
28	Cost:	Motor-Inverter B: $5200
29	Cost:	Motor-Inverter C: $5500
30	Given:	Gravitational acceleration g = 9.81 m/s2
31	Given:	Drag coefficient Cd = 0.27
32	Given:	W101 = 1550 mm
33	Given:	W103 = 1820 mm
34	Given:	Tire size = 225/45R18
35	Calculated:	Tire Diameter = 659.7 mm
36	Given:	Density of air ρ = 1.26 kg/m3
37	Given:	cross-sectional area of the vehicle : A = W103*H101
38	Given:	damping is at 30% of critical damping
39	Assumption	Frame rail material is AISI 1020 Steel and the below mechanical properties are estimated based on that except the density which is given already:
40	Assumption	Young's Modulus E=2.15e+11 Pa
41	Assumption	Shear Modulus G=78000 N/mm2
42	Assumption	Rolling coefficient Cr=0.009
43	Assumption	Coefficient of friction (dry asphalt) = 0.9
44	Assumption	The density of the frame rail material is 7800 kg/m3

Variables computation

Below is the sequence of computing every variable and the calculation method:

1. Vehicle Mass

Calculation method: The total mass is the summation of the below values

Input	Calculations Comment	unit	Dependency
Passengers mass	4 (passengers) x 101 (kg/passenger)	kg	
Cargo mass	4 (passengers) x 7 (kg/passenger)	kg	
Infotainment mass	15 kg	kg	
Sunroof mass	10 kg	kg	
Unsprung mass	60 (kg/corner) x (4 corners)	kg	
Gearbox mass	if (gb_low_torque = 1) gearbox_mass = 28 (kg) else if(gb_high_torque = 1) gearbox_mass = 45(kg) end	kg	
Differential mass	10 (kg)	kg	
Axles mass	5 (kg/half shaft) x 4 (half shafts)	kg	
Body mass	(v_length x v_height) x 50 (kg/m2)	kg	Vehicle length (2D) Vehicle height (2D)
Frame rail mass	3 x 2x volume of the frame rail (one side) *7800	kg	Design of the frame rail
Prop shaft mass	if prop_shaft=1 prop_shaft_mass=10 x prop_shaft_length	kg	prop_shaft_length
Battery mass	250 (Wh/kg) x battery_capacity	kg	battery_capacity

Motor-Inverter mass	if motor_inverter_type=A
	motor_inverter_mass=85+35;
	else if motor_inverter_type=B
	motor_inverter_mass=41+16;
	else if motor_inverter_type=C
	motor_inverter_mass=55+12;
	end
	total_motor_inverter_mass=motor_inverter_mass*num-motors
		kg	Motor inverter type

and so on			

2. Frame Rail Dimension (Max Stress, and Max deflection)

Calculation method: the max moment will be calculated based on the SFBM function. The inputs listed below are dependent on other decisions as mentioned before. Accordingly, the worst-case scenario would be assumed here with a preliminary layout.

Input	Calculations Comment	unit	Dependency
Preliminary 2D layout	Assumed; all dimensions will be drive from here	-	
Configuration	AWD	-	
no of motors	4	-	
Motor type	A	-	

After having the max bending moment from the solver, the next step will be determining the max stress:

$$\sigma = \frac{M_{max} y}{I}$$

where:
sigma: is the stress
M: the bending moment
I: the second moment of inertia
y: distance from the neutral axis, in this case will be assigned h/2 to get the max stress.

For the max deflection, a simplification assumption will be made of having a concentrated load equals the weight of the vehicle applied at the center distance of the wheel base. This assumption creates much higher load compared to the actual load, and accordingly results higher deflection compared to the real one. So, if the frame design could survive this load, it could survive the real one.

The reason for choosing this way for calculating the deflection is saving the time while creating a complicated model to compute this output.

3. CG calculations (CG to front axle over the wheelbase ratio)

Calculation method: the only inputs are the component masses and the vehicle layout, then the CG equations used for the horizontal and vertical distances of the CG. Then, the ratio of the CG to front axle over the wheelbase length is calculated (one of the requirements)

Input	Calculations Comment	unit	Dependency
Preliminary 2D vehicle layout	Assumed	-	
Mass of each item	.	kg	

Volume of each item	.	m^3	

4. Vehicle Costing

Calculation method: the summation of all items will be used.

Input	Calculations Comment	unit	Dependency
Cost of each item	Assumed	$	
Design variables	Based on these variables the cost of each item should change, like the cost of the motor(s) and gearbox(s), based on the configuration and number selected.	kg	

5. Vehicle Performance Model (max speed, 0-100km/h time, range, 12% slope)

This Simulink model will be used to calculate every requirement/performance characteristics related to the vehicle performance in the longitudinal direction like the max speed, 0-100 km/h time, .. etc.

The model sill consists of submodels working on each part of the vehicle starting from the road till the motors as below:

5.1 Road Loads submodel

This model will calculate the vehicle velocity and acceleration, based on the traction forces, drag force, rolling resistance force, and the weight component of the vehicle in case of slope.

Resultant force = Traction force - Drag force - Rolling Force - Longitudinal weight component

5.1.1. Rolling Force

Calculation method: The rolling force is mainly dependent on the rolling coefficient Cr, tire normal load, from the below equation,

$$\text{Rolling Force} = C_r \,.\, \text{Tire Normal Force}$$

Input	Calculations Comment	unit	Dependency
Tires reactive forces	based on the weight distribution	N	Reactive force
Rolling coefficient	0.009	-	

5.1.2 Drag Force

Calculation method: based on the below equation

$$F_d = \frac{\boldsymbol{\rho}.\mathrm{A}.\mathrm{C_d}.\mathrm{V}^2}{2}$$

Input	Calculations Comment	unit	Dependency
Air density	Density of air ρ = 1.26 $kg/m3$	$kg/m3$	
Cross Sectional Area	A = W103*H101 = 1.82 * H101	m2	
Drag coefficient	Drag coefficient Cd = 0.27		
Vehicle speed		m/s	Vehicle speed calculations

5.1.3 Traction Force

This traction force depends on the motor performance, driveline, and reactive forces on the wheels, which will be determined in the next sections.

5.1.4 Longitudinal weight component

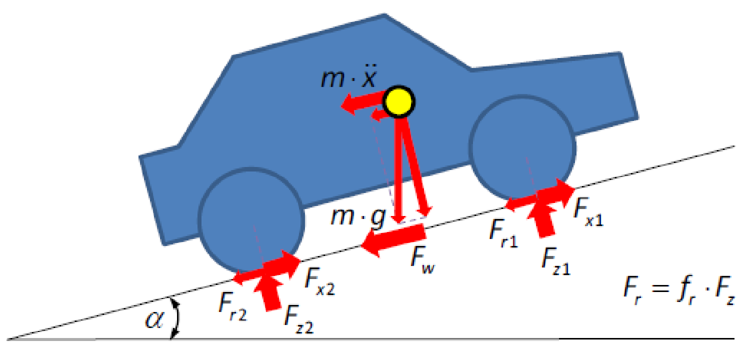
In case of the vehicle running over a slope, an additional force in the longitudinal direction will be considered which is the weight component of the car.

Longitudinal weight component = sprung mass * g * sin(slope angle)

5.1.5 Vehicle longitudinal acceleration

Based on the above the vehicle acceleration could be calculated as below.

<u>Calculation method:</u> based on the above equations, the vehicle acceleration could be calculated as below (the screenshot is from lecture notes):



$$m.a = (F_{x1} + F_{x2}) - F_{rolling} - F_w - mg.\sin(\boldsymbol{\theta})$$

$$a = \frac{(F_{x1} + F_{x2}) - F_{rolling} - F_w - mg.\sin(\boldsymbol{\theta})}{m}$$

Input	Calculations Comment	unit	Dependency
Traction Forces	The driveline and powertrain calculations model	N	Driveline and powertrain calculations
Rolling force	Rolling Force calculations	N	Rolling force

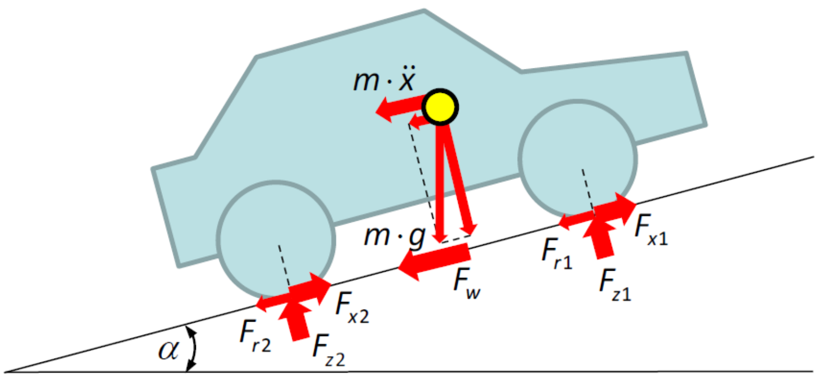
Drag Force	Drag Force calculations	N	Drag Force
Vehicle mass	Vehicle mass calculations	kg	Vehicle mass calculations
g	parameter	m/s2	
Slope	input	degrees	
Vehicle mass	Vehicle mass model	kg	vehicle mass calculations

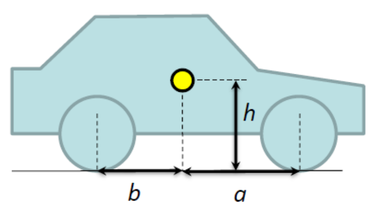
5.1.6 The vehicle longitudinal velocity

Calculation method: It will be directly computed as an integration of the acceleration of the vehicle.

5.2 Tires reactive forces submodel

Calculation method: It could be calculated based on the vehicle mass, slope, vehicle acceleration, and the dimensions of the vehicle.





The equation for the front normal force Fz1 is as below:

$$F_{z1} = \frac{mg.cos(\theta).b\ -\ m.\ddot{x}.h\ -\ \mathrm{m.g.sin}(\theta).\mathrm{h}}{wheelbase\_length} = \frac{mg.cos(\theta).b\ -\ m.\ddot{x}.h\ -\ \mathrm{m.g.sin}(\theta).\mathrm{h}}{2.7}$$

for the rear normal force Fz2, it could be calculated from the below equation:

$$F_{z2} = \frac{mg.cos(\theta).a\ +\ m.\ddot{x}.h\ +\ \mathrm{m.g.sin}(\theta).\mathrm{h}}{2.7}$$

Input	**Calculations Comment**	**unit**	**Dependency**
Vehicle Mass	Vehicle Mass calculations	*kg*	Vehicle Mass calculations
g	9.81	ms2	
Road slope		degrees	input

Distance between rear axle and CG	b	m	Vehicle layout, and CG calculations
Distance between front axle and CG	a	m	Vehicle layout, and CG calculations
CG height	CG_height	m	Vehicle layout, and CG Ẋcalculations
Vehicle acceleration	ẍ	m/s2	Vehicle Accelerations calculations

5.3 Driveline submodel

This model represents the driveline from the motor till the wheels, it includes braking forces, drive ratio, traction limit, and forces distribution.

5.3.1 Forces distribution

The distribution of the forces depends on the configuration of the car, depending on the configuration, the traction force will be directed to a specific axle.

if configuration=”RWD”
Traction force rear = traction force
Traction force Front = 0

else if configuration=”FWD”
Traction force Rear =0
Traction force Front = traction force

else if configuration=”AWD”
Traction force Rear =Traction force / 2
Traction force Front = Traction force / 2
end

Then, this traction force is multiplied by the number of motors per each axle.

5.3.2 Traction limit

Calculation method: This to be calculated based on the weight distribution, the slope, and the vehicle configuration.

Traction limit Rear = Traction coefficient * Fz2
Traction limit Front = Traction coefficient * Fz1

Input	Calculations Comment	unit	Dependency
Wheel Normal Force	From wheels normal force calculations	N	wheels normal force model
Traction coefficient	Assumed		
configuration			input

5.4 Electrical Motor submodel

This model includes no equations, it will require lookup of the proper torque, and efficiency based on the motor speed.

5.5 Battery Range submodel

The battery range calculations are dependent on the distance travelled, energy consumption, and battery capacity, and the assumed useful capacity.

The distance will be the integration of the vehicle velocity calculated before. At the same time the below variable will be monitored through the modeling and once it equals zero, this would be the range of the battery.

Remaining energy = (Battery capacity * useful capacity %) - Energy consumed

The energy consumed is the integration of (the motor power at any point / motor efficiency at the same point).

6. Profit Calculation

As by this time we should have the cost, 0-100km/h time, max speed, and range, we should be ready to go for profit calculation. For the profit calculations, the profitPredict will be used with the previous stated 4 inputs.

7. Vehicle Dynamics Model (Understeering, Kus value, turning radius, and max lateral acceleration)

The vehicle dynamics model (bicycle model) created in HW4 will be used here with some slight modifications. The purpose will be assuring that the vehicle UNDERSTEERS, then assign a constant turning angle value, and increase or decrease the longitudinal speed till reaching the required diameter with lateral acceleration at some point higher than 0.8 g.

As long as the vehicle understeers, the speed adjustment along with the steering input will definitely lead to a specific turning diameter without unexpected performance. In case the max lateral acceleration was low compared to the 0.8g. This could be adjusted by increasing the turning angle, and the speed.

As in the model the vehicle runs at constant speed, there would be no need to consider weight transfer, or any other complications from the performance model. The required inputs here will be the CG location, and some assumptions about the tire stiffness values.

8. Suspension Spring Constant and damping coefficient

Source (1): http://downloads.optimumg.com/Technical_Papers/Springs%26Dampers_Tech_Tip_1.pdf
Source (2): https://optimumg.com/spring-dampers-part-four

Spring Constant

Starting with the basic equation from physics, relating natural frequency, spring rate, and mass:

$$f = 1/(2\pi)\sqrt{\frac{K}{M}}$$

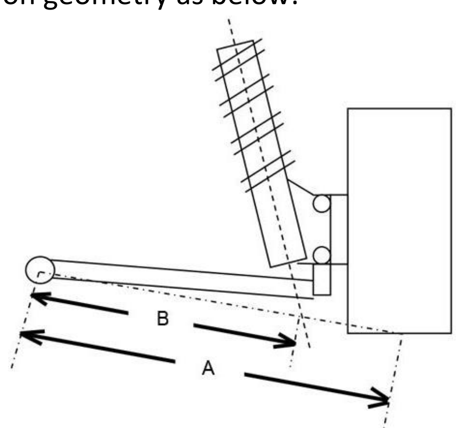
f = Natural frequency (Hz)
K = Spring rate (N/m)
M = Mass (kg)

Solving for spring rate, and applying to a suspension to calculate spring rate from a chosen ride frequency, measured motion ratio , and mass :

$$K_s = 4\pi^2 f_r^2 m_{sm} MR^2$$

K_s = Spring rate (N/m)
m_{sm} = Sprung mass (kg)
f_r = Ride frequency (Hz)
MR = Motion ratio (Wheel/Spring travel)

MR is determined by the suspension geometry as below:



For simplicity, we assume MR of 1.
To calculate the spring rate, we will solve for the sprung mass per each corner, and the required frequency (front and rear).

Damping Coefficient

The spring damping coefficient could be estimated from the following.

$$\text{Initial Slope} = (4\pi\zeta_{ride}\omega_{ride}m_{sm}) \quad N/(m/s)$$

ζ_{ride} = Damping ratio in ride
ω_{ride} = Ride frequency (Hz)
m_{sm} = Sprung mass supported by damper (kg)

Computational Order

To summarize the computational order, it will be as follows:

1. Vehicle mass
2. Frame Rail Dimension (Max Stress, and Max deflection)
3. CG calculations (CG to front axle over the wheelbase ratio)
4. Vehicle costing
5. Vehicle Performance Model (max speed, 0-100km/h time, range, 12% slope)
6. Profit calculations
7. Vehicle Dynamics Model (Understeering, Kus value, turning radius, and max lateral acceleration)
8. Suspension Spring Constant and damping coefficient

TASK

2

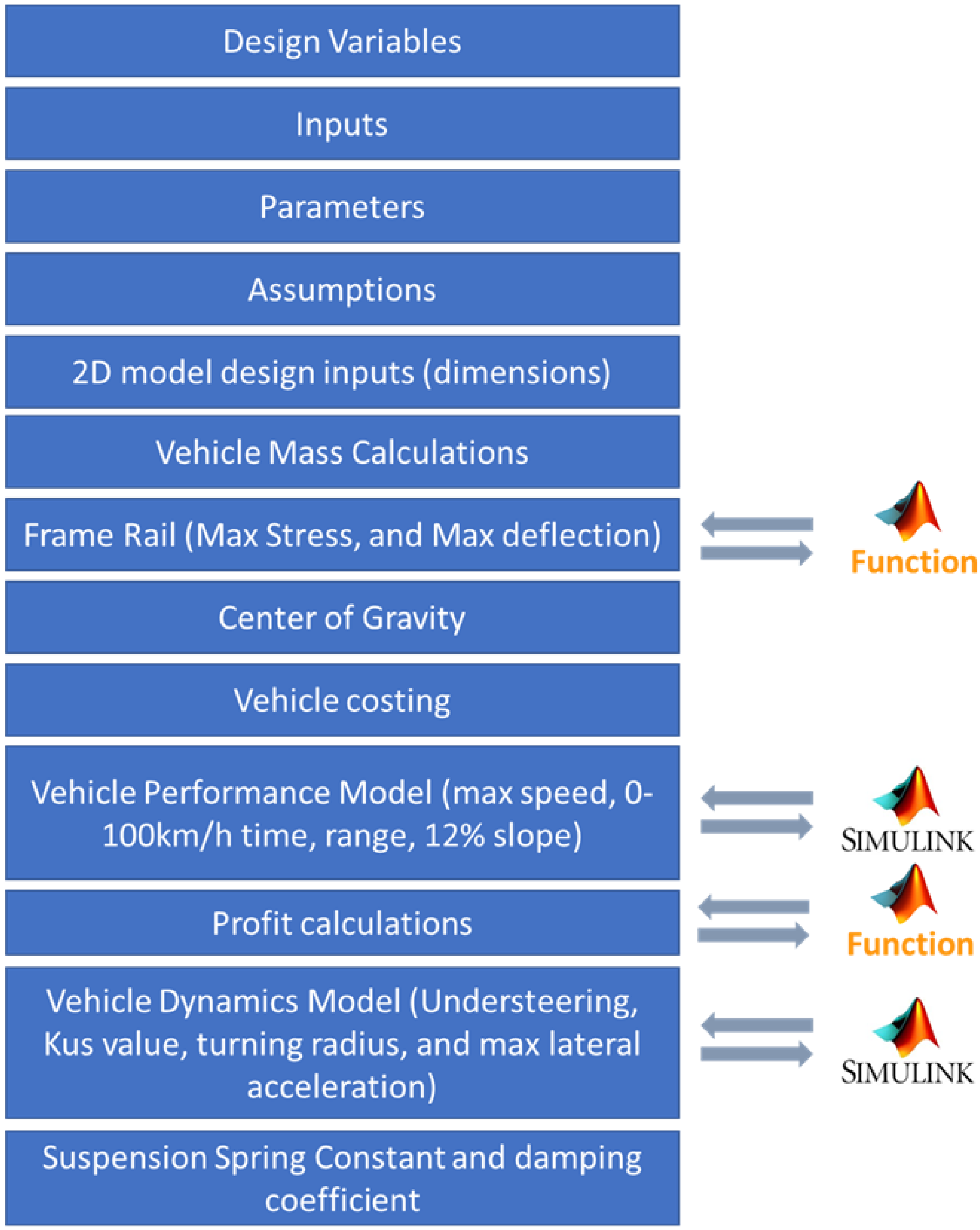
CREATE THE MODELS TO EVALUATE EACH OF THE REQUIREMENTS AND OBJECTIVE

All requirements, calculations, and objectives are calculated in only one main code, which sometimes perform just simple calculations, call functions, or run a Simulink model. The reason for trying to have everything in one code is making it simpler for the optimization and discovering the design space task. In this later task, a range of the design variable could be added to the start of this main code, and some indexing functions at the end of it, and all the results for one iteration could be saved, and we can run many iterations.

The below variables are outputs need to be calculated to evaluate the design (i.e meeting the requirements or needed for calculating the profit):

1. Profit
2. Cost
3. 0-100 km/h time
4. range of the battery
5. top speed
6. The ability to ascend a 12% slope at 30 km/h
7. The max stress in the frame rail
8. The max deflection in the frame rail
9. Vehicle dynamics requirements
10. Spring Constant
11. Spring Damping Coefficient
12. CG location (CG_a/L101)

Based on the above the below code structure is created.



Below each model / code section will be presented.

Design Variables

```
%% Design Variables
battery_capacity=30000;             %battery capcity in Wh
motorinverter_select='A';           %A, B, or C
configuration='FWD1';               %FWD1, FWD2, RWD1, RWD2, AWD2, or AWD4
drive_ratio=7;                      %final drive ratio
w=0.090;                            %(m) frame rail width
h=0.180;                            %(m) frame rail height
t=0.001;                            %(m) frame rail thickness
frame_rail_area=(w*h)-((w-(2*t))*(h-(2*t)));

if ((configuration =='FWD1') | (configuration =='RWD1'))
    num_motor=1;
    num_gearbox=1;                  %two motors could be connected to the samce GB to double the torque
    num_inverter=1;
    num_differential=1;
elseif ((configuration =='FWD2') | (configuration =='RWD2'))
    num_motor=2;
    num_gearbox=1;                  %two motors could be connected to the samce GB to double the torque
    num_inverter=2;
    num_differential=1;
elseif (configuration =='AWD2')
    num_motor=2;
    num_gearbox=2;
    num_inverter=2;
    num_differential=2;
elseif configuration =='AWD4'
    num_motor=4;
    num_gearbox=2;
    num_inverter=4;
    num_differential=2;
end
```

Inputs

```
%% Inputs
slope=0;                            %the road slope in %
```

Parameters

Please note that all the inputs are dependent on the design variables, like the motor type and the configuration

```
%% Parameters
g=9.81;                                %Gravitational acceleration g = 9.81 m/s2
Cd=0.27;                               %Drag coefficient Cd = 0.27
r=659.7/1000/2;                        %for Tire size = 225/45R18, the diameter = 659.7 m
rho=1.26;                              %Density of air ρ = 1.26 kg/m3
steel_density= 7800;                   %kg/m^3

frf=1.2;                               %Front ride requency (Hz)
frr=1.44;                              %Front ride requency (Hz)
damping_ratio=0.3;                     %damping ratio

% Dimensions
battery_width=1.25;                    %max battery width
W103=1820/1000;                        %vehicle width (m)
W101=1550/1000;                        %Tread_front
L101=2700/1000;                        %the wheel base length

if motorinverter_select=='A'           %%motor, inverter, and gearbox paratmers
    motor_mass=85;                     %kg
    inverter_mass=35;                  %kg
    motorinverter_cost=7500;           %dollars
    motorinverter_power=175000;        %W; continuous
    gearbox_mass=45;                   %high torque gearbox is needed based on the map
    gearbox_cost=7500;                 %high torque gearbox is needed based on the map
    motor_cylinder_dia = 0.390;
    inverter_box_h = 0.422;
    gearbox_h = 0.3;
    load MotorA_Data.mat               %to load motor A map
elseif motorinverter_select=='B'
    motor_mass=41;                     %kg
    inverter_mass=16;                  %kg
    motorinverter_cost=5200;           %dollars
    motorinverter_power=70000;         %W; continuous
    gearbox_mass=28;                   %low torque gearbox is needed based on the map
    gearbox_cost=4000;                 %low torque gearbox is needed based on the map
    motor_cylinder_dia = 0.276;
    inverter_box_h = 0.393;
    gearbox_h = 0.25;
    load MotorB_Data.mat               %to load motor B map
elseif motorinverter_select=='C'
    motor_mass=55;                     %kg
    inverter_mass=12;                  %kg
    motorinverter_cost=5500;           %dollars
    motorinverter_power=100000;        %W; continuous
    gearbox_mass=45;                   %high torque gearbox is needed based on the map
    gearbox_cost=7500;                 %high torque gearbox is needed based on the map
    motor_cylinder_dia = 0.28;
    inverter_box_h = 0.2;
    gearbox_h = 0.3;
    load MotorC_Data.mat               %to load motor C map
end
```

Assumptions

```
%% Assumptions

E=2.15*(10^11);                     %Young's Modulus Pa
G=78000;                            %Shear Modulus N/mm2

Cr=0.009;                           %Rolling coefficient Cr=0.009
mu=0.9;                             %Coefficient of friction (dry asphalt) = 0.9
BrkGain = 1e4;                      %Max Brake effort gain [N]
eff_dl =0.9;                        %Driveline efficiency [unitless]

useful_battery_capacity=0.95;       %the useful capacity of the battery is 90% of the capacity

Jz = 2380;                          % Moment of inertia of the vehicle [kgm^2]
Caf = 1290;                         % Tire cornering stiffness front [N/deg]
Car = 3000;                         % Tire cornering stiffness rear [N/deg]
Cmzf = 30;                          % Tire aligning stiffness front [Nm/deg]
Cmzr = 20;                          % Tire aligning stiffness rear [Nm/deg]

MR=0.9;                               % Motion Ratio
```

2D Model inputs

```
%% 2D Model inputs
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LOADS IN FRAME RAIL SECTION NEEDS TO BE UPDATED MANUALLY BASED ON THE
% DESIGN %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%the below values to be entered from the 2D model and are required for
%calculations, they are just assumed now to run the model
L108=3944/1000;                         %The total length of the vehicle
H101=1420/1000;                         %The total vehicle height
battery_thickness=0.05;                 %battery thickness
L104 = 0.622;                             %front overhang
H156 = 0.140;                         % ground clearance
H5_1 = 0.500;                         % front passenger H-Point
H5_2 = 0.550;                           % rear passenger H-Point

% X loads locations // needed for the frame rail calculations later
cargofront_X_dis = 0.4;                 % Cargo         (front)
motorf_X_dis = 0.4;                     % Motor         (front)
inverterf_X_dis = 0.4;                  % Inverter      (front)
gearboxf_X_dis = 0.4;                   % Gearbox       (front)
differentialf_X_dis = 0.4;              % Differential  (front)
frontaxle_X_dis = 0.7;                  % Axle          (front)
batteryfront_X_dis = 0.7;               % Battery       (front)
infotainment_X_dis = 0.9;               % Infortainment
frontpass_X_dis = 1.9;                  % Passenger     (front)
sunroof_X_dis = 2.0;                    % Sunroof
body_X_dis = L108/2;                    % Body
rearpass_X_dis = 2.8;                   % Passenger     (rear)
motorr_X_dis = 3.1;                     % Motor         (rear)
inverterr_X_dis = 3.1;                  % Inverter      (rear)
gearboxr_X_dis = 3.1;                   % Gearbox       (rear)
differentialr_X_dis = 3.1;              % Differential  (rear)
```

```
rearaxle_X_dis = 3.4;                   % Axle           (rear)
batteryrear_X_dis = 3.4;                % Battery        (rear)
cargorear_X_dis = 4;                    % Cargo          (rear)
framerail_X_dis_start = 0;              % Frame Rail
framerail_X_dis_end = L108;             % Frame Rail

% Y loads locations // needed for the frame rail calculations later
cargofront_Y_dis = H156 + 0.436 + 0.225/2 ;        % Cargo          (front)
motorf_Y_dis = H156 + motor_cylinder_dia/2 ;       % Motor          (front)
inverterf_Y_dis = H156 + inverter_box_h/2 ;        % Inverter       (front)
gearboxf_Y_dis = H156 + gearbox_h/2 ;              % Gearbox        (front)
differentialf_Y_dis = H156 + r ;                   % Differential   (front)
frontaxle_Y_dis = H156 + r ;                       % Axle           (front)
batteryfront_Y_dis = H156 + battery_thickness/2 ;  % Battery        (front)
infotainment_Y_dis = H5_1;                         % Infortainment
frontpass_Y_dis = H5_1;                            % Passenger      (front)
sunroof_Y_dis = H101;                              % Sunroof
body_Y_dis = H101/3;                               % Body
rearpass_Y_dis = H5_2;                             % Passenger      (rear)
motorr_Y_dis =  H156 + motor_cylinder_dia/2 ;      % Motor          (rear)
inverterr_Y_dis = H156 + inverter_box_h/2 ;        % Inverter       (rear)
gearboxr_Y_dis = H156 + gearbox_h/2 ;              % Gearbox        (rear)
differentialr_Y_dis = H156 + r;                    % Differential   (rear)
rearaxle_Y_dis = H156 + r ;                        % Axle           (rear)
batteryrear_Y_dis = H156 + battery_thickness/2 ;   % Battery        (rear)
cargorear_Y_dis = H156 + 0.436 + 0.225/2;          % Cargo          (rear)
framerail_Y_dis = H156 + h/2;                      % Frame Rail
```

Vehicle Mass Calculations

The vehicle mass calculations are summation basic calculations of the all of the masses included in the parameters. As mentioned earlier, parameters selection is flexible based on the decision made from the 2D layout, or design variables.

```
%% Vehicle Mass Calculations
% SI units have been used all through out the simulation unless otherwise stated

frame_rail_mass=3*2*frame_rail_area*L108*7800;          %kg (based on the frame rail design and 2D inputs
body_mass=L108*H101*50;                                 %kg
mass=(4*passenger_mass)+cargo_mass+info_mass+sunroof_mass+unsprung_mass+ ...
    total_gearbox_mass+total_differential_mass+axle_mass+body_mass+ ...
    frame_rail_mass+battery_mass+total_motorinverter_mass;
```

Frame Rail Calculations

To make this section flexible with any design variables selection, it was constructed assuming everything is in the model (AWD4, batteries in the front and rear, cargo in the front and rear, .. etc.). Then, these options are changing as a function of the design variables (like making the front motor load =0 if RWD option was selected).

The model assumptions:

1. Assume the maximum loads on the frame to equal twice the static gravitational loads.
2. Assume that the loads are supported by a frame consisting of two rectangular steel tubes, and each load force is divided equally over the two rails and acts on them at the location of the corresponding center of gravity. The maximum stress occurs where the bending moment is largest.
3. As the unsprang mass is part of the suspension system, it won't be counted in the loads acting on the frame rail.

4. Based on 3 and 4 the loads per axles will remain the same (multiplied by 2 then divided by 2).

After having the max bending moment from the solver, the next step will be determining the max stress:

$$\boldsymbol{\sigma} = \frac{\mathrm{M_{max}y}}{\mathrm{I}}$$

where:
sigma: is the stress
M: the bending moment
I : the second moment of inertia
y : distance from the neutral axis, in this case will be assigned h/2 to get the max stress.

Then based on the stress the frame rail parameters could be changed to have optimum frame rail mass matching these requirements. Based on the above, the matlab code was constructed to reflect these design parameters, as below:

```matlab
%% Frame Rail calculations
% as per the assumptions in the report the below analysis was performed

% calculating the different loads (all of them in N)
load_cargof= 0;
    if ((configuration =='RWD1') | (configuration =='RWD2'))
     load_motorf=0;
     load_inverterf=0;
     load_gearboxf=0;
     load_differentialf=0;
     elseif ((configuration =='FWD2') | (configuration =='AWD4'))
     load_motorf= -2*motor_mass*g;
     load_inverterf= -2*inverter_mass*g;
     load_gearboxf= -1*gearbox_mass*g;
     load_differentialf= -1*differential_mass*g;
     elseif ((configuration =='FWD1') | (configuration =='AWD2'))
     load_motorf= -1*motor_mass*g;
     load_inverterf= -1*inverter_mass*g;
     load_gearboxf= -1*gearbox_mass*g;
     load_differentialf= -1*differential_mass*g;
     end
load_axlef= -axle_mass*g/2;
load_batteryf= -battery_mass*g/2;
load_infotainment= -info_mass*g;
load_passf= -2*passenger_mass*g;
load_sunroof= -sunroof_mass*g;
load_body=-body_mass*g;
load_passr= -2*passenger_mass*g;
     if ((configuration =='RWD1') | (configuration =='AWD2'))
     load_motorr= -1*motor_mass*g;
     load_inverterr= -1*inverter_mass*g;
     load_gearboxr= -1*gearbox_mass*g;
     load_differentialr= -1*differential_mass*g;
     elseif ((configuration =='FWD2') | (configuration =='FWD1'))
     load_motorr=0;
     load_inverterr=0;
     load_gearboxr=0;
     load_differentialr=0;
     elseif ((configuration =='RWD2') | (configuration =='AWD4'))
     load_motorr= -2*motor_mass*g;
     load_inverterr= -2*inverter_mass*g;
     load_gearboxr= -1*gearbox_mass*g;
     load_differentialr= -1*differential_mass*g;
     end
load_axler= -axle_mass*g/2;
load_batteryr= -battery_mass*g/2;
load_cargor= -cargo_mass*g;
load_framerail= -frame_rail_mass*g/L108;


% The SFBM function
Name1 = 'Frame Rail';
LengthSupport1 = [L108,L104,(L101+L104)];        % Length and Supports
% Concetrated Loads

F1 = {'CF',load_cargof,cargofront_X_dis};
F2 = {'CF',load_motorf,motorf_X_dis};
F3 = {'CF',load_inverterf,inverterf_X_dis};
F4 = {'CF',load_gearboxf,gearboxf_X_dis};
F5 = {'CF',load_differentialf,differentialf_X_dis};
F6 = {'CF',load_axlef,frontaxle_X_dis};
F7 = {'CF',load_batteryf,batteryfront_X_dis};
F8 = {'CF',load_infotainment,infotainment_X_dis};
F9 = {'CF',load_passf,frontpass_X_dis};
F10 = {'CF',load_sunroof,sunroof_X_dis};
F11 = {'CF',load_body,body_X_dis};
F12 = {'CF',load_passr,rearpass_X_dis};
F13 = {'CF',load_motorr,motorr_X_dis};
```

```
F14 = {'CF',load_inverterr,inverterr_X_dis};
F15 = {'CF',load_gearboxr,gearboxr_X_dis};
F16 = {'CF',load_differentialr,differentialr_X_dis};
F17 = {'CF',load_axler,rearaxle_X_dis};
F18 = {'CF',load_batteryr,batteryrear_X_dis};
F19 = {'CF',load_cargor,cargorear_X_dis};

% Distributed Loads
D1 = {'DF',load_framerail,[framerail_X_dis_start,framerail_X_dis_end]};

% Call the function - output in the max bending mometn - N.m
bending_moment=SFBM(Name1,LengthSupport1,F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,F11,F12,F13,F14,F15,F16,F17,F18,F19,D1);
bending_moment_max=max(abs(bending_moment));                    %max bending moment

% Calculating the max stress on the beam (the hieghest moment at the most
% far line of the section (the biggest stresses)
second_moment_area=((w*(h^3))/12)-(((w-2*t)*((h-2*t)^3))/12);
stress=(bending_moment_max*(h/2))/(second_moment_area*1000000)          %MPa
```

Another parameter should be considered while changing the dimensions which is the max deflection of the beam.

The correct calculation method for max deflection for a beam with many different loads like our case will be hectic. So, to avoid this long time of developing this model. We will calculate the max deflection based on the below assumption:

- The frame rail will be represented as two supporters (the two axles), with load at the middle equals the total weight of the vehicle.

We know that this is a very rough estimate. However, the resulting deflection from this calculation will be much higher than the real deflection. So, if the frame rail can meet the requirements with this rough estimate, it will definitely meet the requirements in the real conditions.

BEAM BENDING

L = overall length W = point load, M = moment w = load per unit length	End Slope	Max Deflection	Max bending moment
M	$\frac{ML}{EI}$	$\frac{ML^2}{2EI}$	M
W	$\frac{WL^2}{2EI}$	$\frac{WL^3}{3EI}$	WL
w	$\frac{wL^3}{6EI}$	$\frac{wL^4}{8EI}$	$\frac{wL^2}{2}$
M M	$\frac{ML}{2EI}$	$\frac{ML^2}{8EI}$	M
W, ½ L ½ L	$\frac{WL^2}{16EI}$	$\frac{WL^3}{48EI}$	$\frac{WL}{4}$
w	$\frac{wL^3}{24EI}$	$\frac{5wL^4}{384EI}$	$\frac{wL^2}{8}$

Max deflection = (mass*g*(L101^3))/48*E*I

```
% Calculating the deflection
% rough assumption of having the vehicle weight affecting at the center
% distance betweeen the two wheels (the most sever condition)
% two supporters and one concentrated load at the middle equals (m.g)

deflection_max=(mass-unsprung_mass)*g*(L101^3)/(48*E*second_moment_area);
```

Vehicle Cost Calculations

Vehicle cost is the sum of the individual cost items, considering the selected configuration, and the other design variables. Most of the masses are calculated already in the parameters section.

```
%% Vehicle Costing Calculations
body_cost=1000*W103*L108*H101;
frame_rail_cost=frame_rail_mass*2;
battery_cost=145*battery_capacity/1000;

cost=misc_electronics_cost+wheel_assembly_cost+(gearbox_cost*num_gearbox) ...
    +differntial_cost+axles_cost+(num_motor*motorinverter_cost)+body_cost ...
    +frame_rail_cost+battery_cost
```

Vehicle Performance Model

This model is the base of the calculation of all performance related requirements and characteristics (0-100 km/h, max speed, 12% slope at 30 km/h, range ... and so on).

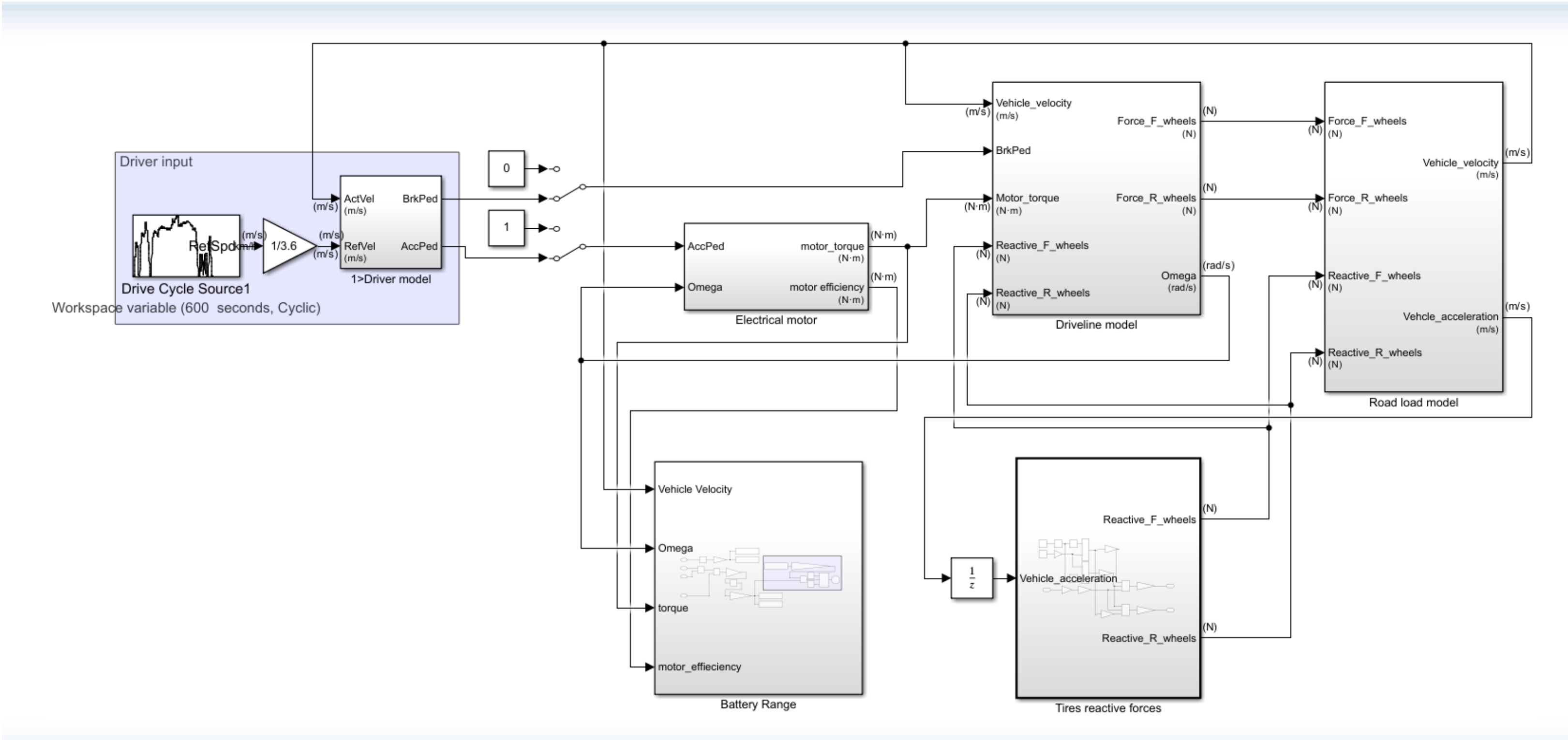
The model was created to be flexible and automatically consider all changes in design variables, and inputs. The model considers the same main MATLAB code, to be able to use it in the whole optimization at the end.

The performance model has three versions which are identical in everything except for the driving mode:

1. **Vehicle_Performance_Model_range**: in this model the driving model is following the US06_Drive_Cycle using a PI controller. This model is being used for the range calculations.
2. **Vehicle_Performance_Model_max**: in this model the driving mode is maximum performance (Acceleration Pedal signal of 1, and Brake Pedal Signal of 0). This model is being used for the 0-100 km/h, max speed, 12% slope at 30 km/h, ... and the other related characteristics to the maximum performance)
3. **Vehicle_Performance_Model_slope**: in this model the driving mode is constant speed of 30 km/h applied, and the PI controller used to copy this driving condition.

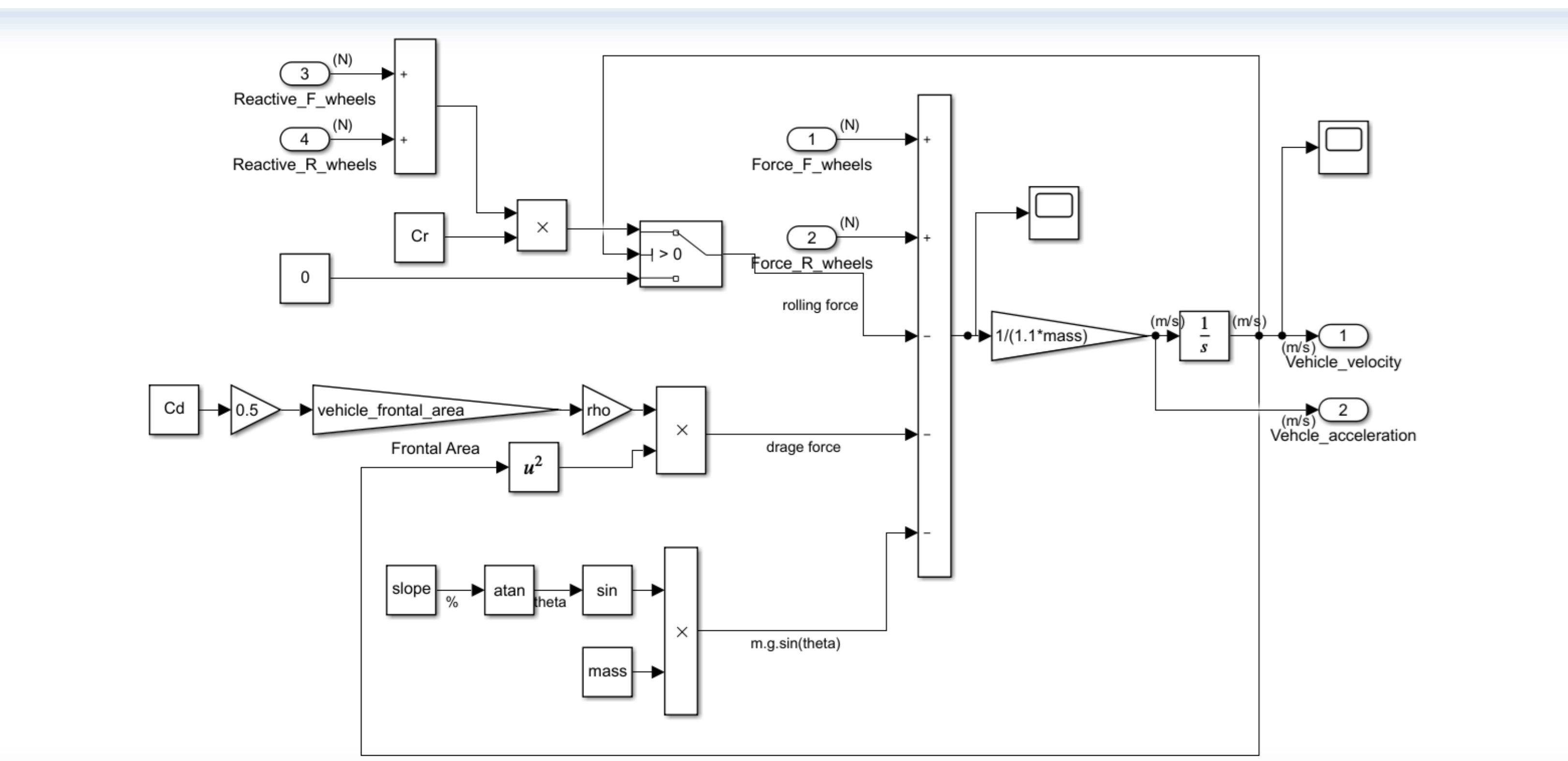
The main reason behind considering different models is making it easier to call each model separately in every code section and maintain all the outputs for simulations every iteration.

The main layout of the model is as below:



The two signals coming from the left-hand side are the Brake Pedal signal and Acceleration Pedal signal which are different based on the version. The model consists of other subsystems, each one of them will be described below with its corresponding MATLAB code.

Road Load Submodel



The modeling is based on the equations in Task 1. Switch was considered for rolling resistance while stationary.

The model has four inputs:

1. 1, and 2 are the reactive forces on the front and rear axles.
2. 3, and 4 are the traction forces provided to the wheels in the front and rear axles also.

The model outputs:

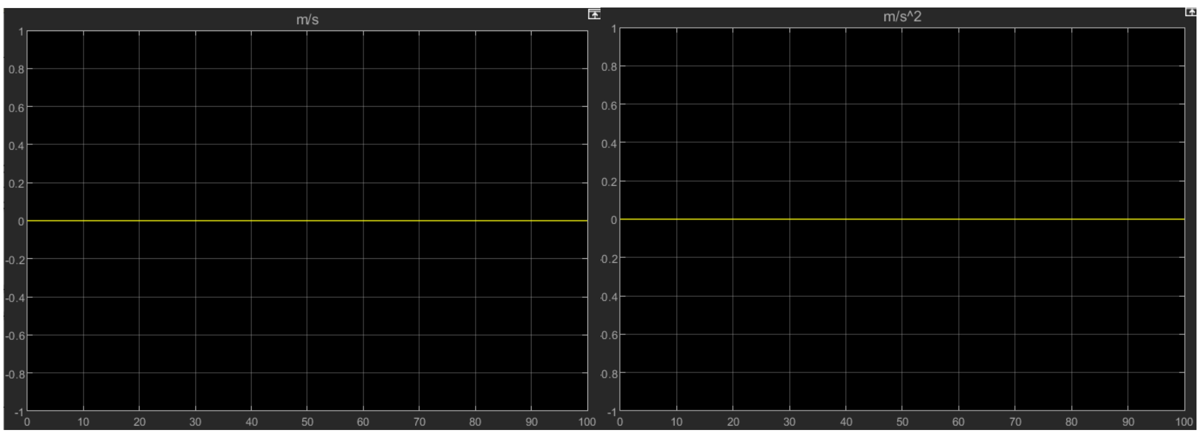
1. Vehicle Velocity.
2. Vehicle Acceleration.

The model shall:

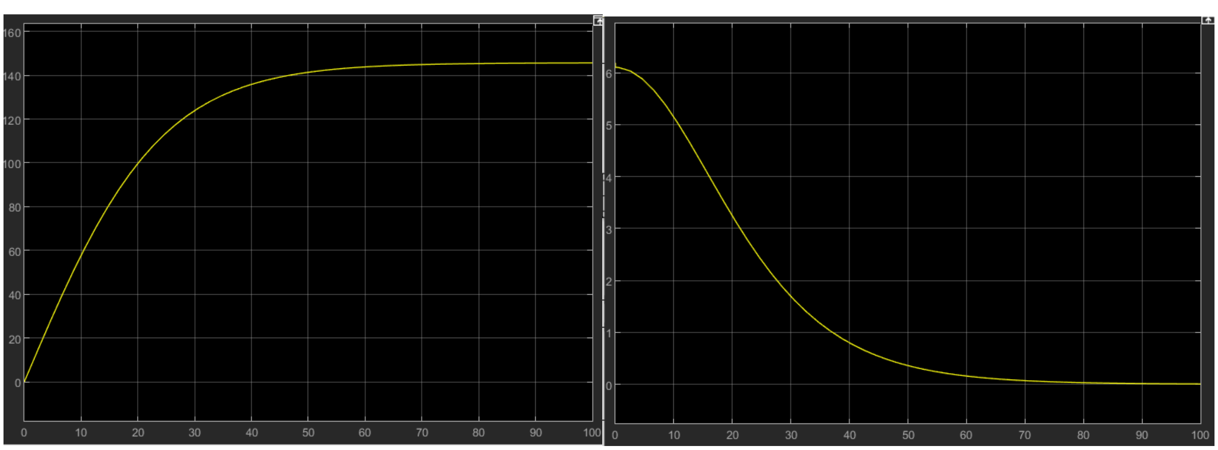
1. Calculate the resultant forces, acceleration of the vehicle in the longitudinal direction and the velocity based on the equations of motion specified before.

Testing the model:

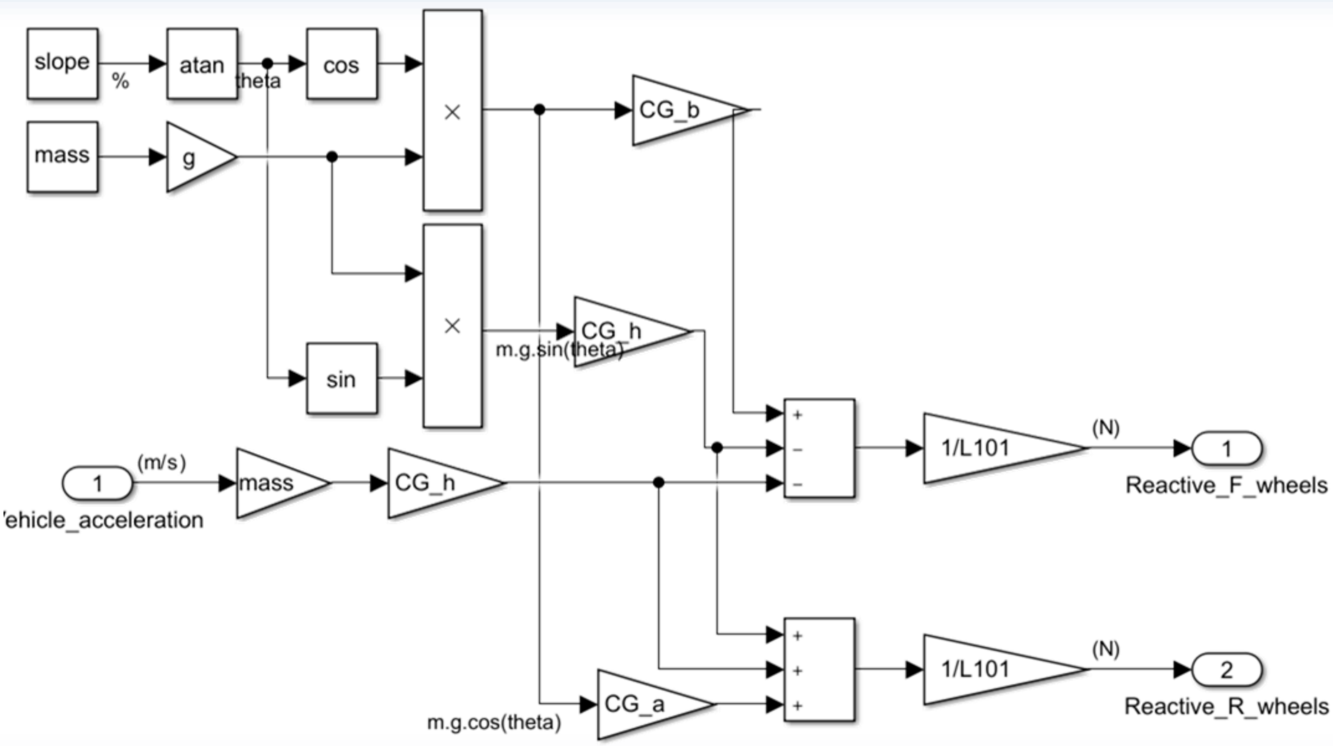
- With zero forces the system should respond with 0 m/s velocity, and 0 accelerations. Which is shown below:



- When applying constant forces on the wheels, the velocity should increase till reaching the maximum speed at which the acceleration will decrease to zero due to the increase of the drag force. Which is shown in the graphs below (velocity scope on the left (n/s) , and the acceleration on the right (m/s^2)).



Tires Reactive Forces Submodel



The modeling is based on the equations in Task 1.
The model has one input:

1. Vehicle acceleration from the previous model.

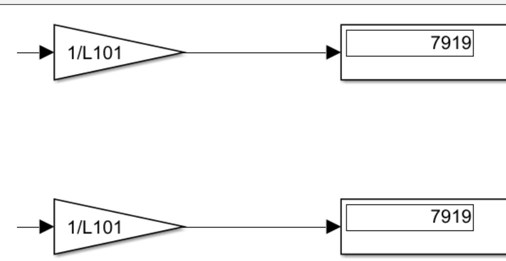
The model outputs:

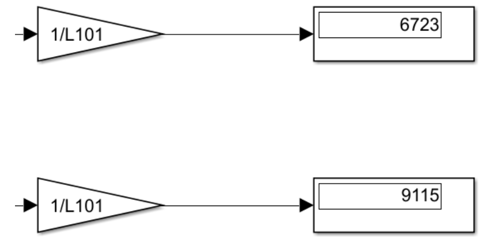
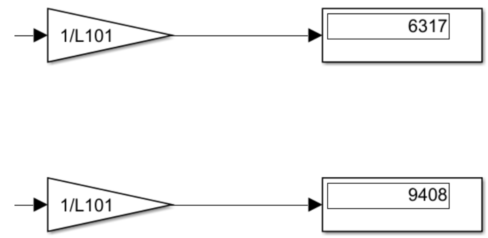
1. Reactive force on the front wheels.
2. Reactive forces on the rear wheels.

The model shall:

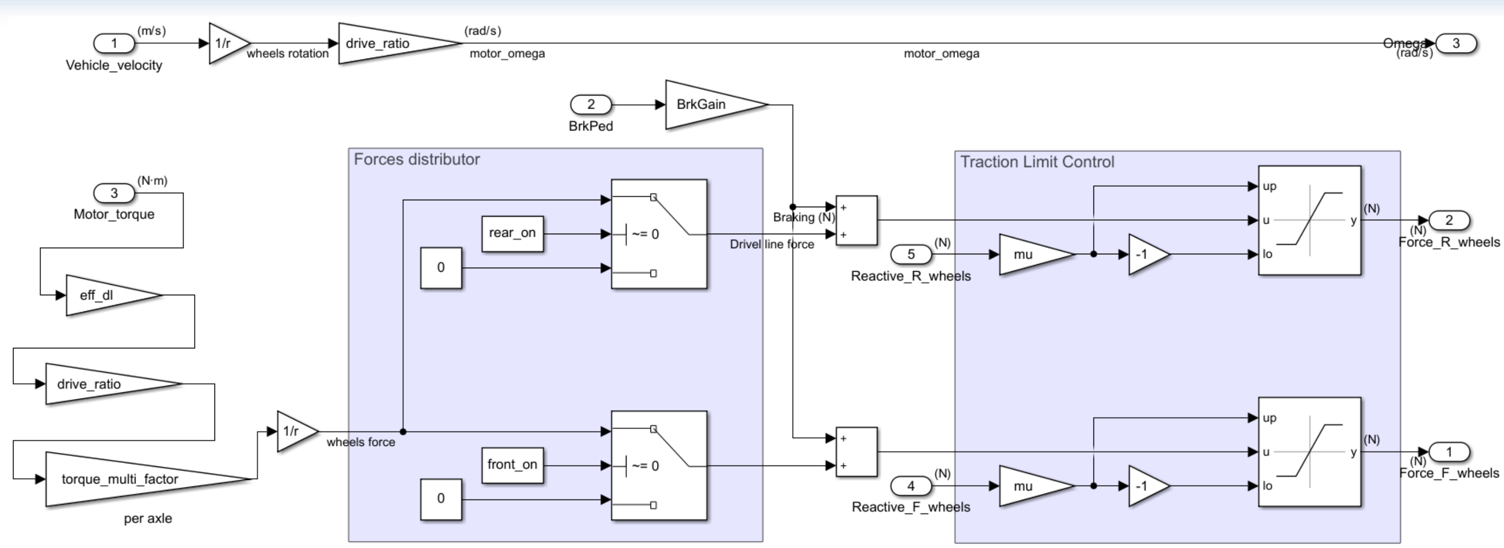
1. Calculate the reactive forces based on the weight distribution changes due to acceleration, and slope.

Testing the model:

Condition	**Result** (Front axle → top | Rear axle → bottom)
While no acceleration on a flat surface the weight should be distributed equally.	

While accelerating forward, the rear axle should have higher reactive force.	
Adding the slope to the previous condition should decrease the reactive force more on the rear wheels.	

Driveline Submodel



The model has five inputs:

1. Vehicle velocity
2. Electrical Motor Torque
3. Brake Pedal Input
4. Reactive normal forces on the front wheels
5. Reactive normal forces on the rear wheels.

The model outputs:

1. Front wheels traction forces.
2. Rear wheels traction forces.
3. Electrical motor rotational speed.

The model shall:

1. Multiply the torque based on the number of motors per axle.

2. Direct the traction forces to the proper axles based on the configuration selected.
3. Not to allow exceeding the traction limits for the output forces.
4. Consider the drive ratios for the speed and force calculations.
5. Consider the braking forces during force distribution to the wheels.

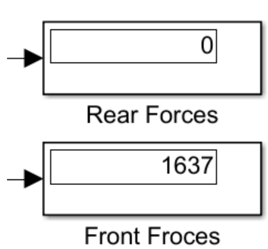
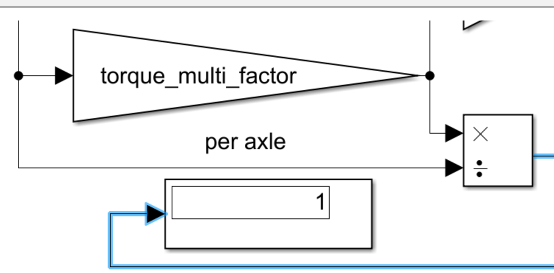
The model depends on the following matlab script to perform requirements 1 and 2.

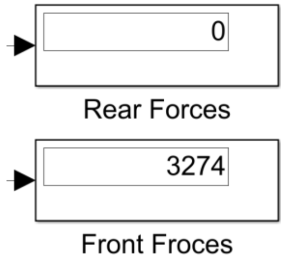
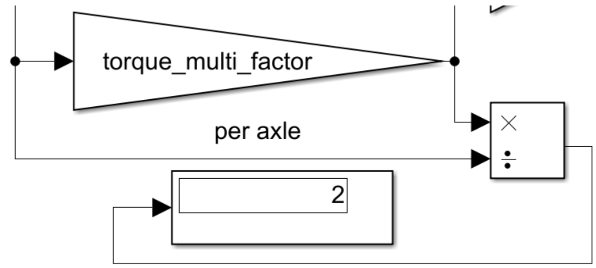
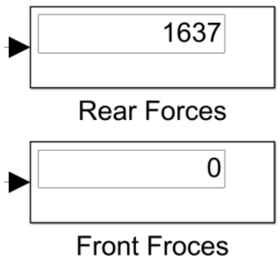
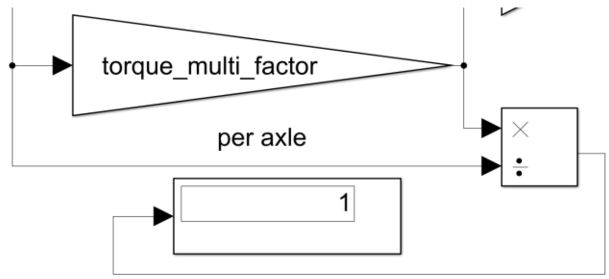
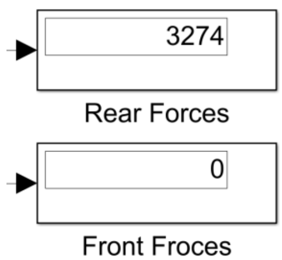
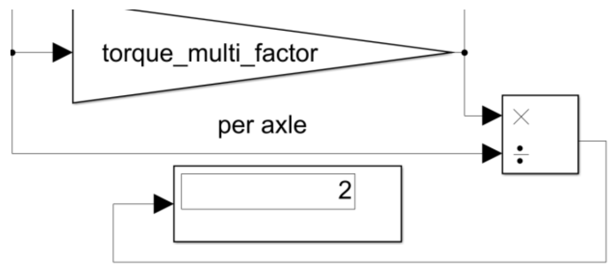
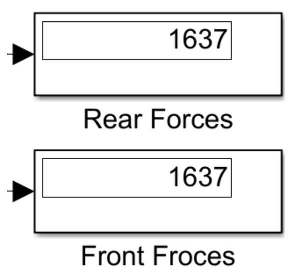
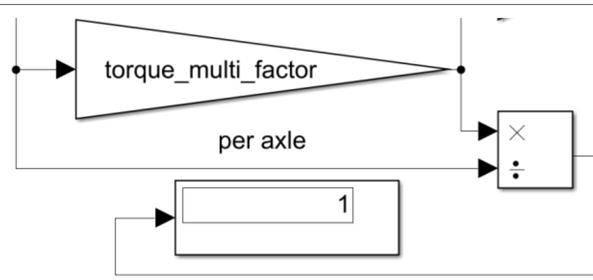
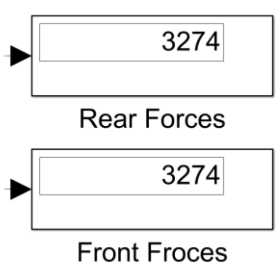
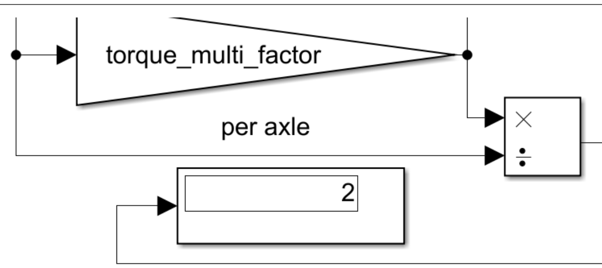
```
%% Vehicle Perfromance Model Setting and Parameters
%road load model
vehicle_frontal_area=H101*W103;

%Forces Distributer Model
front_on=0;      %0 means no forces to the front, 1 means forces are going to front
rear_on=0;       %0 means no forces to the rear, 1 means forces are going to rear
if ((configuration =='FWD1') | (configuration =='FWD2'))
    front_on=1;      %0 means no forces to the front, 1 means forces are going to front
    rear_on=0;       %0 means no forces to the rear, 1 means forces are going to rear
elseif ((configuration =='RWD2') | (configuration =='RWD1'))
    front_on=0;      %0 means no forces to the front, 1 means forces are going to front
    rear_on=1;       %0 means no forces to the rear, 1 means forces are going to rear
elseif ((configuration =='AWD4') | (configuration =='AWD2'))
    front_on=1;      %0 means no forces to the front, 1 means forces are going to front
    rear_on=1;       %0 means no forces to the rear, 1 means forces are going to rear
end

%Torque multiplication based on the number of engines per axle
torque_multi_factor=0;
if (configuration =='AWD2')
    torque_multi_factor=1;
elseif (configuration =='FWD1')
    torque_multi_factor=1;
elseif (configuration =='RWD1')
    torque_multi_factor=1;
elseif (configuration =='RWD2')
    torque_multi_factor=2;
elseif (configuration =='AWD4')
    torque_multi_factor=2;
elseif (configuration =='FWD2')
    torque_multi_factor=2;
end
```

Testing the model:

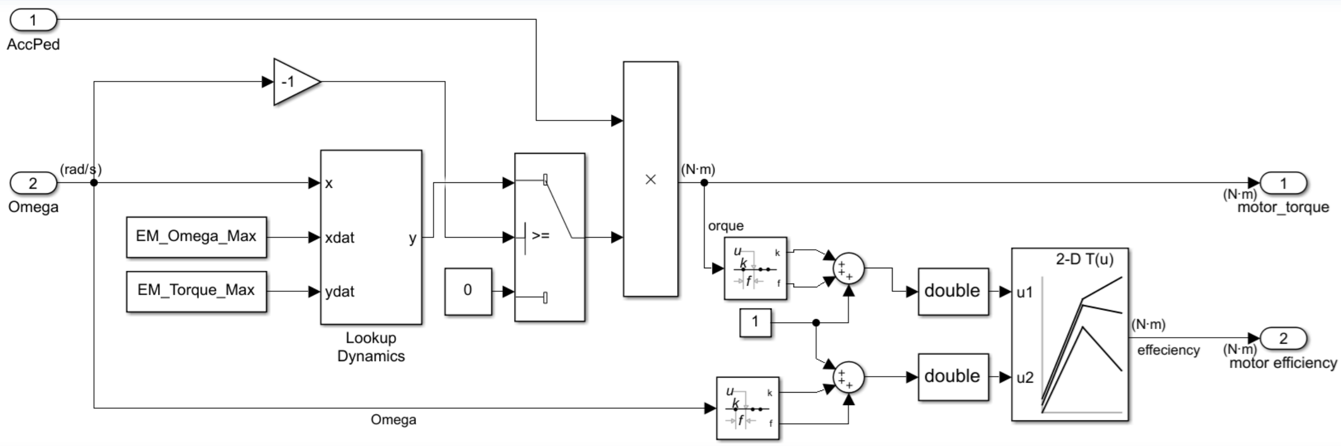
Most of the models are just multiplications, except for the torque multiplication and the load distribution, which will be tested.

- Adjusting the configuration variable in the main code “Design Variable” to different configurations and observing the output.
- Observing the torque multiplication ratio per axle for each configuration.

Configuration	Wheels forces	Torque Multiplication Ratio
FWD1		

FWD2		
RWD1		
RWD2		
AWD2		
AWD4		

Electrical Motor Submodel




```
%Electrical Motor Model
%Breakpoint values
omega_BP=max(size(EM_Omega_Map));
omega_max_BP=max(size(EM_Omega_Max));
torque_BP=max(size(EM_Torque_Map));
torque__max_BP=max(size(EM_Torque_Max));
%max Torqua and omega
omega_max=max(EM_Omega_Max);
torque__max=max(EM_Torque_Max);
```

The first part of the model (on the left) uses the current motor speed to determine the max corresponding torque, then multiply this max torque by the ratio of the acceleration pedal. This will give the required torque. Then, for the right part, the indexing of the torque and omega values is needed before the 2D map lookup, which is added before the 2D lookup block. The plus 1 is added as a correction to the indexing.

The model has two inputs:

1. Acceleration Pedal Input: from 0 to 1, where 1 is the max acceleration required.
2. Omega: which is the angular velocity of the engine.

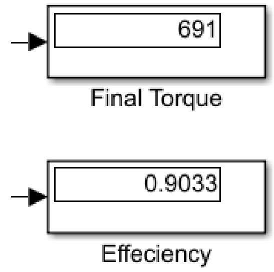
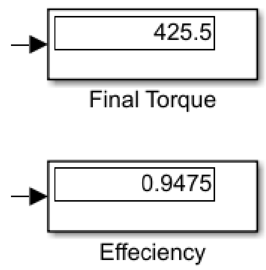
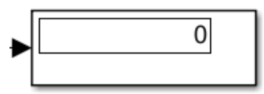
The model outputs:

1. Motor Torque: which will be transferred to the driveline, and eventually the wheels.

The model shall:

1. Calculate the right efficiency value with respect to each omega and torque.
2. Provide the right amount of torque based on the electrical motor maps.
3. The torque not to exceed the maximum torque specified by the maps.

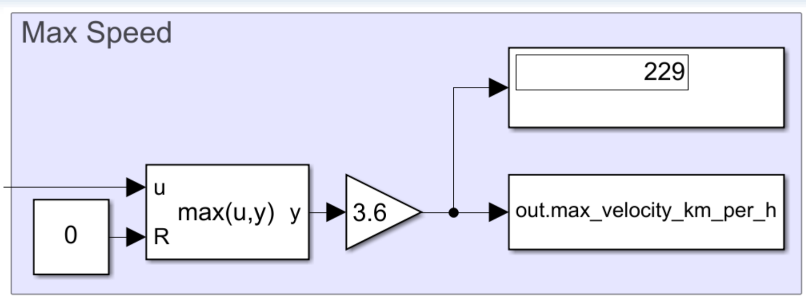
Testing the model:

Condition	Result
Based on the map of motor A, at motor speed of 200, and acceleration pedal input of 1, the torque shall be 765 N.m, and the efficiency between 0.9-0.91. As shown, the system responds as expected.	
To assure that the system will not exceed the max torque, a point at the borders of the map of 556 rad/s was checked, and at this point the torque shall not exceed 448.4 N.m and efficiency of ~0.94 with 1 acc input. The response meets the expectations as below.	
For the omega values outside the max omega, the motor should not provide any torque. The output of omega of 650 - which is higher than the max omega - is 0 N.m torque as shown.	

Max speed calculations

The submodel uses the vehicle speed input (the input on the left) to determine the max speed during the running range. The output is then sent to the workspace to be printed in the main code.

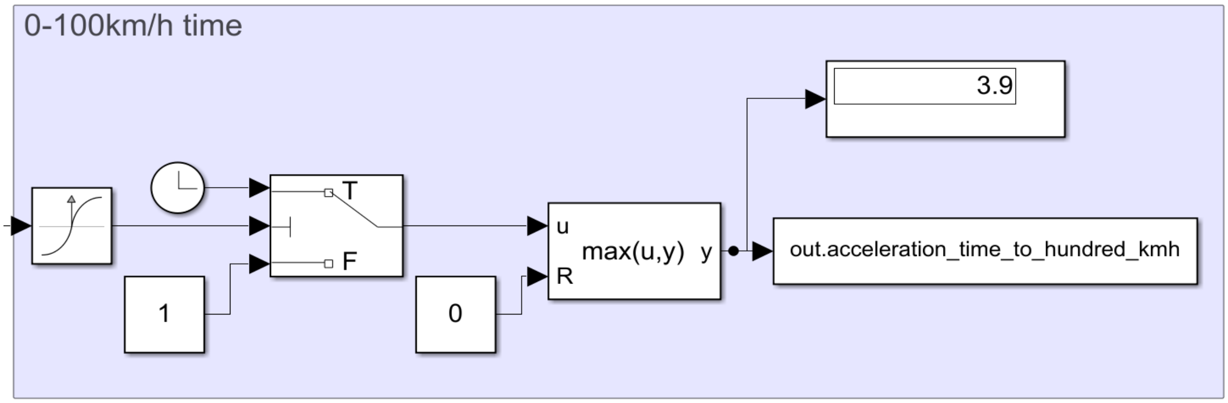
The testing for this model is comparing the result to the velocity scope over time, and they were identical.




```
%% Max speed calculations (1 of 3 don't change the sequence)

sim_max=sim('Vehicle_Performance_Model_max.slx');
max_speed=max(sim_max.max_velocity_km_per_h)                    %(m/s)
```

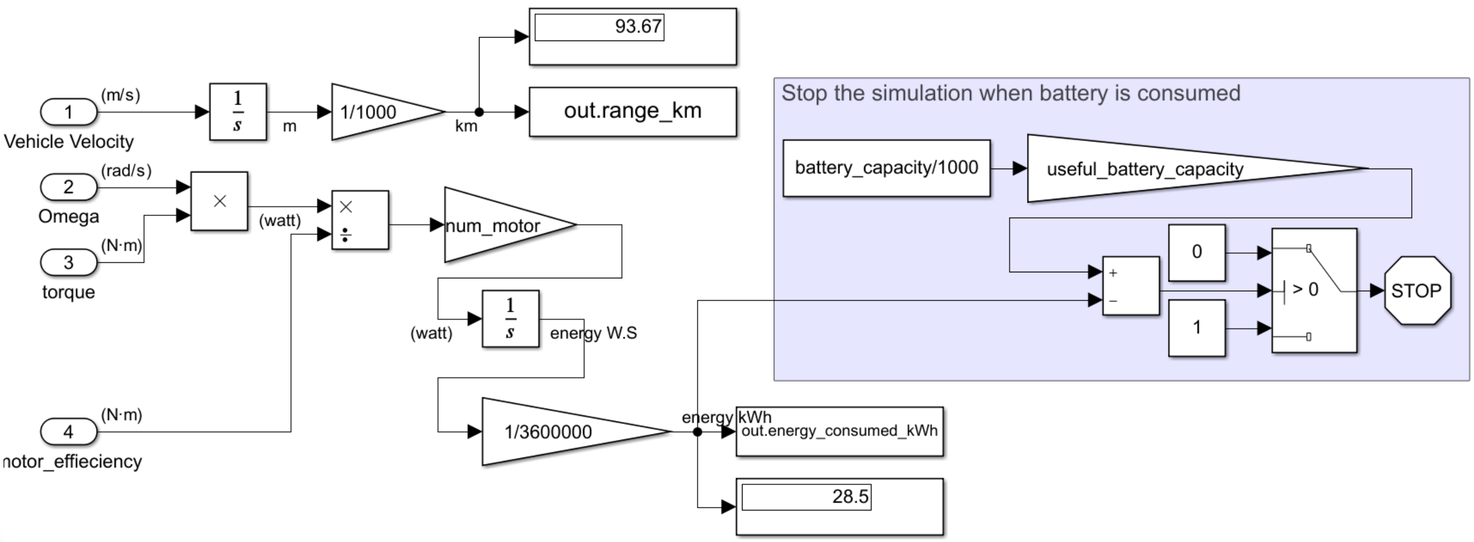
0-100km/h Time Calculations

The submodel uses the vehicle speed input to determine the time estimation from 0-100 km/h. The output then is being sent to the workspace to be printed with the main code.
The testing for this model is comparing the result to the velocity scope over time, and they were identical.




```
%% 0-100 km/h time calculations (2 of 3 don't change the sequence)

acceleration_time=max(sim_max.acceleration_time_to_hundred_kmh) %(s)
```

Battery Range Calculations




```
%% Range Calculations  calculations (3 of 3 don't change the sequence)

sim_range=sim('Vehicle_Performance_Model_range.slx');
range_battery=max(sim_range.range_km)                %(km)
```

The model uses the vehicle speed and the motor consumption to determine the distance (range in km) and the total energy consumption from the motor. When the energy consumption equals the useful battery capacity, the submodel stops the whole model, and prints the range in km to the workspace.

The model has four inputs:

1. Vehicle Velocity
2. EM regular speed
3. EM torque
4. EM efficiency

The model outputs:

1. Battery Range (km).
2. Energy consumption (kWh).

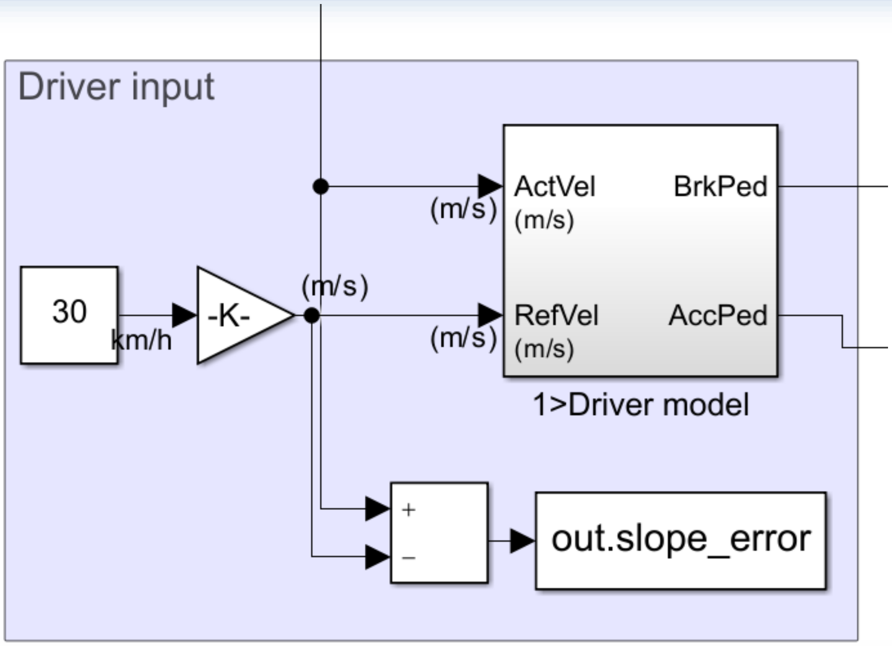
The model shall:

1. Calculate the energy consumption from the motors.
2. Calculate the distance travelled since the start of the model.
3. Stops the simulation when the battery capacity is consumed.

Assumptions:

- The useful battery capacity is 95% of the capacity.

Assessing the vehicle ability to ascend a 12% slope at 30 km/h



This is the third mode (the slope model described at the beginning of task 2) which uses the above block as the driving condition and produces the two signals on the left at the acceleration and braking signals.

The slope_error is the difference between the actual speed and the reference speed of 30 km/h. This variable is sent to the workspace and used in the below script.

```
%% Ascend a 12% slope speed (minimum at 30 km/h)

slope=0.12;                     %the road slope in %
sim_slope=sim('Vehicle_Performance_Model_slope.slx');

slope_difference_variation=sim_slope.slope_error.Data(end);

if abs(slope_difference_variation)<0.001
    slope_result='The vehicle is able to ascend a 12% slope at 30 km/h.'
else
    slope_result='The vehicle is NOT able to ascend a 12% slope at 30 km/h.'
end
```

The above script assesses the error and compares it to a threshold and feedback with the ability to ascend the slope with the required speed or not, based on the error value.

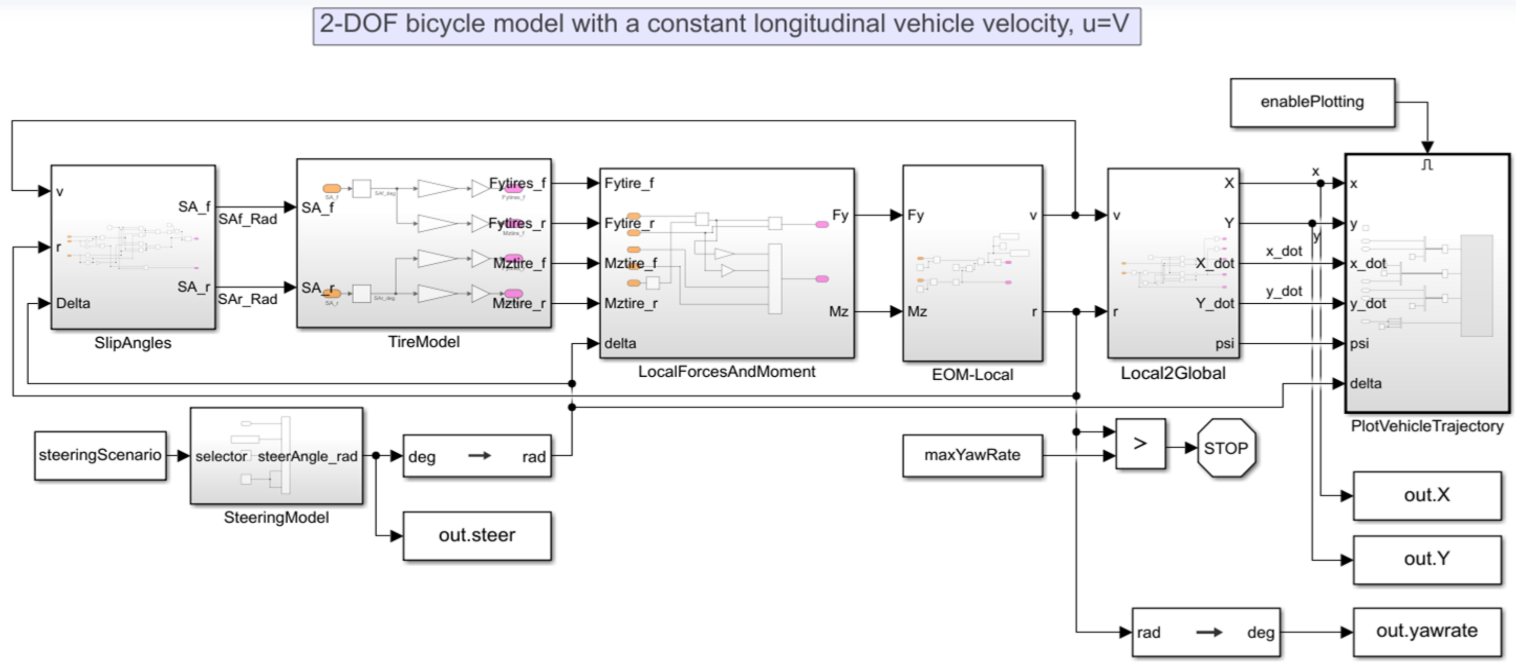
Profit Calculations

```
%% Profit calculations

profit_input=[cost; range_battery; acceleration_time; max_speed];
[profit,demand,price] = profitPredict(profit_input)
```

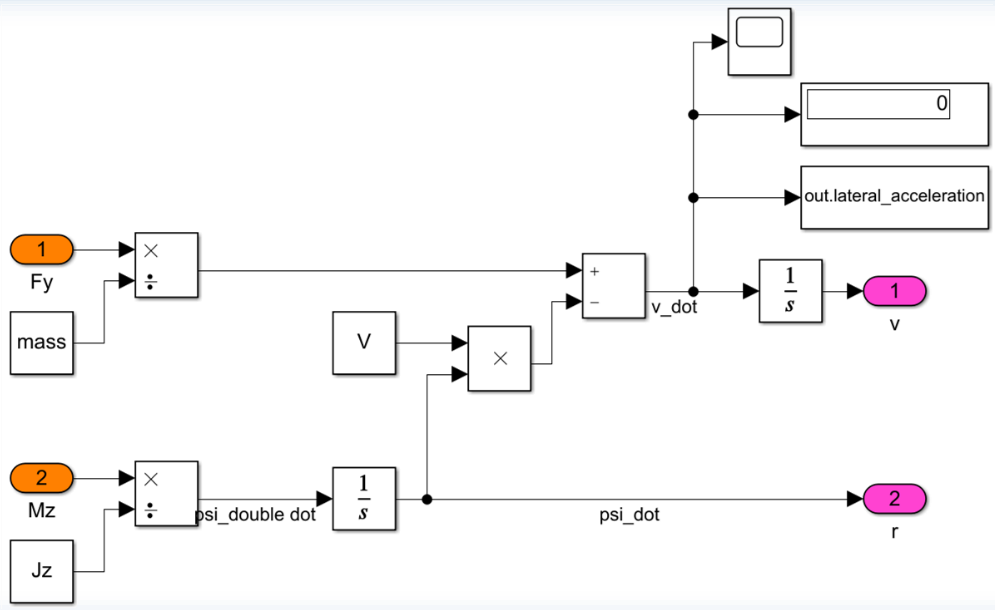
The profit is being calculated using the profitPredict function.

Assessing the vehicle ability to turn on a 100m diameter skidpad at up to 0.8g lateral acceleration

The model used for this one is the same model used for HW4 “2-DOF bicycle model with a constant longitudinal vehicle velocity, u=V” with some slight modifications. This model was tested before and graded with no issues. That is why we will continue using it directly.



The EOM block was changed to send the lateral acceleration to the workplace.

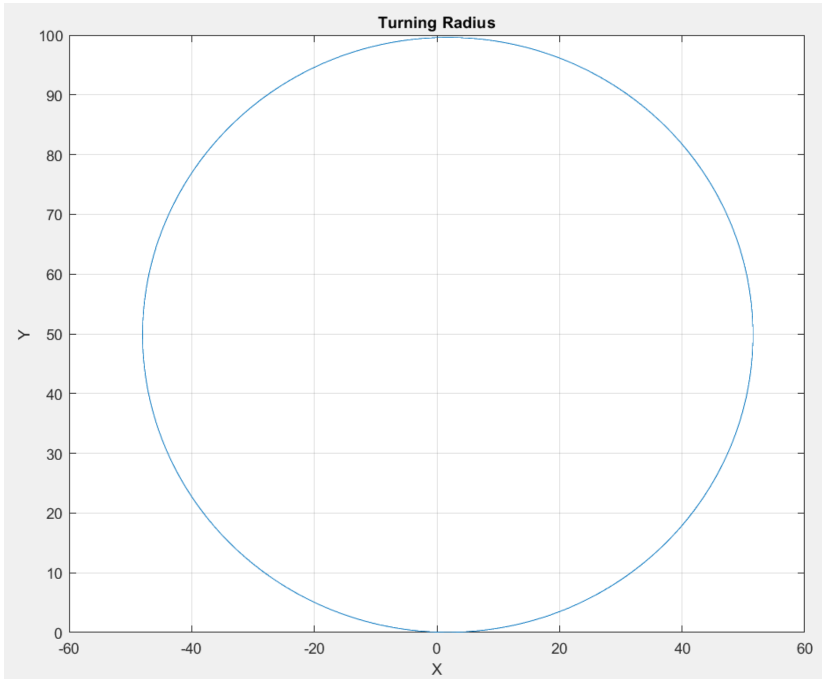


Then the following script was created. The main purpose of the script is calculating the lateral acceleration while the vehicle is moving at 100 m diameter. Then, it prints the output about the vehicle capability to take that turn. The vehicle is meant to be understeering, this is being controlled by CG location and the tire cornering stiffness. The assumptions are as follows:

- Jz = 2380; % Moment of inertia of the vehicle [kgm^2]
- Caf = 1450; % Tire cornering stiffness front [N/deg]
- Car = 3000; % Tire cornering stiffness rear [N/deg]
- Cmzf = 30; % Tire aligning stiffness front [Nm/deg]

- Cmzr = 20;   % Tire aligning stiffness rear [Nm/deg]
- The vehicle has a constant longitudinal speed of flat road (slope =0%), so load distribution on wheels is only a function of the distance of the CG to the front and rear

```
%% Assesing the stability at 100 m radius with up to 0.8 g lateral accleration

Kus=((CG_location*mass)/(2*Caf))-(((1-CG_location)*mass)/(2*Car))

steeringScenario = 1;
steerAngle = 6;   % Ackermann steering angle in [deg]
enablePlotting = false;        % turn on the plotting features (off is faster!)
maxYawRate = 20;               % [rad/s] -- stop the simulation if yaw rate is >

%Intial values
V = 30;                       % velocity [m/s] (1 kmph is 40/3.6 m/s)
turning_data = sim('vehicle_dynamics_turning.slx',200);
Turning_data=[turning_data.X.Data turning_data.Y.Data];
[Lc,R,K] = curvature(Turning_data); % external function
radius=mean(R(50:end-1,:)); % excluding the first 50 points before the steady speed

%Determining the speed for 100 m diamter (50 m radius)
while (radius<49.5) || (50.5<radius)
    if radius<50
        V=V+0.5;
    elseif radius>50
        V=V-0.5;
    end
    turning_data = sim('vehicle_dynamics_turning.slx',200);
    Turning_data=[turning_data.X.Data turning_data.Y.Data];
    [Lc,R,K] = curvature(Turning_data); % external function
    radius=mean(R(50:end-1,:)); % excluding the first 50 points before the steady speed
end

% calculating the simulated turning radius
turning_data = sim('vehicle_dynamics_turning.slx',1000);
Turning_data=[turning_data.X.Data turning_data.Y.Data];
[Lc,R,K] = curvature(Turning_data); % external function
radius=mean(R(50:end-1,:)) % excluding the first 50 points before the steady speed
lateral_acceleration_max=max(abs(turning_data.lateral_acceleration.data))

if lateral_acceleration_max>=(g*0.8)
    turnning_result='[Pass]'
elseif lateral_acceleration_max<(g*0.8)
    turning_result='[Adjust]'    %Increase the speed and steering angle
end

% figure(1)
% plot(turning_data.X.Data,turning_data.Y.Data);
% grid on
% title('Turning Radius')
% xlabel('X')
% ylabel('Y')
```



Ride Frequency (Spring constant, and Damping coefficient)

Based on the equations specified in Task1, the below script is calculating the spring coefficient.

```
%% Spring Constant Calculations

front_corners_sprung_mass=(mass-unsprung_mass)*(1-CG_location)/2;
rear_corners_sprung_mass=(mass-unsprung_mass)*(CG_location)/2;

spring_rate_front=4*(pi^2)*(frf^2)*front_corners_sprung_mass*(MR^2)      % N
spring_rate_rear=4*(pi^2)*(frr^2)*rear_corners_sprung_mass*(MR^2)        % N
%% Damping Coefficient

spring_damping_coeffiecient_front = 4*pi*damping_ratio*frf*front_corners_sprung_mass;    %N.s/m
spring_damping_coeffiecient_rear = 4*pi*damping_ratio*frr*rear_corners_sprung_mass;      %N.s/m
```

TASK

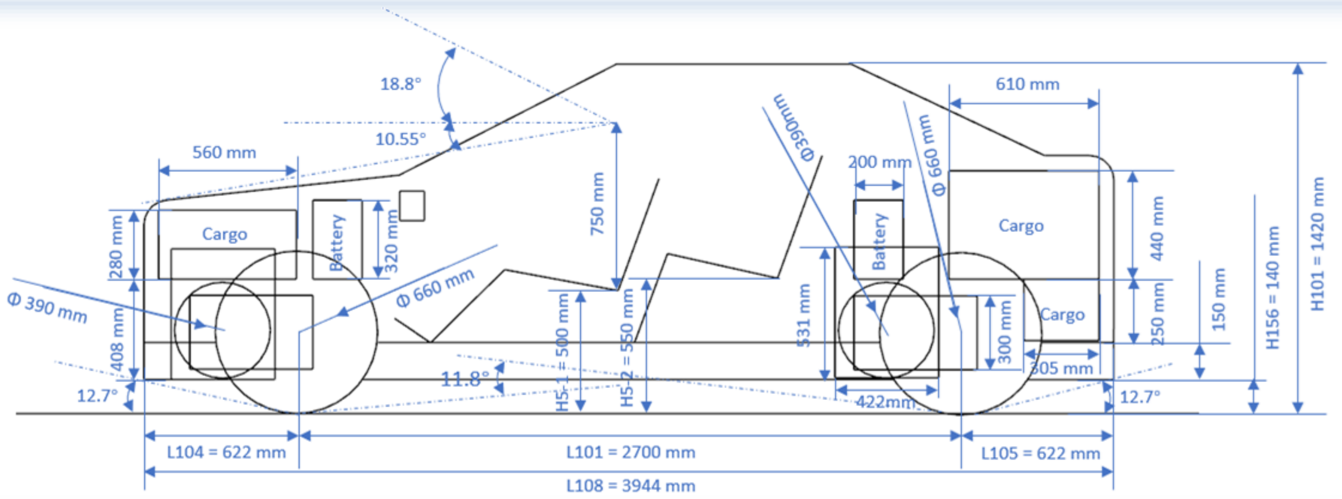
3

EXPLORE THE DESIGN SPACE TO FIND A FEASIBLE SOLUTION

In the way of determining the most feasible design. The approach is changing the design variables for each design configuration (driving axles and motor type), and observing the profit (and its inputs), with these changes. Then, comparing all these different configurations to find the best feasible design.

Frame Rail Design

To be able to move forward with the iterations, the first thing needs to be calculated is the dimensions of the frame rail. However, it is necessary to have a 2D layout design and known configuration to be able to calculate the stresses on the frame rail. The below layout was assumed in the preliminary calculations as this will be the worst-case scenario from loads standpoint to the frame rail. Then after selecting a specific design, the frame rail dimensions will be reoptimized for the new loads.



Assumptions:

1. AWD configuration with 4 motors (assumed as this will be the highest load).
2. Assume motor A was selected (the heaviest option).
3. Assume the maximum loads on the frame to equal twice the static gravitational loads.
4. Assume battery capacity to be 70 kWh - acceptable range for electrical cars -, the mass is (70,000/250).

Selecting the frame rail preliminary dimensions

The optimization sections in the code was add at the start and end of the frame rail calculations (to be commented out later) to go through several iteration to calculate the max stress and deflection for each w, t, and h (aspect_ratio*w).

```
%% Frame Rail optimization - intialization (comment out after finishing it)
frame_rail_optimization=zeros(0,6); %w,aspect ratio,t,area,stress, deflection
aspect_ratio=1;
i=0;
for w=0.04:0.005:0.15
    for aspect_ratio=1:0.5:5
        h=w*aspect_ratio;
        for t=0.001:0.001:0.015
            i=i+1;
            frame_rail_area=(w*h)-((w-(2*t))*(h-(2*t)));
```

Then after the frame rail calculations

```
%% Frame Rail optimization - end (comment out after finishing it)
close all
%w,aspect ratio,t,area,stress, deflection
frame_rail_optimization(i,1)=w;
frame_rail_optimization(i,2)=aspect_ratio;
frame_rail_optimization(i,3)=t;
frame_rail_optimization(i,4)=frame_rail_area;
frame_rail_optimization(i,5)=stress;
frame_rail_optimization(i,6)=deflection_max;
        end
    end
end
```

The output was 3105 iterations' results, after filtering the iteration with the max stress constraint (less than 215 MPa) → 2908, and after filtering the max deflection constraint (less than 0.015 m) → 2684.

Ranking the results with the least area, the results were as below:

Width	Aspect Ratio	Thickness	Area	Max Stress	Max Defelction	No	height
0.045	4.500	0.001	0.000491	181.488	0.015	241.000	0.203
0.050	4.000	0.001	0.000496	177.198	0.014	361.000	0.200
0.065	3.000	0.001	0.000516	163.165	0.014	736.000	0.195
0.075	2.500	0.001	0.000521	160.463	0.014	991.000	0.188
0.045	5.000	0.001	0.000536	153.316	0.011	256.000	0.225
0.060	3.500	0.001	0.000536	151.597	0.012	616.000	0.210
0.090	2.000	0.001	0.000536	153.314	0.014	1381.000	0.180
0.055	4.000	0.001	0.000546	146.638	0.011	496.000	0.220
0.050	4.500	0.001	0.000546	147.214	0.011	376.000	0.225
0.080	2.500	0.001	0.000556	141.175	0.012	1126.000	0.200

These results are the lowest ranked from the area and almost have the same area.

As the aspect ratio has an influence on the height of the center of gravity of the vehicle (the lowest the best as it will move the H-Point up, increase the Area and the CG). Accordingly, the highlighted option was selected as it has the lowest height and meets all other requirements:

w=0.090 m
h=0.18 m
t= 0.001 m

Please note that these parameters are preliminary and will be fine-tuned based on the decision made of the final design variables.

EXPLORING THE DESIGN SPACE

The analysis to explore the design space is being performed by iterating the design variables for each configuration. For each iteration, the following outputs are calculated to test the profit along with the requirements:

-	Iteration #1	Iteration #2	Iteration #3	Iteration #4 ...
configuration				
motorinverter_select				
battery_capacity				
drive_ratio				
profit				
cost				
acceleration_time				
max_speed				
range_battery				
slope_result				
stress				
deflection_max				
Kus				
lateral_acceleration_max				
diameter				
turning_result				
spring_rate_front				
spring_rate_rear				
spring_damping_coeffiecient_front				
spring_damping_coeffiecient_rear				

The iterations range and scripts are as below:

At the start of the main code.

```
%% Design Variables
solutions=zeros(0,0);
i=0;

configuration_s=1;              %(1)FWD1, (2)FWD2, (3)RWD1, (4)RWD2, (5)AWD2, or (6)AWD4
    if configuration_s==1
        configuration='FWD1';
    elseif configuration_s==2
        configuration='FWD2';
    elseif configuration_s==3
        configuration='RWD1';
    elseif configuration_s==4
        configuration='RWD2';
    elseif configuration_s==5
        configuration='AWD2';
    elseif configuration_s==6
        configuration='AWD4';
    end

for motorinverter_select_s = 1:1:3      %(1)A, (2)B, (3)C
    if motorinverter_select_s==1
        motorinverter_select='A';
    elseif motorinverter_select_s==2
        motorinverter_select='B';
    elseif motorinverter_select_s==3
        motorinverter_select='C';
    end

    for battery_capacity = 30000:10000:120000
        for drive_ratio = 1:1:10
            i=i+1;
w=0.090;                            %(m) frame rail width
h=0.180;                            %(m) frame rail height
t=0.001;                            %(m) frame rail thickness
```

and at the end of the main code:

```
%% Loops end (comment if not needed)
% each row represents one iteration
% configuration; motor type; battery capacity; drive ratio;
% profit; cost ; 0-100 km/h ; top speed; battery range;
% asecending 12% slope ; max stress ; max deflection;
% kus value; lateral accelration value; turinging diameter value; VD pass/fail; spring
% constant; spring damping coeffiecint; CG location

            solutions(i,1)=configuration_s;
            solutions(i,2)=motorinverter_select_s;
            solutions(i,3)=battery_capacity;
            solutions(i,4)=drive_ratio;
            solutions(i,5)=profit;
            solutions(i,6)=cost;
            solutions(i,7)=acceleration_time;
            solutions(i,8)=max_speed;
            solutions(i,9)=range_battery;
            solutions(i,10)=slope_result;
            solutions(i,11)=stress;
            solutions(i,12)=deflection_max;
            solutions(i,13)=Kus;
            solutions(i,14)=lateral_acceleration_max;
            solutions(i,15)=diameter;
            solutions(i,16)=turning_result;
            solutions(i,17)=spring_rate_front;
            solutions(i,18)=spring_rate_rear;
            solutions(i,19)=spring_damping_coeffiecient_front;
            solutions(i,20)=spring_damping_coeffiecient_rear;

          end
      end
 end
```

DESIGN SPACE EXPLORATIONS RESULTS

After running the main MATLAB code presented above, now we have many iterations for each design configuration, in this section we will be working on interpreting the iterations results.

In each set of iteration per configuration, there would be infeasible solution which are:

- Not meeting the requirements,
- or have unrealistic results for one of the KPIs (like having very low max speed).

These infeasible solutions will be filtered. An example of these infeasible solutions are the below highlighted solutions which have a max speed less than 100 km/h, which will not be considered in the further stages.

#	configuration	motorinverter_select	battery_capacity	drive_ratio	cost	acceleration_time	max_speed
1	1	1	30000	1	$ 35,491	23.1	240.6561769
2	1	1	30000	2	$ 35,491	10.8	283.6669215
3	1	1	30000	3	$ 35,491	7.6	228.089711
4	1	1	30000	4	$ 35,491	7.6	171.4390313
5	1	1	30000	5	$ 35,491	7.6	137.5343433
6	1	1	30000	6	$ 35,491	7.6	114.900662
7	1	1	30000	7	$ 35,491	1	98.64090567
8	1	1	30000	8	$ 35,491	1	86.44636541
9	1	1	30000	9	$ 35,491	1	76.95578607
10	1	1	30000	10	$ 35,491	1	69.36751653

Legend to understand the numbers:

- Configuration : (1) FWD1, (2) FWD2, (3) RWD1, (4) RWD2, (5) AWD2, or (6) AWD4
- Motorinverter_select : (1) A, (2) B, or (3) C
- slope_result : (1) Passed the requirement, or (0) Failed the requirement
- turning_result : (1) Passed the requirement, or (0) Failed the requirement

For each configuration and motor type, there are 100 iterations for different drive ratios and battery capacity, which will be filtered later based on the previously stated criteria.

FWD1 Motor A | B | C

Infeasible solutions:

Designs not meeting one of the requirements or have unrealistic results. All requirements will be checked if anyone with red flag will be mentioned in this section.

- **Max Speed and Drive Ratios infeasible [range 7 to 10]** are not feasible drive ratios as they are giving max speed less than 100 km/h. The reason for that is this high drive ratio consumes the max omega range of the motor very quickly before reaching the proper max speed.

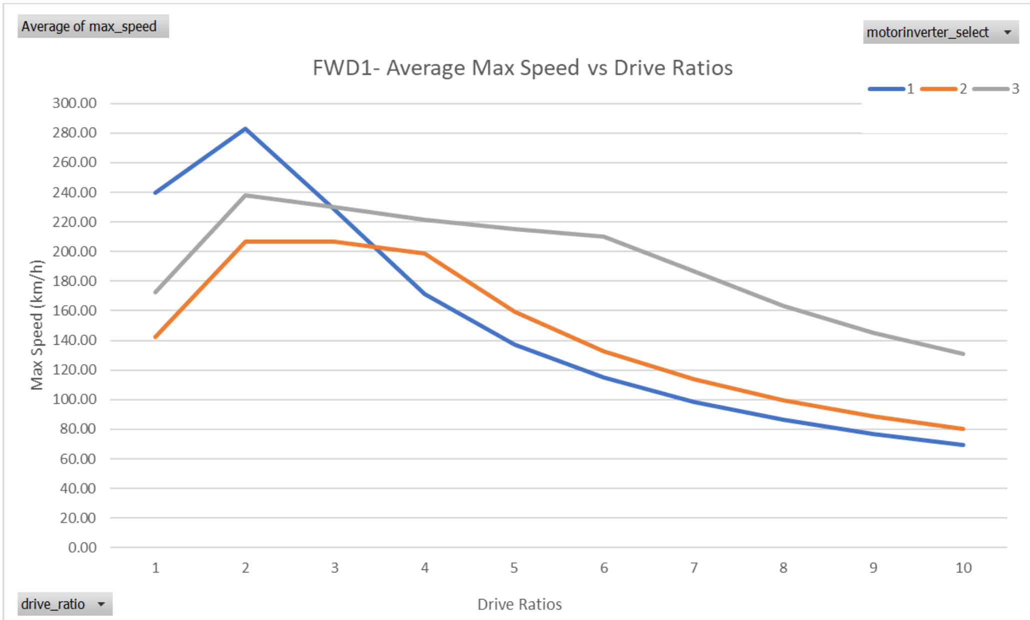
Drive Ratio	Max speed (each column represents different battery capacity)									
1	240.6562	240.4398	240.2232	240.0064	239.7894	239.5722	239.3548	239.1372	238.9194	238.7014
2	283.6669	283.5477	283.4284	283.3091	283.1898	283.0704	282.951	282.8315	282.7121	282.5926
3	228.0897	228.0857	228.0816	228.0783	228.0739	228.0715	228.0649	228.0657	228.0628	228.0602
4	171.439	171.4238	171.4104	171.4209	171.3914	171.3726	171.3612	171.338	171.3518	171.321
5	137.5343	137.5134	137.4834	137.4735	137.4549	137.4373	137.4207	137.4048	137.3895	137.3651
6	114.9007	114.9049	114.9062	114.8842	114.8607	114.8373	114.8198	114.7996	114.78	114.7612
7	98.64091	98.35409	98.36547	98.62505	98.63707	98.65335	98.65492	98.64522	98.65559	98.64552
8	86.44637	86.44765	86.44875	86.45113	86.44499	86.44845	86.45452	86.45702	86.45882	86.45762
9	76.95579	76.96021	76.95758	76.95536	76.95675	76.94838	76.96485	76.96735	76.969	76.96979
10	69.36752	69.36886	69.06008	69.17824	69.37146	69.3727	69.37454	69.37199	69.37633	69.37679

- **Zero Profit and Drive Ratio of 1** : as shown below, drive ratio of 1 shows 0 profit for all ranges due to the high acceleration time associated with it.

#	configuration	motorinverter_select	battery_capacity	drive_ratio	profit
191	1	2	120000	1	$ 0
181	1	2	110000	1	$ 0
171	1	2	100000	1	$ 0
161	1	2	90000	1	$ 0
151	1	2	80000	1	$ 0
141	1	2	70000	1	$ 0
131	1	2	60000	1	$ 0
121	1	2	50000	1	$ 0
111	1	2	40000	1	$ 0
101	1	2	30000	1	$ 0

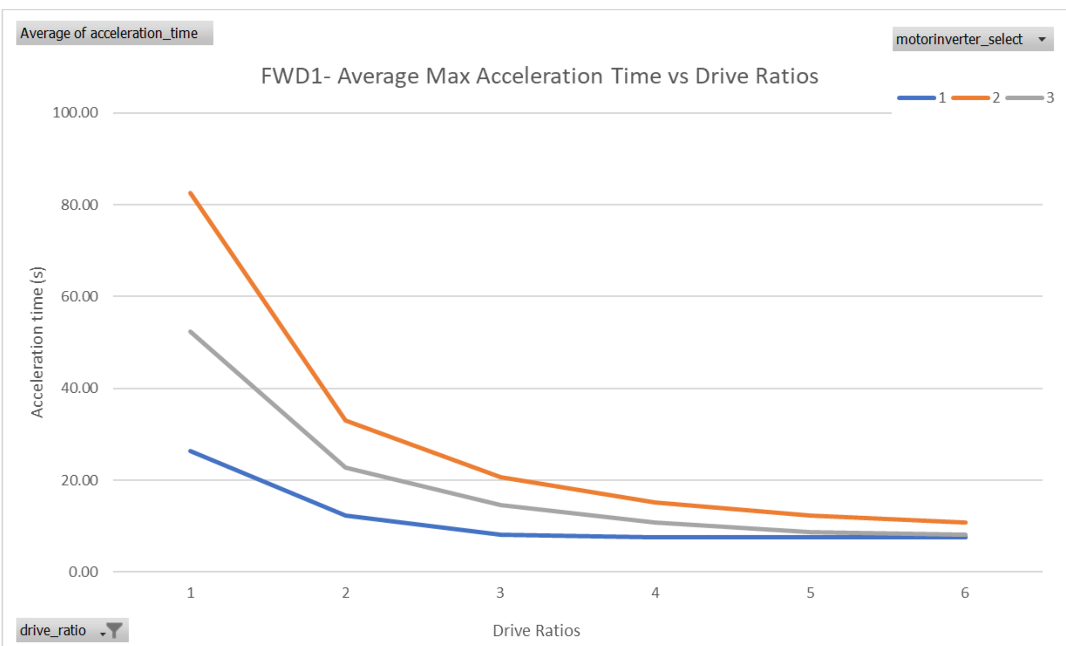
Drive ratio

A. **Drive Ratios** impact on **Max speed**



The peak average speed occurs with **drive ratio of 2**.

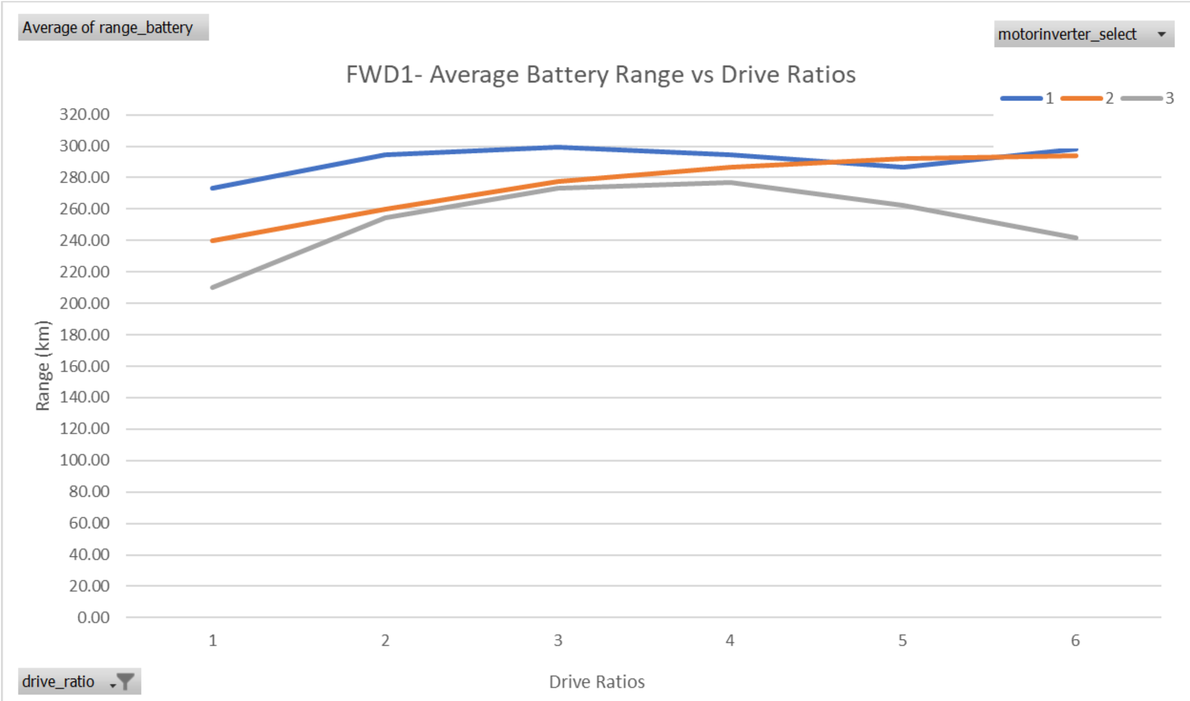
B. **Drive Ratios** impact on **Acceleration time**



The minimum acceleration time is happening as expected at the highest feasible **drive ratio of 6.**

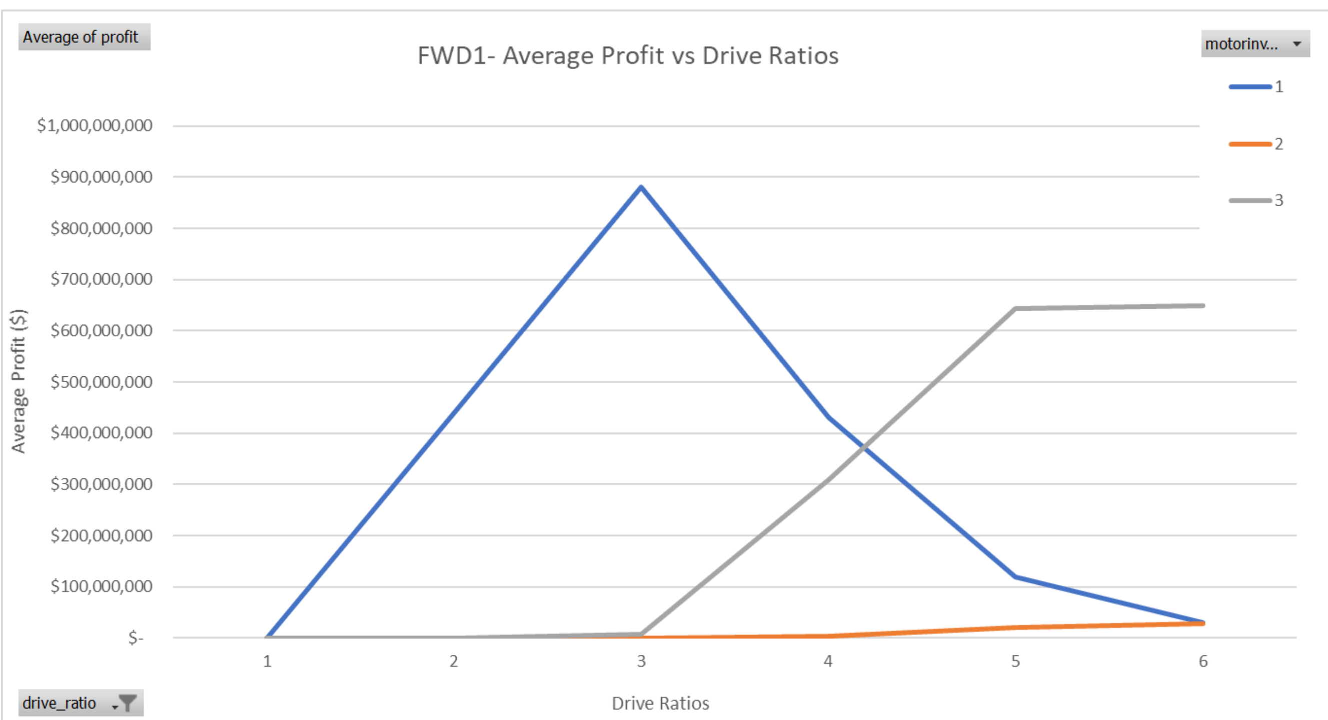
C. **Drive Ratio** Impact on **Cost** and **Range**

Based on the model, the DR would have no impact on the cost, and the impact on the range would be minimal with no clear trend.



D. Drive Ratio Impact on **Profit**

Excluding the infeasible solutions, the drive ration impact on profit is as below.



As per the graph the drive ratio influence on the profit is highly coupled with the type of motor used. The two significant one is **Drive ratio of 3 with motor A**, and **Drive ratio of 5 and 6 with motor C.**

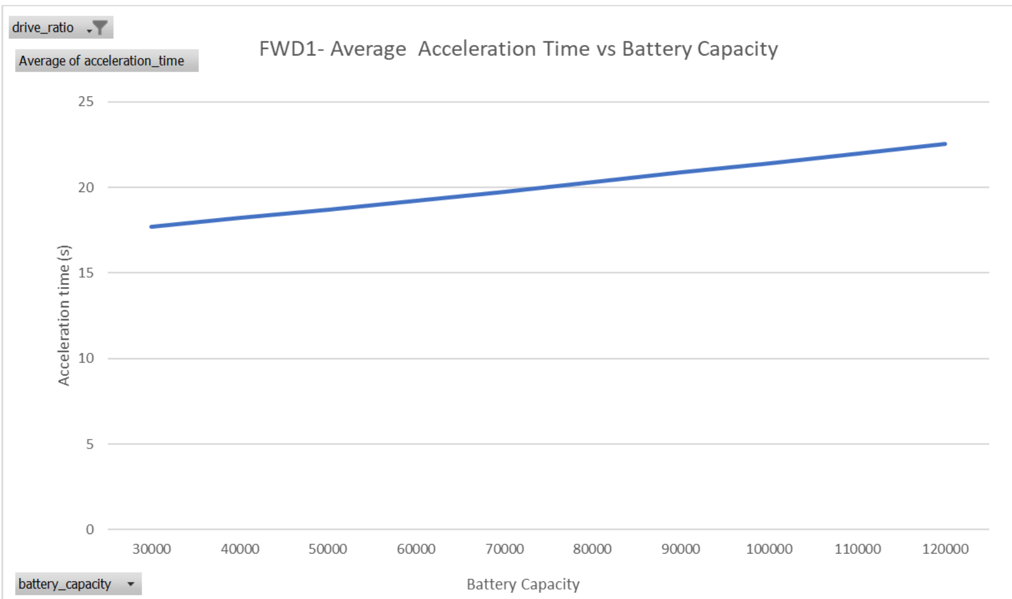
Compared to motor A, motor C has high speed range and lower torque, so it would be more beneficial for this motor to have high drive ratio to increase the operating torque over the speed range it has. On the other hand, motor A has a high torque operating range, that is why it is more beneficial to have lower torque for the same logic. Motor B has no edge over the two motors, that's why it is not have peak profit.

Battery Capacity

A. **Battery Capacity** impact on **Max speed**

Battery capacity is expected to have no impact on the maximum speed.

B. **Battery Capacity** impact on **Acceleration time**

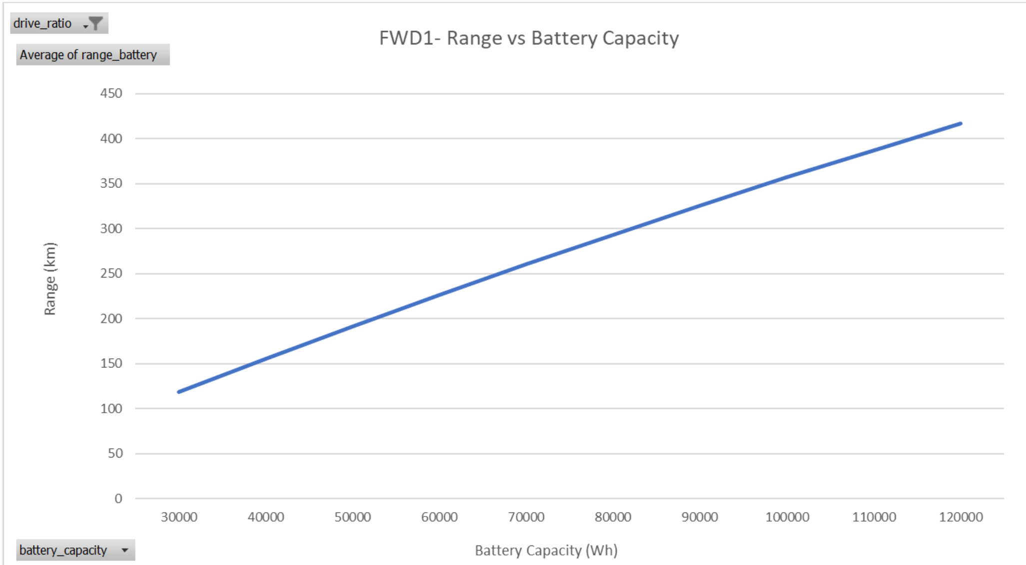


The higher the battery capacity the higher the higher the mass corresponding to this capacity. That is why this larger inertia will consume from the propulsion energy, that is why we can see that the higher the capacity is the higher the acceleration time.

D. **Battery Capacity** Impact on **Cost**

The higher the battery capacity the higher the cost with a linear correlation.

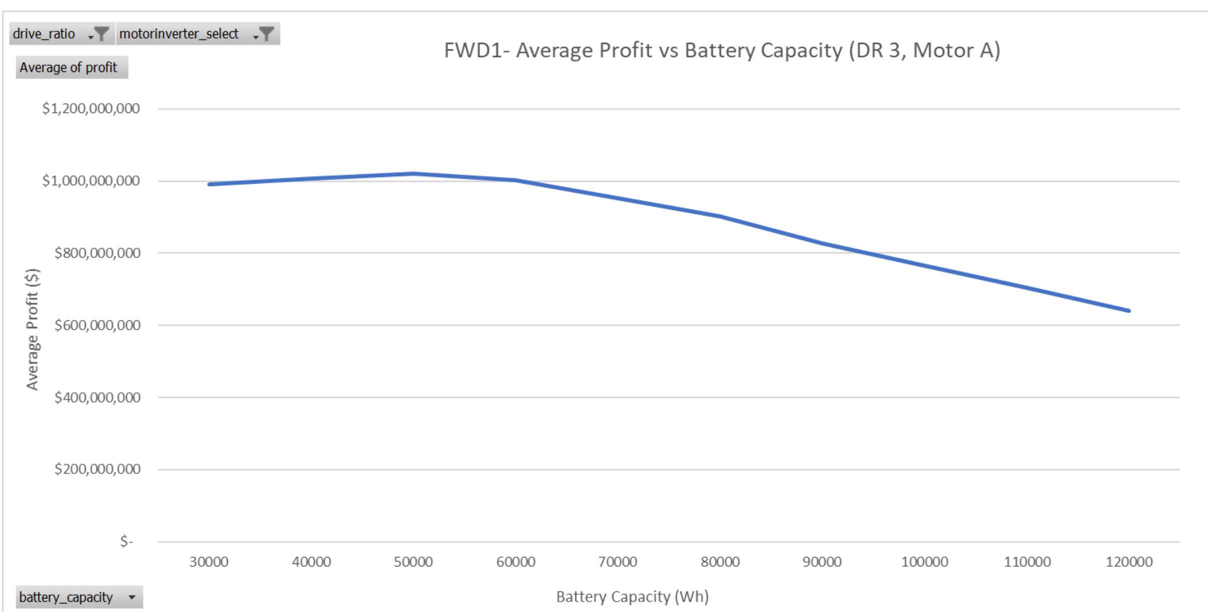
E. **Battery Capacity** impact on **Range**

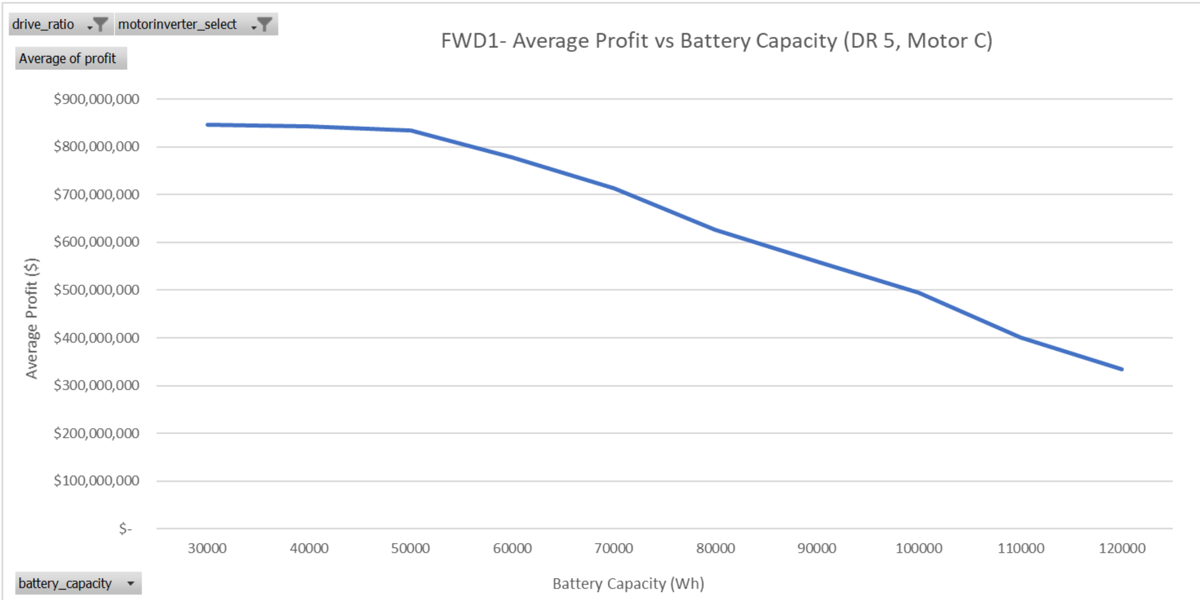


F. **Battery Capacity** impact on **Profit**

To be able to fairly assess the battery capacity impact on cost, this should be tested while one of each design variables.

The drive ratio of 3 with motor A was selected for the first graph, and drive ration 5 with Motor C for the second graph.

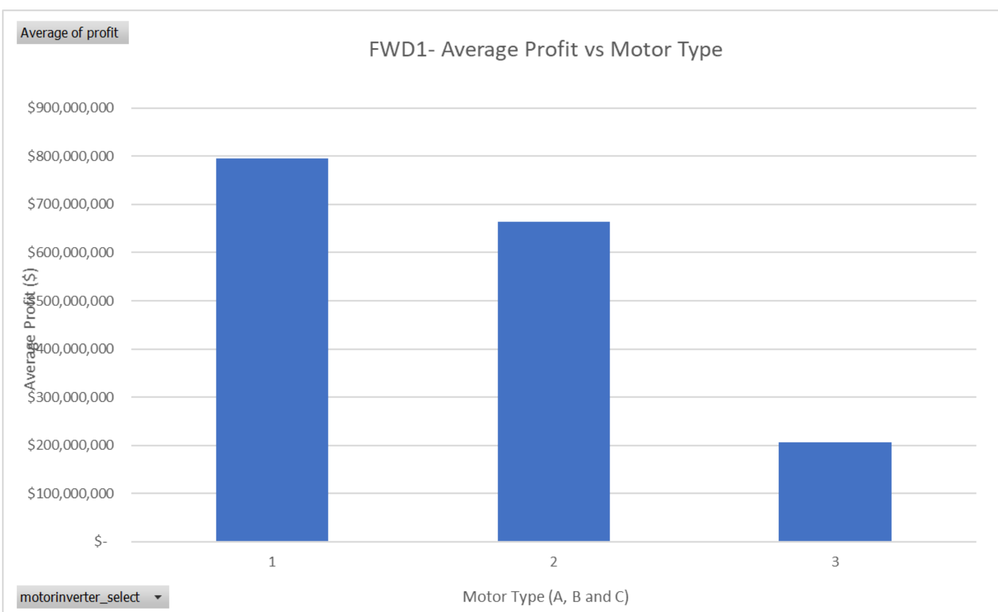




Battery capacity of 50 kWh has a peak profit for DR3, Motor A case; and battery capacity from 30-50 kWh shows profit peak for DR5, and motor C.

Motor type

A. **Motor type** impact on **Profit**



Motor A has on average higher profit for all iterations compared to the other motors.

FWD1 - Conclusion

For the FWD1, the **best ranges** for each design variable are as below:

Variable	Best Value	Comment	Feasible
Configuration	FWD1		-
Drive Ratio	3	with motor A	2:6
Battery Capacity	50 kWh		30:50 kWh
Motor Type	A		A, and C
Max Profit	$1,020,218,949		-

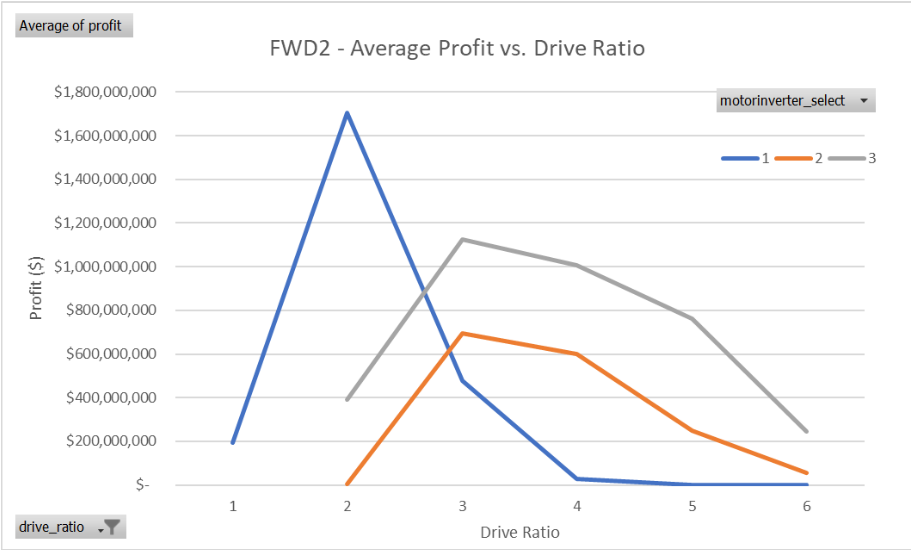
The solution with the max profit for the FWD1 is:

configuration	FWD1
motorinverter_select	A
battery_capacity	50,000
drive_ratio	3
profit	$ 1,020,218,949
cost	$ 38,391
acceleration_time	7.6
max_speed	228.1
range_battery	210.6
slope_result	1
stress	144.23
deflection_max	0.01
Kus	0.19
lateral_acceleration_max	17.27
diameter	99.20
turning_result	1
spring_rate_front	15248
spring_rate_rear	18854
spring_damping_coeffiecient_front	1498
spring_damping_coeffiecient_rear	1544

FWD2 Motor A | B | C

This configuration has the same physics of the first configuration, that is why it had the same infeasible ranges and trends for each of the design variables, but different values. Below is the summary for this configuration with each design variable to the profit.

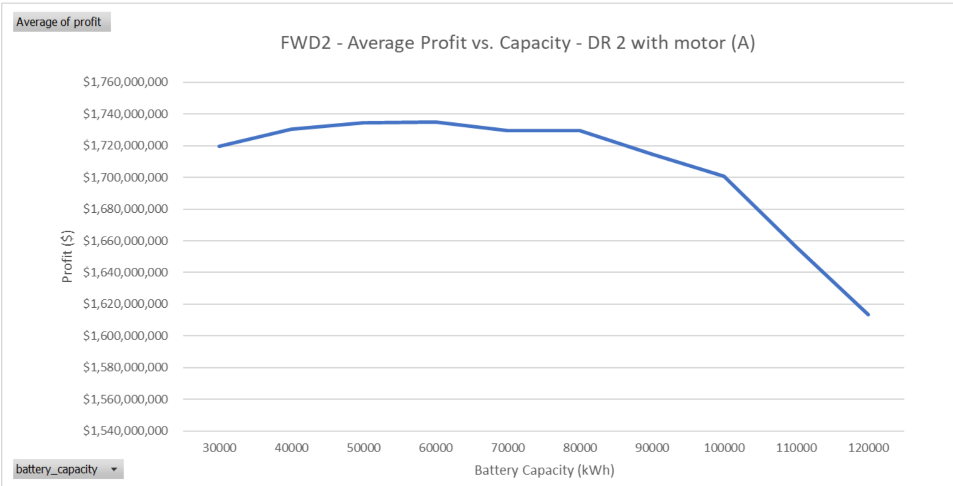
Drive Ratio



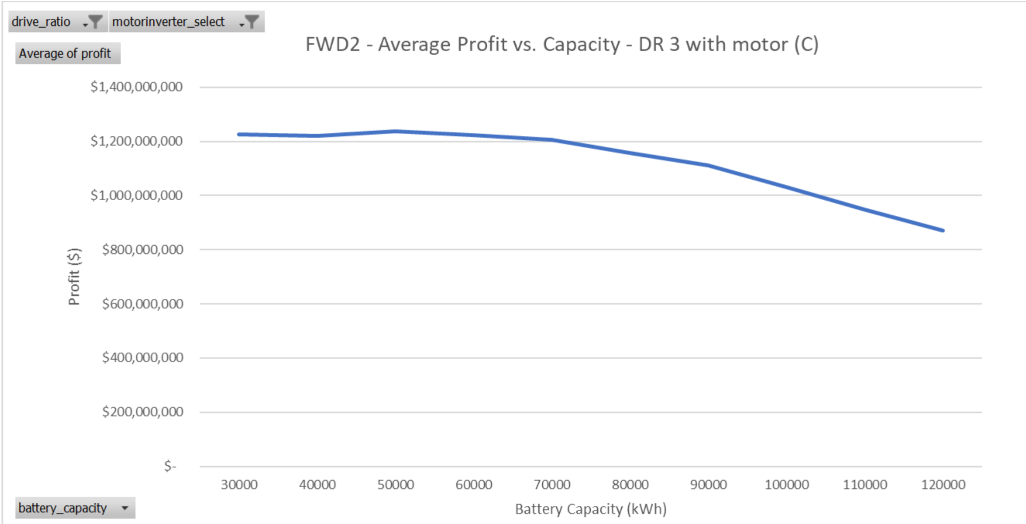
DR of 2 with motor **A**, **DR of 3** with motors **B** and **C**

The reason for having closer best DR between the motors in FWD2 compared to FWD1 is in FWD2 the low torque of motor C which required high DR doesn't exist here due to the torque multiplication, as we have two motors working on the same axles multiplying the torque.

Battery Capacity

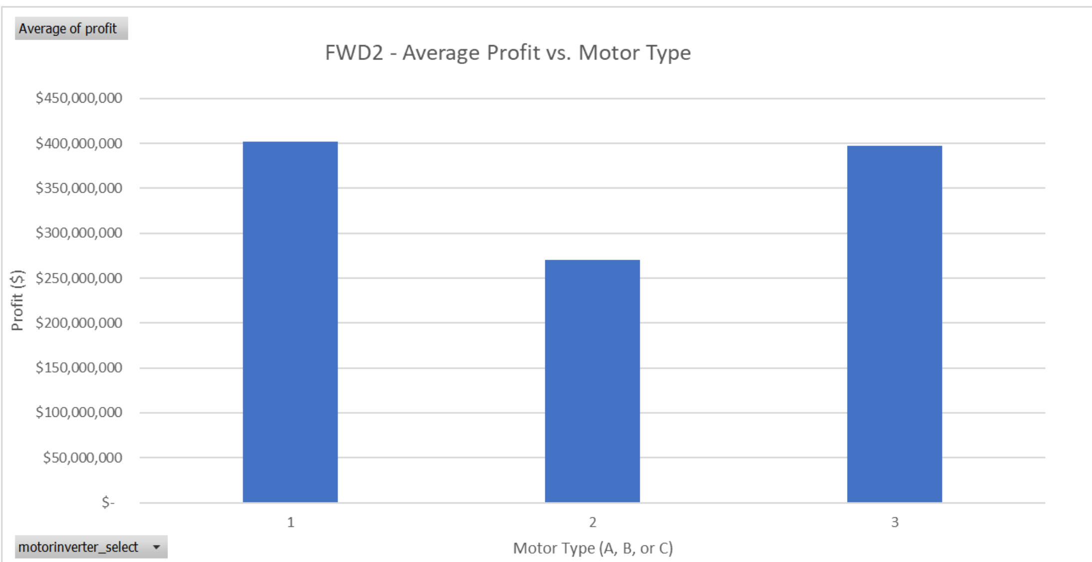


With DR 2, and motor A, the peak profit is at **battery capacity of 60 kWh**



With DR 3, and motor C, the peak profit is at **battery capacity of 50 kWh**

Motor Type



Motors A and B have the highest profit average in this configuration.

FWD2 - Conclusion

For the FWD2, the **best ranges** for each design variable are as below:

Variable	Best Value	Comment	Feasible
Configuration	FWD2		-
Drive Ratio	2	with motor A	2:6
Battery Capacity	60 kWh		50:60 kWh
Motor Type	A		A, B, and C
Max Profit	$ 1,735,014,861		-

The solution with the max profit for the FWD2 is:

configuration	FWD2
motorinverter_select	A
battery_capacity	60,000
drive_ratio	2
profit	$ 1,735,014,861
cost	$ 47,341
acceleration_time	6.9
max_speed	341.98
range_battery	235.67
slope_result	Pass
stress	139.0
deflection_max	0.011
Kus	0.259
lateral_acceleration_max	15.57
diameter	100.09
turning_result	Pass
spring_rate_front	18820.15
spring_rate_rear	19015.21
spring_damping_coeffiecient_front	1848.96
spring_damping_coeffiecient_rear	1556.77

RWD1 Motor A | B | C

Infeasible solutions:

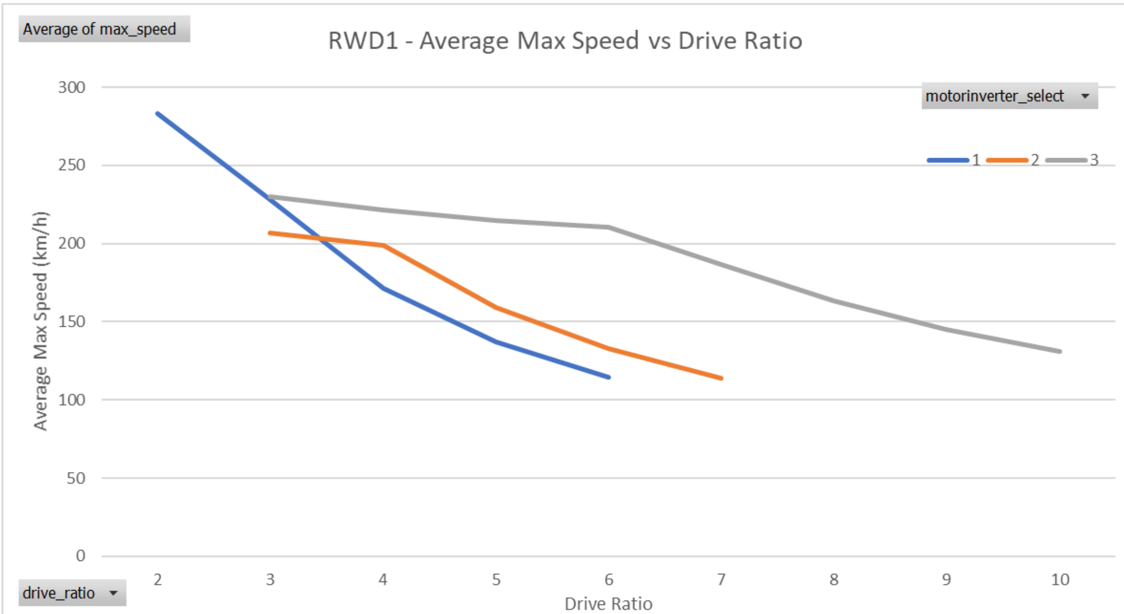
Designs not meeting one of the requirements or have unrealistic results. All requirements will be checked if anyone with red flag will be mentioned in this section.

- **Max Speed** like FWD there are some of the solutions does not have a max speed higher than 100km/h. However, the gear ratios casing this low speed are not always causing this low speed, so the filtering will depend on eliminating only this infeasible solution not whole drive ratios.
- **Zero Profit and Drive Ratio of 1** : as shown below, drive ratio of 1 shows 0 profit for all ranges due to the high acceleration time associated with it.

motorinverter_select	battery_capacity	drive_ratio	profit	cost	acceleration_time	max_speed	range_battery
2	120000	1	$ 0	$ 42,741	95.2	140.6500262	376.3124437
2	110000	1	$ 0	$ 41,291	92.3	141.0196605	347.1922216
2	100000	1	$ 0	$ 39,841	89.4	141.3883286	317.9486858
2	90000	1	$ 0	$ 38,391	86.6	141.7560378	286.3868238
2	80000	1	$ 0	$ 36,941	83.8	142.1227957	256.7815727
2	70000	1	$ 0	$ 35,491	81	142.4886096	225.5355956
2	60000	1	$ 0	$ 34,041	78.3	142.8534867	194.3927622
2	50000	1	$ 0	$ 32,591	75.6	143.2174342	163.6610311
2	40000	1	$ 0	$ 31,141	73	143.5804592	130.8830149
2	30000	1	$ 0	$ 29,691	70.4	143.9425686	99.50498884
3	120000	1	$ 0	$ 46,541	59.7	171.3617208	323.4729381

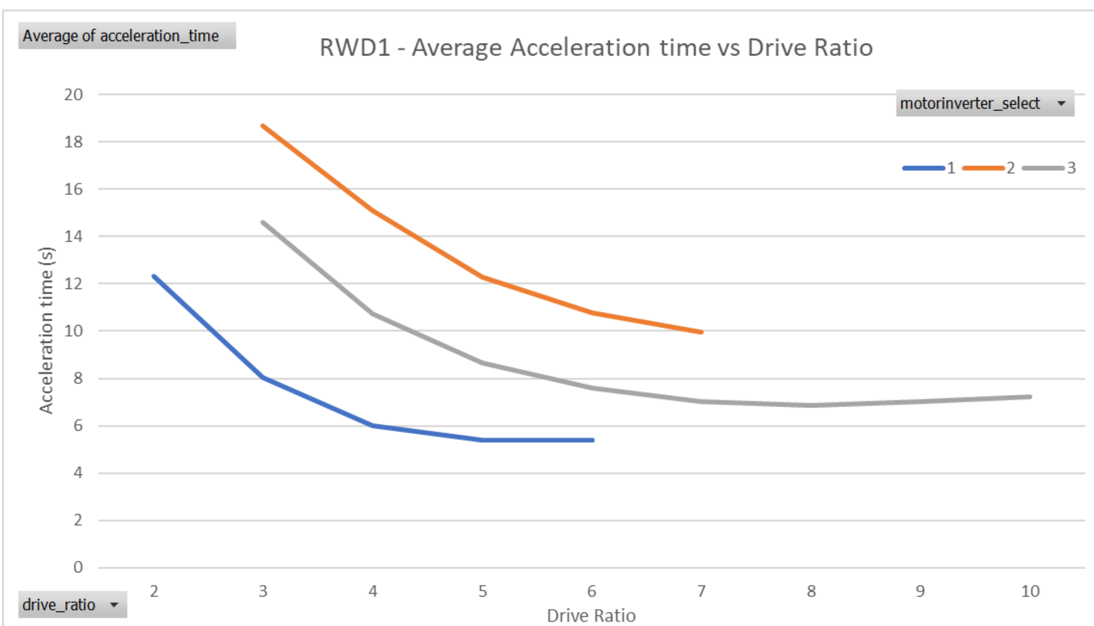
Drive ratio

A. **Drive Ratios** impact on **Max speed**



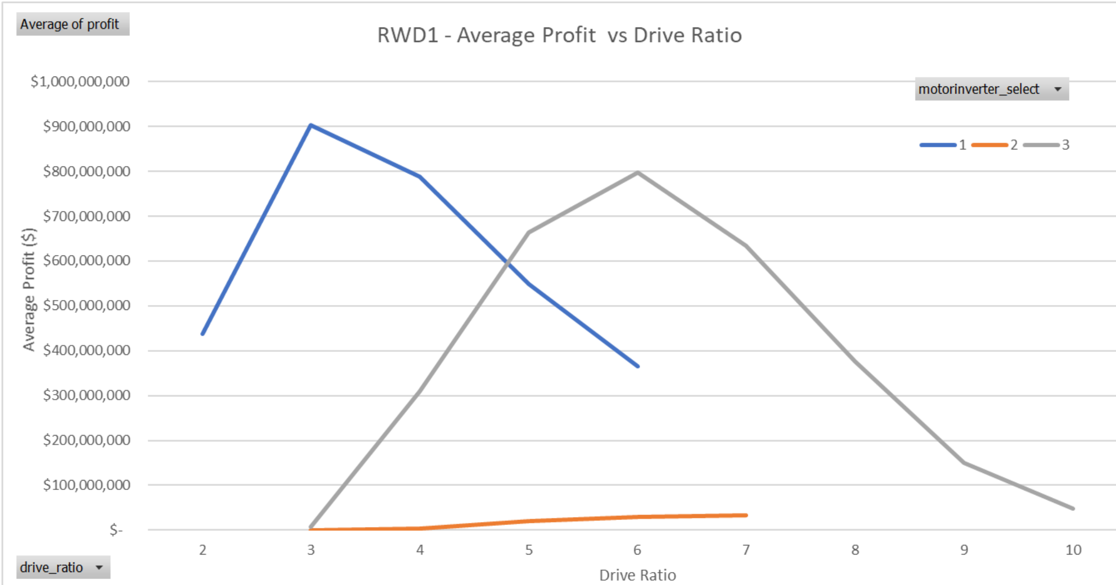
Drive ratios 2 with motor A, and 3 with motors B and C are having the highest max speed km/h.

B. **Drive Ratios** impact on **Acceleration time**



The highest drive ratios for each motor are the best drive ratios for the acceleration time. Please note that the disconnected lines are due to the infeasible solutions filtration; so, for motor A for example, the higher drive ratios than 6 are definitely delivering better acceleration time (less), but this high DR, will cause consuming all speed range of the motor very quickly, so the vehicle won't be able to achieve reasonable max speed.

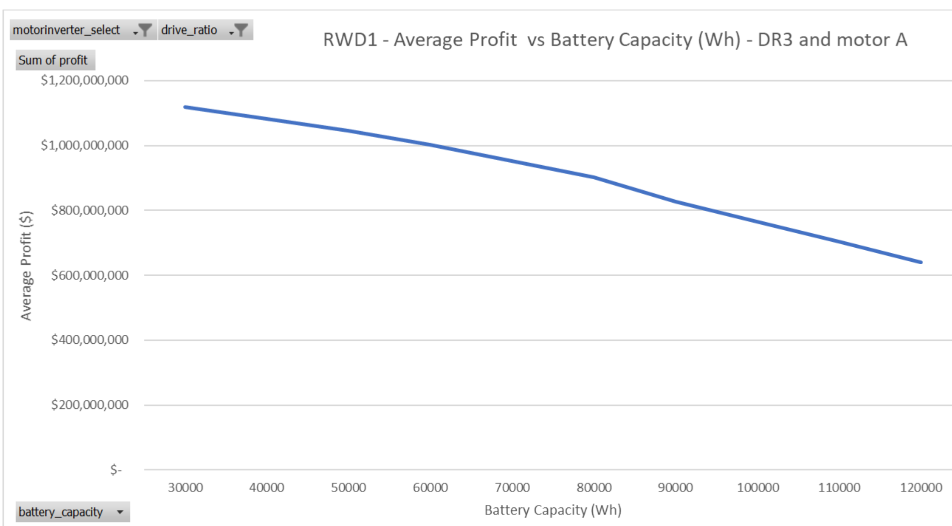
C. **Drive Ratios** impact on **profit**

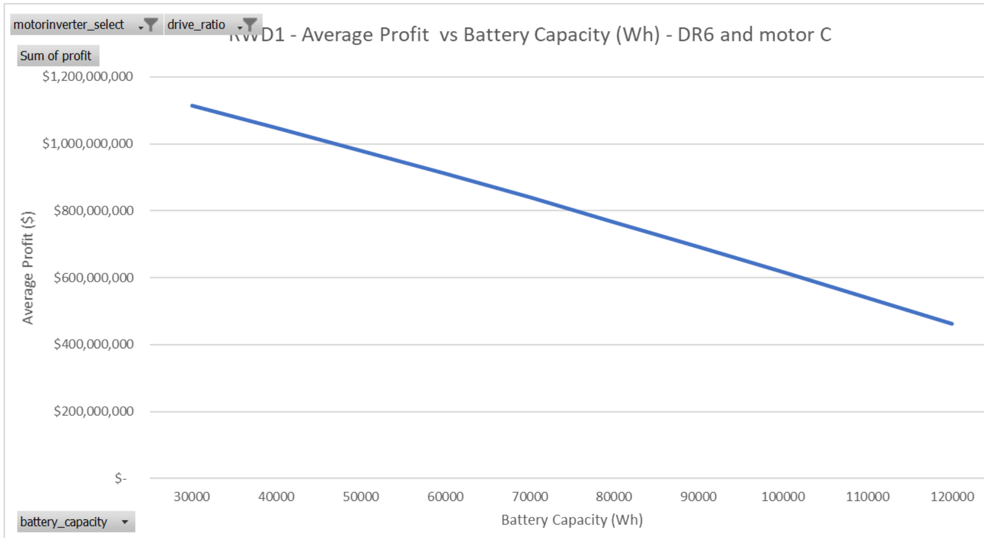


The most profitable combination of drive ratio and motors are drive ratio of 3 and motor A, and drive ratio of 6 and for motor C. (Note that this is a combination of acceleration time and max speed results).

Battery Capacity

A. **Battery Capacity** impact on **Profit**

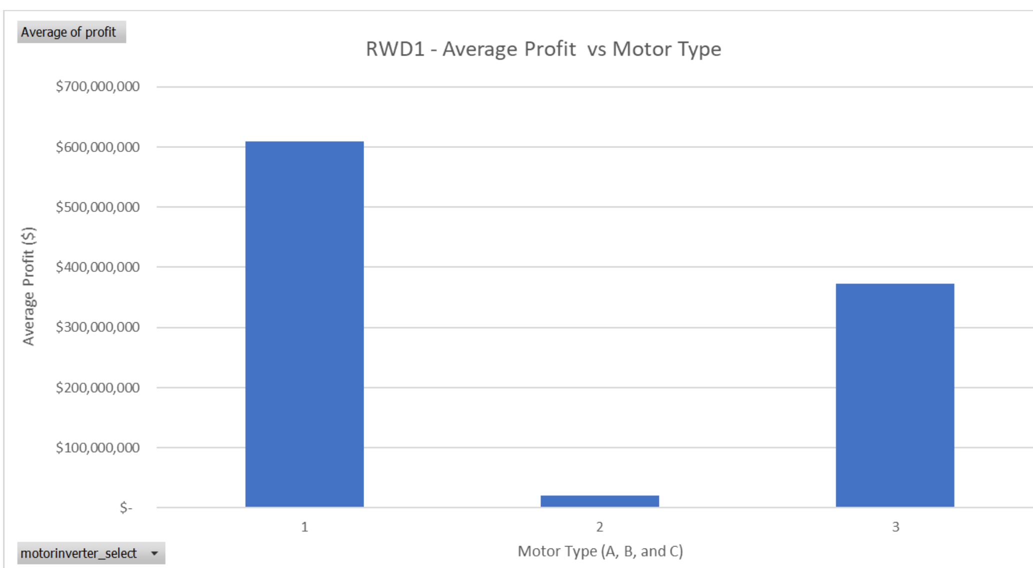




The peak for profit here is outside the tested range, it will be assumed to be <30 kWh. This means ultimately the best solution could be zero kWh; accordingly, this could be infeasible solution. This to be considered while selecting the best optimized solution at the end to ignore this one as it is unrealistic.

Motor Type

A. **Motor Type** impact on **Profit**



Motor A has the highest profit average.

RWD1 - Conclusion

For the RWD1, the **best ranges** for each design variable are as below:

Variable	Best Value	Comment	Feasible
Configuration	RWD1		-
Drive Ratio	3	with motor A	2:6
Battery Capacity	<30kWh		30:120 kWh
Motor Type	A		A, B, and C
Max Profit	$ 1,119,194,306		-

The solution with the max profit for the RWD2 is as below, technically this is not the best solution here as the best solution could be having a battery capacity of 0 kWh (please go to battery-profit section). If eventually this was the best optimized solution, it should be investigated more (lower battery ranges to find the peak) or ignored if the peak is zero kWh:

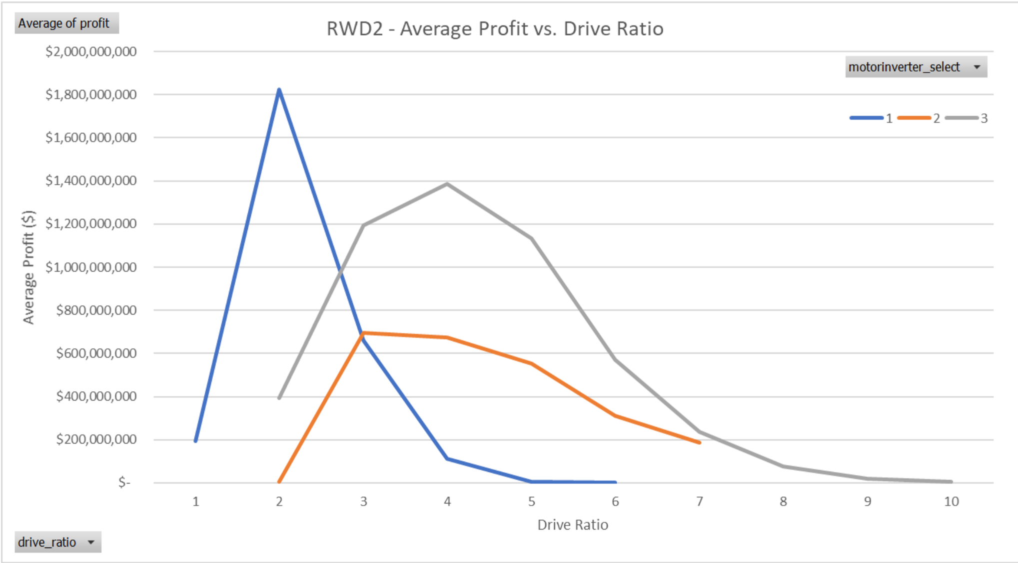
configuration	3
motorinverter_select	1
battery_capacity	30000
drive_ratio	3
profit	$ 1,119,194,306
cost	$ 35,491
acceleration_time	7
max_speed	228
range_battery	130
slope_result	1
stress	159
deflection_max	0
Kus	0
lateral_acceleration_max	75
diameter	99
turning_result	1
spring_rate_front	10215
spring_rate_rear	23450
spring_damping_coeffiecient_front	1004
spring_damping_coeffiecient_rear	1920

RWD2 Motor A | B | C

As RWD2 has the same physics as RWD1, almost both of them have the same trends, please find the summary below:

Drive ratio

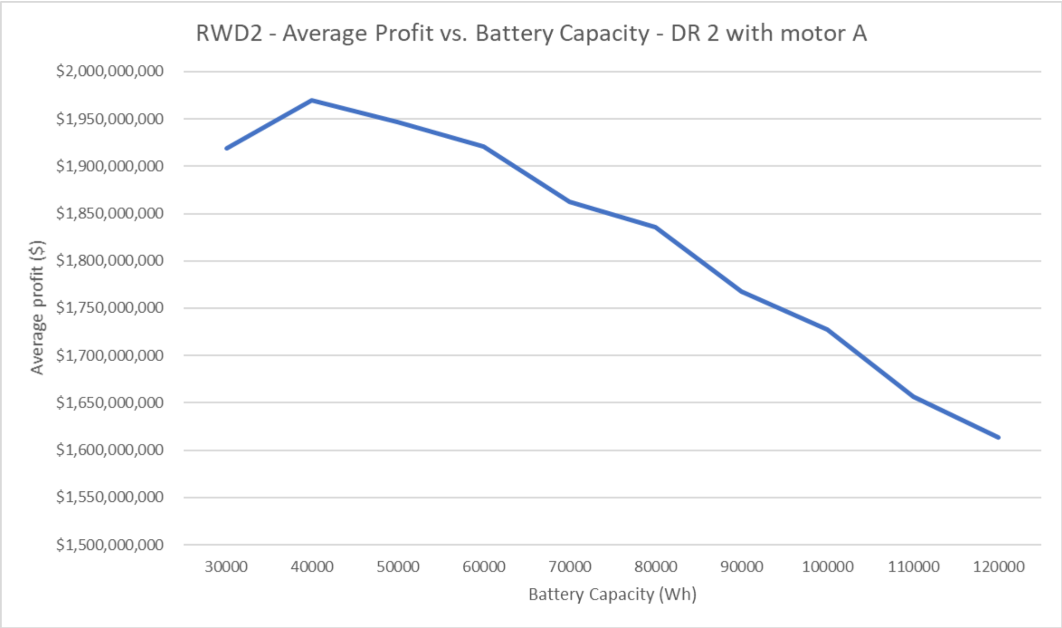
A. Drive Ratios impact on Profit



The tradeoff point is the drive ratio of 2 for motor A, drive ratio of 4 for motor C, and drive ratio of 3 for motor B.

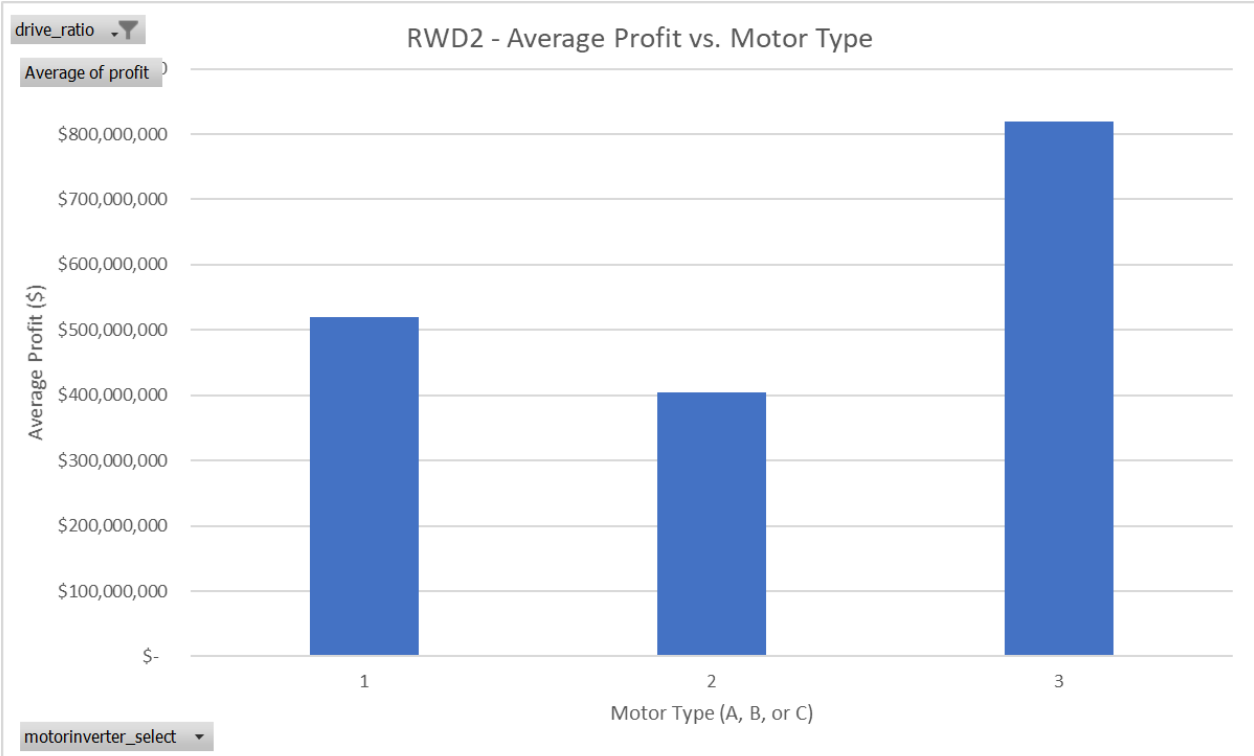
Battery Capacity

A. Battery Capacity impact on Profit



The peak profit is at battery capacity of **40 kWh**.

Motor Type



Motors C and A has the highest average profit of the iterations.

RWD2 - Conclusion

For the RWD2, the **best ranges** for each design variable are as below:

Variable	Best Value	Comment	Feasible
Configuration	RWD2		-
Drive Ratio	2	with motor A	2:7
Battery Capacity	40 kWh		30:120 kWh
Motor Type	C		A, B, and C
Max Profit	$ 1,969,301,800		-

The solution with the max profit for the RWD2 is:

configuration	RWD2
motorinverter_select	A
battery_capacity	40,000
drive_ratio	2
profit	$ 1,969,301,800
cost	$ 44,441
acceleration_time	6
max_speed	342.0

range_battery	161.9
slope_result	Pass
stress	160.6
deflection_max	0.0
Kus	0.1
lateral_acceleration_max	27.0
diameter	99.9
turning_result	Pass
spring_rate_front	14116.3
spring_rate_rear	23136.4
spring_damping_coeffiecient_front	1386.8
spring_damping_coeffiecient_rear	1894.2

AWD2 Motor A | B | C

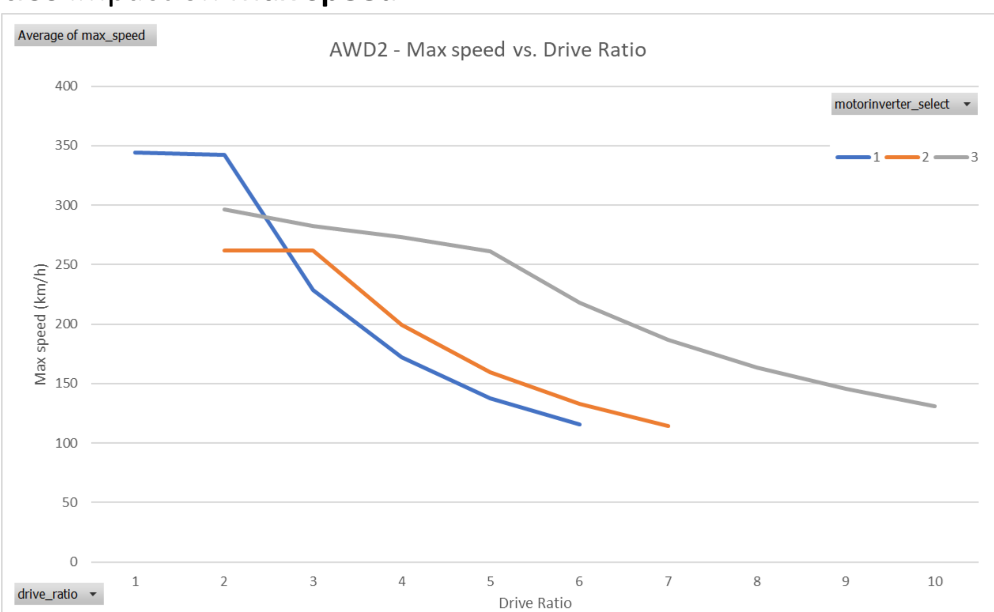
Infeasible solutions:

Designs not meeting one of the requirements or have unrealistic results. All requirements will be checked if anyone with red flag will be mentioned in this section.

- **Max Speed** any solution with max speed less than 100 km/h has been filtered.
- **Zero Profit** any solution with profit equals zero has been filtered.
- **Acceleration time** any solution has the acceleration time higher than 100 s has been filtered.

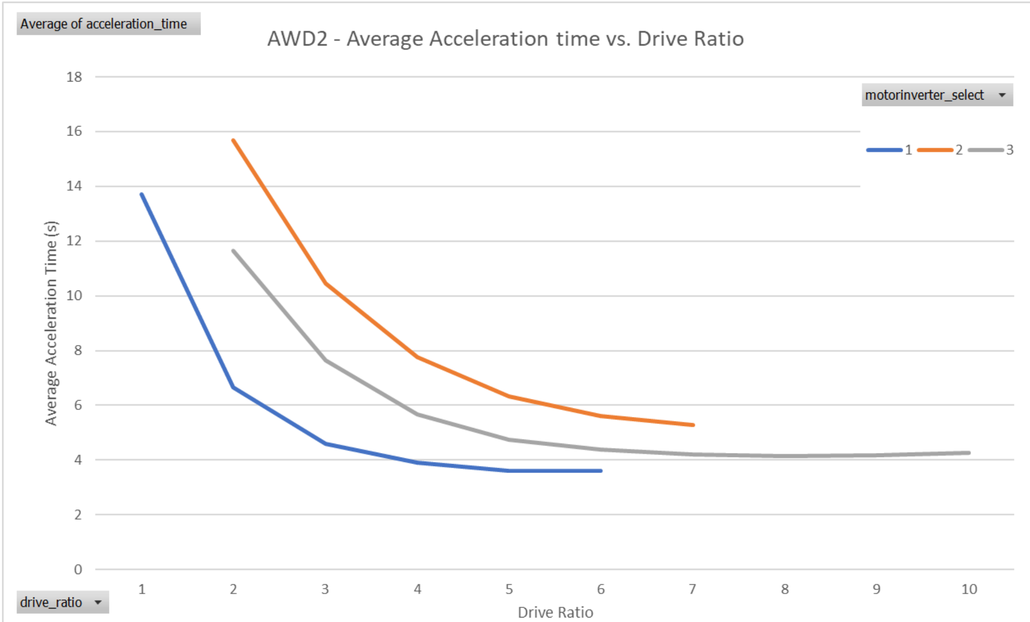
Drive ratio

A. Drive Ratios impact on Max speed



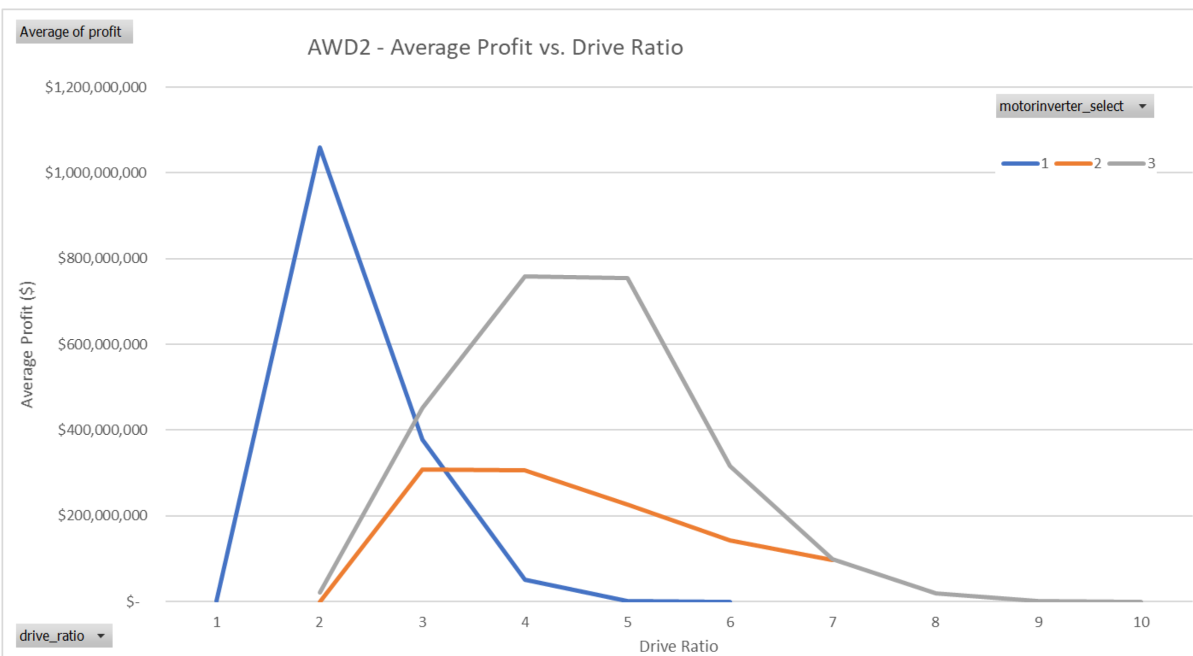
Drive ratios 1 and 2 with motor A, 3 with motors B, and 2 with C are having the highest max speed km/h.

B. **Drive Ratios** impact on **Acceleration time**



The highest drive ratios for each motor are the best drive ratios for the acceleration time.

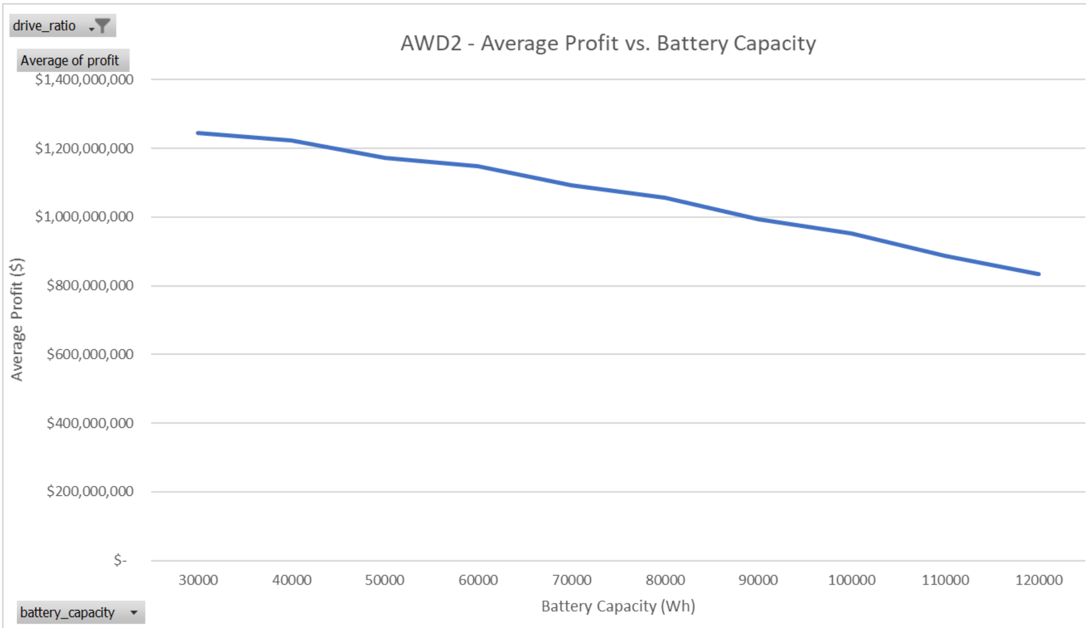
C. **Drive Ratios** impact on **profit**



The most profitable combination of drive ratio and motors are drive ratio of 2 and motor A, and drive ratio of 4 and 5 for motor C.

Battery Capacity

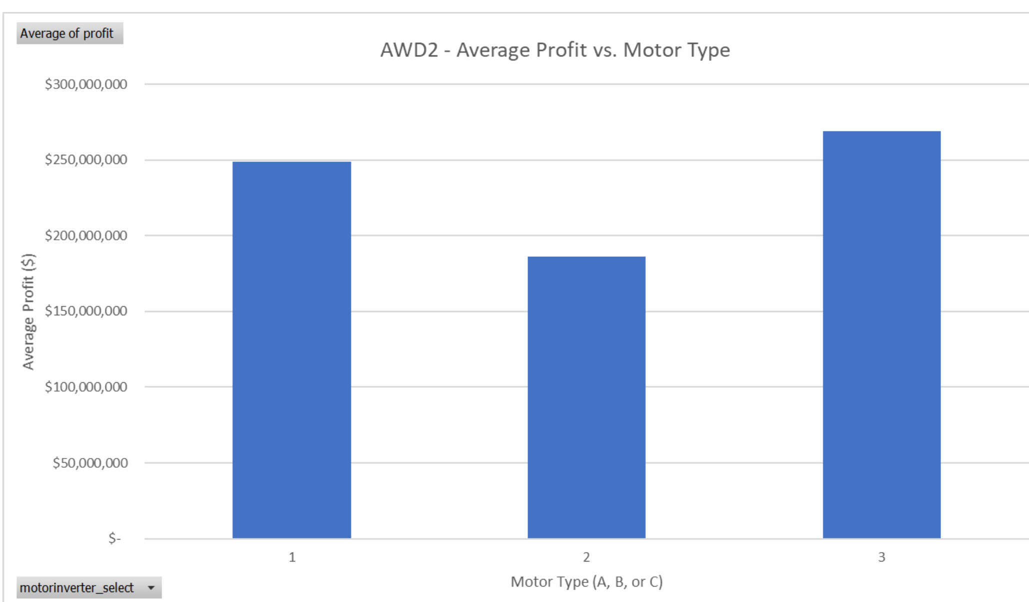
A. **Battery Capacity** impact on **Profit**



The peak for profit here is outside the tested range, it will be assumed to be <30 kWh. This means ultimately the best solution could be near zero kWh; accordingly, this could be infeasible solution. This to be considered while selecting the best optimized solution at the end to ignore this one as it is unrealistic.

Motor Type

A. **Motor Type** impact on **Profit**



Motor C has the highest profit average.

AWD2 - Conclusion

For the AWD2, the **best ranges** for each design variable are as below:

Variable	Best Value	Comment	Feasible
Configuration	AWD2		-
Drive Ratio	2	with motor A	2:7
Battery Capacity	<30kWh		30:120 kWh
Motor Type	C		A, B, and C
Max Profit	$ 1,244,916,140		-

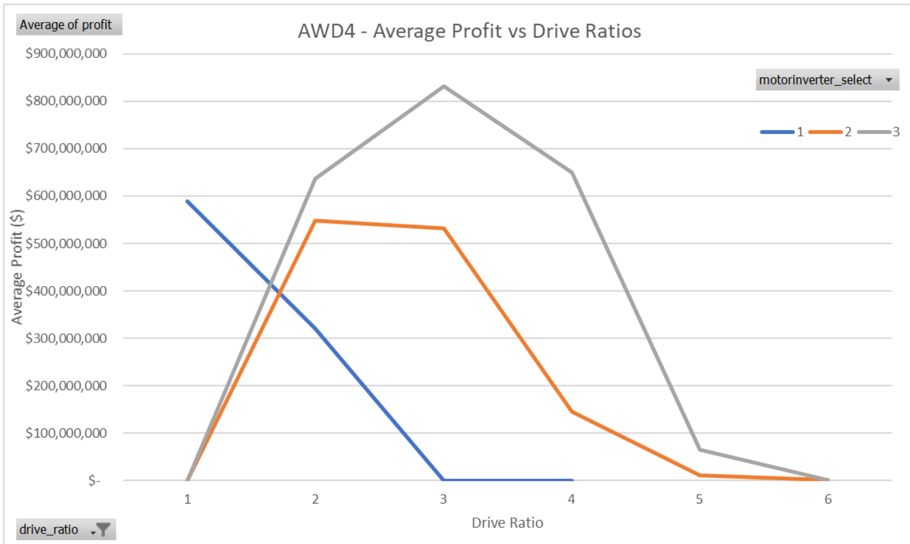
The solution with the max profit for the AWD2 is as below, technically this is not the best solution here as the best solution could be having a battery capacity near 0 kWh (please go to battery-profit section). If eventually this was the best optimized solution, it should be investigated more (lower battery ranges to find the peak) or ignored if the peak is near zero kWh:

configuration	AWD2
motorinverter_select	A
battery_capacity	30,000
drive_ratio	2
profit	$ 1,244,916,141
cost	$ 50,491
acceleration_time	6
max_speed	342
range_battery	121
slope_result	Passed
stress	144
deflection_max	0.010
Kus	0.146
lateral_acceleration_max	26
diameter	100
turning_result	Passed
spring_rate_front	14326
spring_rate_rear	23332
spring_damping_coeffiecient_front	1407
spring_damping_coeffiecient_rear	1910

AWD4 Motor A | B | C

Drive ratio

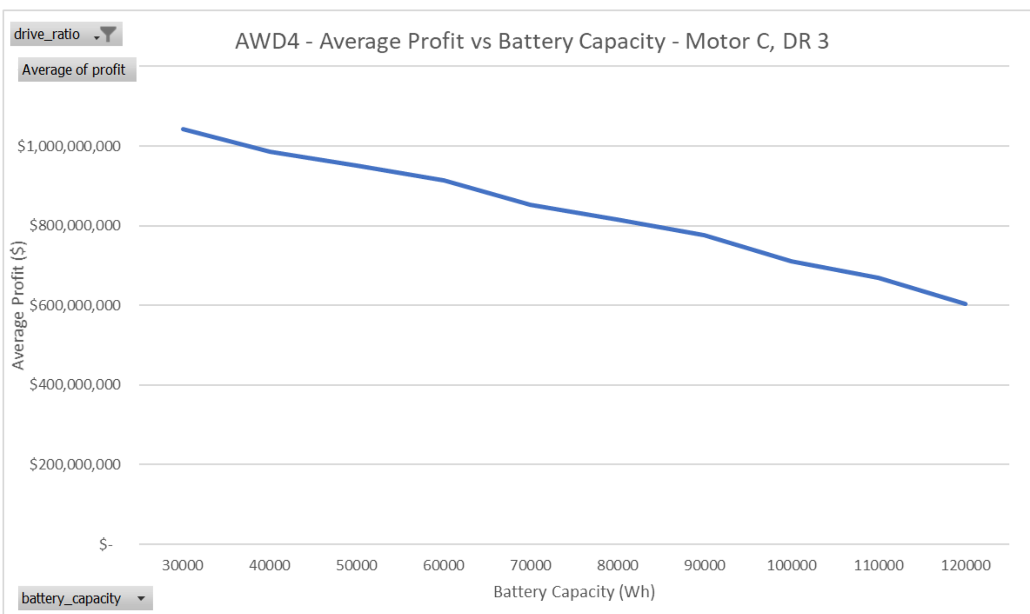
A. **Drive Ratios** impact on **profit**



Drive ratios 1 with motor A, 2 with motors B, and 3 with C are having the highest profit.

Battery Capacity

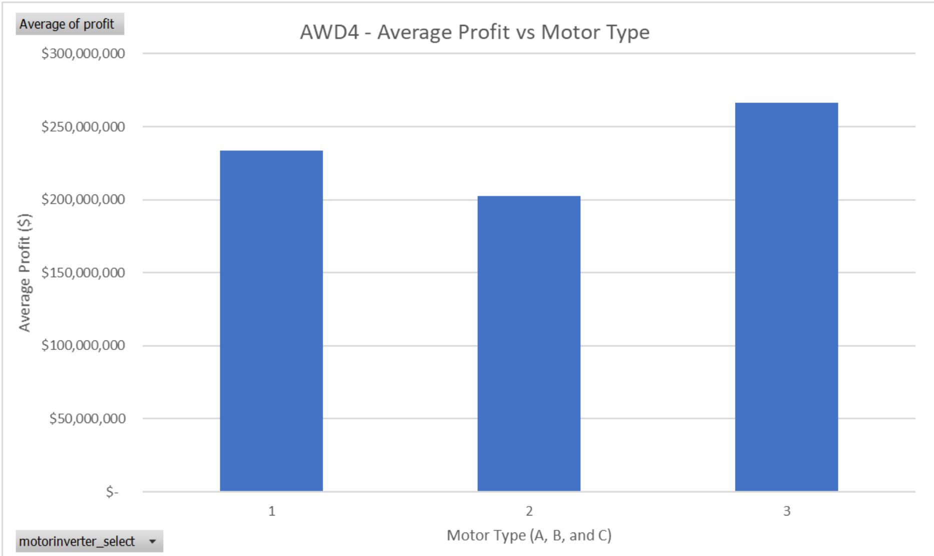
A. **Battery Capacity** impact on **Profit**



The peak for profit here is outside the tested range, it will be assumed to be <30 kWh. This means ultimately the best solution could be near zero kWh; accordingly, this could be infeasible solution. This to be considered while selecting the best optimized solution at the end to ignore this one as it is unrealistic.

Motor Type

A. **Motor Type** impact on **Profit**



Motor C has the highest profit average.

AWD4 - Conclusion

For the AWD4, the **best ranges** for each design variable are as below:

Variable	Best Value	Comment	Feasible
Configuration	AWD4		-
Drive Ratio	3	with motor A	1:5
Battery Capacity	<30kWh		30:120 kWh
Motor Type	C		A, B, and C
Max Profit	$ 1,042,029,349		-

The solution with the max profit for the AWD4 is as below, technically this is not the best solution here as the best solution could be having a battery capacity near 0 kWh (please go to battery-profit section). If eventually this was the best optimized solution, it should be investigated more (lower battery ranges to find the peak) or ignored if the peak is near zero kWh:

configuration	AWD4
motorinverter_select	C
battery_capacity	30000
drive_ratio	3
profit	$ 1,042,029,350
cost	$ 57,491
acceleration_time	4.1
max_speed	348.965

range_battery	86.474
slope_result	Pass
stress	150.233
deflection_max	0.011
Kus	0.161
lateral_acceleration_max	23.050
diameter	100.493
turning_result	Pass
spring_rate_front	15049.114
spring_rate_rear	23218.808
spring_damping_coeffiecient_front	1478.482
spring_damping_coeffiecient_rear	1900.920

SUMMARY AND THE OPTIMIZED DESIGN

Based on the above, the best design and configuration is the RWD2 with motor A, and capacity of 40kWh.

PRELIMINARY Variables Values for the Optimized design	
configuration	RWD2
motorinverter_select	A
battery_capacity	40,000
drive_ratio	2
profit	$ 1,969,301,800
cost	$ 44,441
acceleration_time	6
max_speed	342.0
range_battery	161.9
slope_result	Pass
stress	160.6
deflection_max	0.011
Kus	0.1
lateral_acceleration_max	27.0
diameter	99.9
turning_result	Pass
spring_rate_front	14116.3
spring_rate_rear	23136.4
spring_damping_coeffiecient_front	1386.8
spring_damping_coeffiecient_rear	1894.2

However, this is not the final optimized design yet, as the best frame rail design for this configuration is not calculated yet. Using the same method used to determine the first frame rail dimensions, the optimization code will be activated again to determine the best dimensions.

The output of the frame rail optimization was as below:

Width	Aspect Ratio	Thickness	Area	Max Stress	Max Deflection	No	height
0.05	3.5	0.001	0.000446	214.097	0.01495	346	0.175
0.045	4	0.001	0.000446	214.774	0.01458	226	0.18
0.04	5	0.001	0.000476	190.503	0.01166	121	0.2
0.06	3	0.001	0.000476	188.096	0.01279	601	0.18
0.08	2	0.001	0.000476	190.499	0.01458	1111	0.16
0.07	2.5	0.001	0.000486	180.984	0.01267	856	0.175
0.055	3.5	0.001	0.000491	177.219	0.01128	481	0.1925
0.045	4.5	0.001	0.000491	178.478	0.01080	241	0.2025
0.1	1.5	0.001	0.000496	180.870	0.01478	1636	0.15
0.05	4	0.001	0.000496	174.275	0.01068	361	0.2

The highlighted cells are the lowest values from each variable, as all these ones have almost the same area, and all of them meet the requirements of max deflection and stress. The yellow highlighted option seems to be a good option as it has low height, good aspect ratio, low area, and meeting all the requirements.

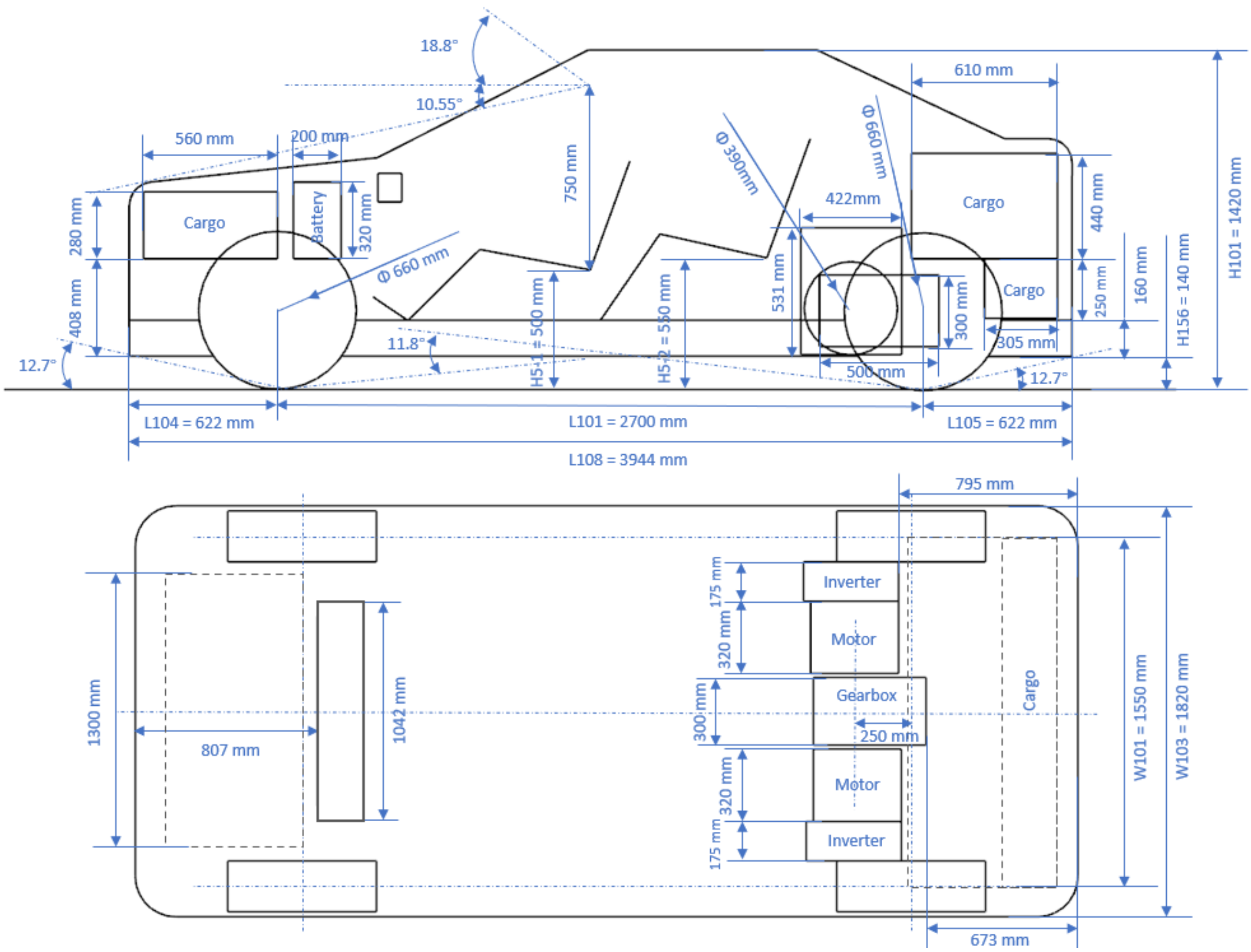
Updating the dimensions of the frame rail along with the other dimensions, and variables in the main MATLAB code, and re-running it again, the results were as below:

Design Variables	
Battery capacity in [kWh]	40
Choice of electric motor & inverter pair	Motor and inverter A
Motor and driveline configuration	RWD2
Final drive ratio	2
Frame rail (w, h, t) [mm]	(8, 16, 1)
Suspension spring constant in [N/m]	Front = 14,404 | Rear = 22,355
Suspension damping coefficient in [Ns/m]	Front = 1,415 | Rear = 1,830
Performance Variables	
Profit [$]	1.9841e+09
Demand	95,084
Price [$]	64,760
Cost [$]	44,419
0-100 km/h acceleration time [s]	6.1500
Top Speed [km/h]	341.98
Range [km]	162.62
Max Frame Rail Stress [MPa]	207.79
Max Frame Rail Deflection [mm]	0.0146
Ability to ascend a 12% slope at 30 km/h	[Pass]
Ability to turn on a 100m diameter at up to 0.8g	[Pass]
Kus value	0.1541
Max lateral acceleration tested [m/s^2]	9.9976
Turning Diameter [m]	99.947
Front ride frequency [Hz]	1.2

Rear ride frequency [Hz]	1.44
CG_a / Wheelbase length ratio	0.5187

2D Layout

RWD (2 motors A)



Cargo volume 0.738 m^3:

- 610 mm x 440 mm x 1550 mm
- 305 mm x 250 mm x 1550 mm
- 560 mm x 280 mm x 1300 mm

Battery volume 0.0667 m^3 (66.7 liters)

- 200 mm x 320 mm x 1042 mm

The minimum distance between battery and the passenger is 380 mm.

Section

4

APPENDIX

THE MATLAB CODE AND THE OUTPUT ITERATIONS

The Iterations results

The Attached Excel sheet includes the 1799 iterations data in detail. File name is “All_iteration_results”.

#	configu	motorin	battery	drive_r	profit	cost	acceler	max_sp	range_	slope_r	stress	deflecti	Kus	lateral_	diamet	turning	spring_	spring_	spring_	spring_
1	1	2	120000	10	$ 2,953,965,747	$ 42,741	1	80.0434	811.5231	1	158.7597	0.011241	0.191738	17.35745	100.1972	1	16727.6	23354.67	1643.383	1912.043
2	3	2	120000	10	$ 2,953,950,510	$ 42,741	1	80.04209	811.5231	1	168.223	0.011241	0.130252	34.39727	100.0906	1	14540.34	26504.32	1428.498	2169.904
3	3	2	110000	10	$ 2,870,125,285	$ 41,291	1	80.0547	753.5593	1	166.7478	0.010927	0.124434	35.76009	99.59868	1	14038.25	25901.16	1379.171	2120.524
4	1	2	110000	10	$ 2,870,124,432	$ 41,291	1	80.05462	753.5593	1	157.2845	0.010927	0.186174	17.67844	99.77792	1	16225.51	22751.5	1594.056	1862.662
5	1	2	100000	10	$ 2,775,884,054	$ 39,841	1	80.06794	692.8746	1	155.8093	0.010612	0.180602	17.99651	99.27813	1	15723.42	22148.34	1544.729	1813.281
6	3	2	100000	10	$ 2,775,883,578	$ 39,841	1	80.0679	692.8746	1	165.2726	0.010612	0.118593	37.83944	100.1851	1	13536.16	25298	1329.844	2071.143
7	1	2	90000	10	$ 2,675,613,189	$ 38,391	1	80.08115	630.5975	1	154.3341	0.010298	0.175021	18.79585	100.2486	1	15221.33	21545.18	1495.401	1763.9
8	3	2	90000	10	$ 2,675,567,297	$ 38,391	1	80.07719	630.5975	1	163.7975	0.010298	0.112726	39.27485	99.22576	1	13034.07	24694.84	1280.516	2021.762
9	3	2	80000	10	$ 2,571,586,107	$ 36,941	1	80.09502	567.3077	1	162.3223	0.009984	0.106831	41.45344	99.24985	1	12531.98	24091.67	1231.189	1972.381
10	1	2	80000	10	$ 2,571,533,744	$ 36,941	1	80.09049	567.3077	1	152.8589	0.009984	0.169429	19.10928	99.54851	1	14719.24	20942.02	1446.074	1714.519
11	3	2	70000	10	$ 2,461,077,875	$ 35,491	1	80.10569	502.3006	1	160.8471	0.00967	0.100905	44.47828	100.1395	1	12029.89	23488.51	1181.862	1923
12	1	2	70000	10	$ 2,461,076,605	$ 35,491	1	80.10558	502.3006	1	151.3837	0.00967	0.163827	19.90973	100.2628	1	14217.15	20338.85	1396.747	1665.139
13	3	1	120000	9	$ 2,404,466,016	$ 48,541	1	77.02983	823.0434	1	172.428	0.011869	0.119204	40.52045	99.30861	1	14733.69	28878.25	1447.493	2364.257
14	2	2	120000	10	$ 2,387,389,732	$ 47,941	1	80.52834	793.1006	1	155.586	0.011689	0.220416	14.70408	99.71843	1	18185.78	23144.69	1786.64	1894.852
15	4	2	120000	10	$ 2,353,508,308	$ 47,941	1	80.2421	785.1316	1	168.223	0.011689	0.163644	24.56629	100.1966	1	16155.02	26068.97	1587.131	2134.262
16	1	2	60000	10	$ 2,347,049,052	$ 34,041	1	80.12353	436.3038	1	149.9085	0.009356	0.158212	20.21787	99.34752	1	13715.06	19735.69	1347.42	1615.758
17	3	2	60000	10	$ 2,347,049,045	$ 34,041	1	80.12353	436.3038	1	159.3719	0.009356	0.094946	46.87323	99.45302	1	11527.8	22885.35	1132.535	1873.62
18	3	1	120000	10	$ 2,345,937,764	$ 48,541	1	69.43446	830.5814	1	172.428	0.011869	0.119204	40.52045	99.30861	1	14733.69	28878.25	1447.493	2364.257
19	3	2	120000	9	$ 2,298,881,748	$ 42,741	1	88.82302	615.1451	1	168.223	0.011241	0.130252	34.39727	100.0906	1	14540.34	26504.32	1428.498	2169.904
20	1	2	120000	9	$ 2,298,881,745	$ 42,741	1	88.82302	615.1451	1	158.7597	0.011241	0.191738	17.35745	100.1972	1	16727.6	23354.67	1643.383	1912.043
21	3	2	110000	9	$ 2,268,731,284	$ 41,291	1	88.83358	570.9958	1	166.7478	0.010927	0.124434	35.76009	99.59868	1	14038.25	25901.16	1379.171	2120.524
22	1	2	110000	9	$ 2,268,728,162	$ 41,291	1	88.83331	570.9958	1	157.2845	0.010927	0.186174	17.67844	99.77792	1	16225.51	22751.5	1594.056	1862.662
23	3	1	110000	9	$ 2,258,454,405	$ 47,091	1	77.02795	748.73	1	170.9528	0.011555	0.113343	42.89807	99.66922	1	14231.6	28275.08	1398.166	2314.876
24	2	2	110000	10	$ 2,253,199,058	$ 46,491	1	80.25048	722.6903	1	154.1108	0.011374	0.214943	15.05253	99.71918	1	17683.69	22541.53	1737.312	1845.471
25	4	2	110000	10	$ 2,234,571,330	$ 46,491	1	80.2392	717.8597	1	166.7478	0.011374	0.157953	25.04732	99.24675	1	15652.93	25465.81	1537.804	2084.881
26	1	2	100000	9	$ 2,234,553,982	$ 39,841	1	88.84461	525.7889	1	155.8093	0.010612	0.180602	17.99651	99.27813	1	15723.42	22148.34	1544.729	1813.281
27	3	2	100000	9	$ 2,234,553,982	$ 39,841	1	88.84461	525.7889	1	165.2726	0.010612	0.118593	37.83944	100.1851	1	13536.16	25298	1329.844	2071.143
28	1	1	120000	9	$ 2,228,292,437	$ 48,541	1	76.96979	777.2778	1	154.559	0.011869	0.231607	14.51093	99.60113	1	18762.85	23076.24	1843.334	1889.248
29	3	2	50000	10	$ 2,226,288,264	$ 32,591	1	80.14158	368.4889	1	157.8967	0.009041	0.088948	50.11713	99.50876	1	11025.71	22282.19	1083.208	1824.239
30	1	2	50000	10	$ 2,226,287,866	$ 32,591	1	80.14154	368.4889	1	148.4334	0.009041	0.152583	21.01864	99.78326	1	13212.97	19132.53	1298.093	1566.377
31	3	1	110000	10	$ 2,201,351,497	$ 47,091	1	69.4457	756.5608	1	170.9528	0.011555	0.113343	42.89807	99.66922	1	14231.6	28275.08	1398.166	2314.876
32	1	2	90000	9	$ 2,193,573,616	$ 38,391	1	88.8549	478.8002	1	154.3341	0.010298	0.175021	18.79585	100.2486	1	15221.33	21545.18	1495.401	1763.9
33	3	2	90000	9	$ 2,193,573,616	$ 38,391	1	88.8549	478.8002	1	163.7975	0.010298	0.112726	39.27485	99.22576	1	13034.07	24694.84	1280.516	2021.762
34	1	1	120000	10	$ 2,164,877,298	$ 48,541	1	69.37679	783.4812	1	154.559	0.011869	0.231607	14.51093	99.60113	1	18762.85	23076.24	1843.334	1889.248
35	1	2	80000	9	$ 2,154,429,941	$ 36,941	1	88.86733	432.2768	1	152.8589	0.009984	0.169429	19.10928	99.54851	1	14719.24	20942.02	1446.074	1714.519
36	3	2	80000	9	$ 2,154,429,941	$ 36,941	1	88.86733	432.2768	1	162.3223	0.009984	0.106831	41.45344	99.24985	1	12531.98	24091.67	1231.189	1972.381
37	2	2	100000	10	$ 2,122,472,138	$ 45,041	1	80.24282	652.2873	1	152.6356	0.01106	0.209467	15.41789	99.65274	1	17181.6	21938.37	1687.985	1796.09
38	3	1	100000	9	$ 2,117,833,132	$ 45,641	1	77.10485	675.4868	1	169.4776	0.011241	0.107458	45.34606	99.73014	1	13729.51	27671.92	1348.839	2265.495
39	4	2	100000	10	$ 2,110,090,088	$ 45,041	1	80.5522	648.1174	1	165.2726	0.01106	0.152246	26.10845	99.64528	1	15150.85	24862.65	1488.476	2035.501
40	1	2	70000	9	$ 2,108,428,416	$ 35,491	1	88.88081	383.9464	1	151.3837	0.00967	0.163827	19.90973	100.2628	1	14217.15	20338.85	1396.747	1665.139
41	3	2	70000	9	$ 2,108,428,416	$ 35,491	1	88.88081	383.9464	1	160.8471	0.00967	0.100905	44.47828	100.1395	1	12029.89	23488.51	1181.862	1923
42	1	2	40000	10	$ 2,094,128,394	$ 31,141	1	80.15826	297.619	1	146.9582	0.008727	0.14694	21.82147	100.0588	1	12710.88	18529.37	1248.765	1516.996
43	3	2	40000	10	$ 2,094,122,423	$ 31,141	1	80.15774	297.619	1	156.4215	0.008727	0.08291	53.48246	99.07875	1	10523.62	21679.02	1033.88	1774.858
44	1	1	110000	9	$ 2,083,043,857	$ 47,091	1	76.969	703.0329	1	153.0838	0.011555	0.226165	14.85018	99.74093	1	18260.77	22473.08	1794.007	1839.867
45	3	1	100000	10	$ 2,060,140,052	$ 45,641	1	69.45101	683.3345	1	169.4776	0.011241	0.107458	45.34606	99.73014	1	13729.51	27671.92	1348.839	2265.495
46	1	2	60000	9	$ 2,053,001,575	$ 34,041	1	88.89452	333.1326	1	149.9085	0.009356	0.158212	20.21787	99.34752	1	13715.06	19735.69	1347.42	1615.758
47	3	2	60000	9	$ 2,053,001,532	$ 34,041	1	88.89452	333.1326	1	159.3719	0.009356	0.094946	46.87323	99.45302	1	11527.8	22885.35	1132.535	1873.62
48	1	1	110000	10	$ 2,020,994,057	$ 47,091	1	69.37633	709.5424	1	153.0838	0.011555	0.226165	14.85018	99.74093	1	18260.77	22473.08	1794.007	1839.867
49	1	2	110000	8	$ 2,013,047,313	$ 41,291	1	99.79914	471.1889	1	157.2845	0.010927	0.186174	17.67844	99.77792	1	16225.51	22751.5	1594.056	1862.662
50	3	2	110000	8	$ 2,013,047,313	$ 41,291	1	99.79914	471.1889	1	166.7478	0.010927	0.124434	35.76009	99.59868	1	14038.25	25901.16	1379.171	2120.524
51	1	2	100000	8	$ 2,012,825,740	$ 39,841	1	99.80591	434.8604	1	155.8093	0.010612	0.180602	17.99651	99.27813	1	15723.42	22148.34	1544.729	1813.281
52	3	2	100000	8	$ 2,012,825,690	$ 39,841	1	99.8059	434.8604	1	165.2726	0.010612	0.118593	37.83944	100.1851	1	13536.16	25298	1329.844	2071.143
53	1	2	120000	8	$ 2,007,155,659	$ 42,741	1	99.80852	505.8645	1	158.7597	0.011241	0.191738	17.35745	100.1972	1	16727.6	23354.67	1643.383	1912.043
54	3	2	120000	8	$ 2,007,155,659	$ 42,741	1	99.80852	505.8645	1	168.223	0.011241	0.130252	34.39727	100.0906	1	14540.34	26504.32	1428.498	2169.904
55	1	2	90000	8	$ 2,005,785,483	$ 38,391	1	99.8174	396.7283	1	154.3341	0.010298	0.175021	18.79585	100.2486	1	15221.33	21545.18	1495.401	1763.9
56	3	2	90000	8	$ 2,005,785,483	$ 38,391	1	99.8174	396.7283	1	163.7975	0.010298	0.112726	39.27485	99.22576	1	13034.07	24694.84	1280.516	2021.762
57	3	2	80000	8	$ 1,996,548,809	$ 36,941	1	99.82958	358.0172	1	162.3223	0.009984	0.106831	41.45344	99.24985	1	12531.98	24091.67	1231.189	1972.381
58	1	2	80000	8	$ 1,996,546,122	$ 36,941	1	99.82934	358.0172	1	152.8589	0.009984	0.169429	19.10928	99.54851	1	14719.24	20942.02	1446.074	1714.519
59	1	2	50000	9	$ 1,994,712,875	$ 32,591	1	88.90589	281.5562	1	148.4334	0.009041	0.152583	21.01864	99.78326	1	13212.97	19132.53	1298.093	1566.377
60	3	2	50000	9	$ 1,994,630,350	$ 32,591	1	88.89867	281.5562	1	157.8967	0.009041	0.088948	50.11713	99.50876	1	11025.71	22282.19	1083.208	1824.239
61	2	2	90000	10	$ 1,988,521,093	$ 43,591	1	80.53719	580.0367	1	151.1604	0.010746	0.203988	15.80143	99.51729	1	16679.51	21335.2	1638.658	1746.709
62	4	2	90000	10	$ 1,986,506,152	$ 43,591	1	80.53609	579.5109	1	163.7975	0.010746	0.146523	27.18429	99.87973	1	14648.76	24259.48	1439.149	1986.12
63	3	2	70000	8	$ 1,981,275,680	$ 35,491	1	99.84241	317.7184	1	160.8471	0.00967	0.100905	44.47828	100.1395	1	12029.89	23488.51	1181.862	1923
64	1	2	70000	8	$ 1,981,275,676	$ 35,491	1	99.84241	317.7184	1	151.3837	0.00967	0.163827	19.90973	100.2628	1	14217.15	20338.85	1396.747	1665.139
65	3	1	90000	9	$ 1,977,585,305	$ 44,191	1	76.89045	603.1051	1	168.0024	0.010927	0.101547	47.86894	99.47133	1	13227.42	27068.76	1299.512	2216.115
66	5	2	120000	10	$ 1,969,814,986	$ 51,941	1	80.50254	783.9778	1	160.3217	0.011987	0.173406	22.54431	99.61298	1	16788.84	26416.13	1649.4	2162.684
67	4	1	40000	2	$ 1,969,301,800	$ 44,441	6	341.9971	161.8594	1	160.6265	0.010298	0.143549	27.00246	99.87711	1	14116.31	23136.41	1386.84	1894.174
68	1	2	30000	10	$ 1,961,033,826	$ 29,691	1	80.17595	226.4006	1	145.483	0.008413	0.14128	22.62626	100.1655	1	12208.79	17926.2	1199.438	1467.615
69	3	2	30000	10	$ 1,961,032,984	$ 29,691	1	80.1759	226.4005	1	154.9463	0.008413	0.076825	57.79704	99.16077	1	10021.53	21075.86	984.5533	1725.477
70	4	1	30000	2	$ 1,958,799,883	$ 42,991	6	341.9973	122.8496	1	159.1513	0.009984	0.137804	28.06537	99.84957	1	13614.22	22533.25	1337.512	1844.793
71	1	2	60000	8	$ 1,958,635,781	$ 34,041	1	99.85591	275.4806	1	149.9085	0.009356	0.158212	20.21787	99.34752	1	13715.06	19735.69	1347.42	1615.758
72	3	2	60000	8	$ 1,958,635,781	$ 34,041	1	99.85591	275.4806	1	159.3719	0.009356	0.094946	46.87323	99.45302	1	11527.8	22885.35	1132.535	1873.62
73	4	1	50000	2	$ 1,947,003,479	$ 45,891	6.1	341.9961	199.2531	1	162.1017	0.010612	0.149275	25.9539	99.72764	1	14618.4	23739.58	1436.167	1943.555
74	1	1	100000	9	$ 1,944,049,385	$ 45,641	1	76.96735	630.3128	1	151.6087	0.011241	0.220723	15.20567	99.818	1	17758.68	21869.92	1744.68	1790.487
75	3	2	50000	8	$ 1,933,930,479	$ 32,591	1	99.8663	232.7058	1	157.8967	0.009041	0.088948	50.11713	99.50876	1	11025.71	22282.19	1083.208	1824.239
76	1	2	50000	8	$ 1,933,929,956	$ 32,591	1	99.86625	232.7058	1	148.4334	0.009041	0.152583	21.01864	99.78326	1	13212.97	19132.53	1298.093	1566.377
77	3	2	40000	9	$ 1,931,385,338	$ 31,141	1	88.91702	228.6344	1	156.4215	0.008727	0.08291	53.48246	99.07875	1	10523.62	21679.02	1033.88	1774.858

The MATLAB Code

The below is the main code (used for task 2), task 3 code just includes add sections for optimization and iteration, and you can find it in the uploaded folder also.

Group A

Initialization

```
clear all
close all
```

Design Variables

```
battery_capacity=40000;              %battery capcity in Wh
motorinverter_select='A';            %A, B, or C
configuration='RWD2';                %FWD1, FWD2, RWD1, RWD2, AWD2, or AWD4
```

```
drive_ratio=2;                      %final drive ratio
w=0.080;                            %(m) frame rail width
h=0.160;                            %(m) frame rail height
t=0.001;                            %(m) frame rail thickness
frame_rail_area=(w*h)-((w-(2*t))*(h-(2*t)));

if ((configuration =='FWD1') | (configuration =='RWD1'))
    num_motor=1;
    num_gearbox=1;                  %two motors could be connected to the samce GB to double the torque
    num_inverter=1;
    num_differential=1;
elseif ((configuration =='FWD2') | (configuration =='RWD2'))
    num_motor=2;
    num_gearbox=1;                  %two motors could be connected to the samce GB to double the torque
    num_inverter=2;
    num_differential=1;
elseif (configuration =='AWD2')
    num_motor=2;
    num_gearbox=2;
    num_inverter=2;
    num_differential=2;
elseif configuration =='AWD4'
    num_motor=4;
    num_gearbox=2;
    num_inverter=4;
    num_differential=2;
end
```

Inputs

```
slope=0;                            %the road slope in %
```

Parameters

```
g=9.81;                             %Gravitational acceleration  = 9.81 /2
Cd=0.27;                            %Drag coefficient Cd = 0.27
r=659.7/1000/2;                     %for Tire size = 225/4518, the diameter = 659.7 mm
rho=1.26;                           %Density of air  = 1.26 /3
steel_density= 7800;                %kg/m^3

frf=1.2;                            %Front ride requency (Hz)
frr=1.44;                           %Front ride requency (Hz)
damping_ratio=0.3;                  %damping ratio

% Dimensions
battery_width=1.25;                 %max battery width
W103=1820/1000;                     %vehicle width (m)
W101=1550/1000;                     %Tread_front
L101=2700/1000;                     %the wheel base length

if motorinverter_select=='A'        %%motor, inverter, and gearbox paratmers
    motor_mass=85;                  %kg
    inverter_mass=35;               %kg
    motorinverter_cost=7500;        %dollars
```

```
    motorinverter_power=175000;     %W; continuous
    gearbox_mass=45;                %high torque gearbox is needed based on the map
    gearbox_cost=7500;              %high torque gearbox is needed based on the map
    motor_cylinder_dia = 0.390;
    inverter_box_h = 0.422;
    gearbox_h = 0.3;
    load MotorA_Data.mat            %to load motor A map
elseif motorinverter_select=='B'
    motor_mass=41;                  %kg
    inverter_mass=16;               %kg
    motorinverter_cost=5200;        %dollars
    motorinverter_power=70000;      %W; continuous
    gearbox_mass=28;                %low torque gearbox is needed based on the map
    gearbox_cost=4000;              %low torque gearbox is needed based on the map
    motor_cylinder_dia = 0.276;
    inverter_box_h = 0.393;
    gearbox_h = 0.25;
    load MotorB_Data.mat            %to load motor B map
elseif motorinverter_select=='C'
    motor_mass=55;                  %kg
    inverter_mass=12;               %kg
    motorinverter_cost=5500;        %dollars
    motorinverter_power=100000;     %W; continuous
    gearbox_mass=45;                %high torque gearbox is needed based on the map
    gearbox_cost=7500;              %high torque gearbox is needed based on the map
    motor_cylinder_dia = 0.28;
    inverter_box_h = 0.2;
    gearbox_h = 0.3;
    load MotorC_Data.mat            %to load motor C map
end

%Masses
passenger_mass=101;             %kg/passenger; left per passenger to be used in the CG calculations
later
cargo_mass=28;                  %kg (4*7)
info_mass=15;                   %kg
sunroof_mass=10;                %kg
unsprung_mass=60*4;             %kg
total_gearbox_mass=gearbox_mass*num_gearbox;            %kg (based on the selection)
differential_mass=10;                                    %kg
total_differential_mass= num_differential*differential_mass;
axle_mass=20;                                           %5kg*4halfs
battery_mass=battery_capacity/250;                              %battery capacity (Wh)/250(Wh/kg) =
battery mass (kg)
total_motorinverter_mass=num_motor*(motor_mass+inverter_mass); %kg (based on the selection)

%Volumes
battery_volume=(battery_capacity/600/1000);             %m^3
battery_crosssection_area=battery_volume/battery_width;

%Costs
misc_electronics_cost=3000;
wheel_assembly_cost=450*4;
differntial_cost=600;
axles_cost=350;
```

Assumptions

```
E=2.15*(10^11);                   %Young's Modulus Pa
G=78000;                          %Shear Modulus N/mm2

Cr=0.009;                         %Rolling coefficient Cr=0.009
mu=0.9;                           %Coefficient of friction (dry asphalt) = 0.9
BrkGain = 1e4;                    %Max Brake effort gain [N]
eff_dl =0.9;                      %Driveline efficiency [unitless]

useful_battery_capacity=0.95;     %the useful capacity of the battery is 90% of the capacity

Jz = 2380;                        % Moment of inertia of the vehicle [kgm^2]
Caf = 1290;                       % Tire cornering stiffness front [N/deg]
Car = 3000;                       % Tire cornering stiffness rear [N/deg]
Cmzf = 30;                        % Tire aligning stiffness front [Nm/deg]
Cmzr = 20;                        % Tire aligning stiffness rear [Nm/deg]

MR=0.9;                             % Motion Ratio
```

Simulink models Paramters

```
time_step = 0.05;
```

2D Model inputs

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LOADS IN FRAME RAIL SECTION NEEDS TO BE UPDATED MANUALLY BASED ON THE
% DESIGN %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%the below values to be entered from the 2D model and are required for
%calculations, they are just assumed now to run the model
L108=3944/1000;                       %The total length of the vehicle
H101=1420/1000;                       %The total vehicle height
battery_thickness=0.05;               %battery thickness
L104 = 0.622;                         %front overhang
H156 = 0.140;                         % ground clearance
H5_1 = 0.500;                         % front passenger H-Point
H5_2 = 0.550;                         % rear passenger H-Point

% X loads locations // needed for the frame rail calculations later
cargofront_X_dis = 0.28;              % Cargo          (front)
motorf_X_dis = 0.4;                   % Motor          (front)
inverterf_X_dis = 0.4;                % Inverter       (front)
gearboxf_X_dis = 0.4;                 % Gearbox        (front)
differentialf_X_dis = 0.4;            % Differential   (front)
frontaxle_X_dis = 0.622;              % Axle           (front)
batteryfront_X_dis = 0.807;           % Battery        (front)
infotainment_X_dis = 0.9;             % Infortainment
frontpass_X_dis = 1.3+0.622;          % Passenger      (front)
sunroof_X_dis = 2.0;                  % Sunroof
body_X_dis = L108/2;                  % Body
```

```
rearpass_X_dis = 2.04+0.622;            % Passenger     (rear)
motorr_X_dis = 0.622+2.7-0.3;           % Motor         (rear)
inverterr_X_dis = motorr_X_dis;         % Inverter      (rear)
gearboxr_X_dis = motorr_X_dis+(0.25/2); % Gearbox       (rear)
differentialr_X_dis = gearboxr_X_dis;   % Differential  (rear)
rearaxle_X_dis = 0.622+2.7;             % Axle          (rear)
batteryrear_X_dis = 3.4;                % Battery       (rear)
cargorear_X_dis = 3.944-0.305;          % Cargo         (rear)
framerail_X_dis_start = 0;              % Frame Rail
framerail_X_dis_end = L108;             % Frame Rail

% Y loads locations // needed for the frame rail calculations later
cargofront_Y_dis = H156 + 0.436 + 0.225/2 ;         % Cargo         (front)
motorf_Y_dis = H156 + motor_cylinder_dia/2 ;        % Motor         (front)
inverterf_Y_dis = H156 + inverter_box_h/2 ;         % Inverter      (front)
gearboxf_Y_dis = H156 + gearbox_h/2 ;               % Gearbox       (front)
differentialf_Y_dis = H156 + r ;                    % Differential  (front)
frontaxle_Y_dis = H156 + r ;                        % Axle          (front)
batteryfront_Y_dis = H156 + battery_thickness/2 ;   % Battery       (front)
infotainment_Y_dis = H5_1;                          % Infortainment
frontpass_Y_dis = H5_1;                             % Passenger     (front)
sunroof_Y_dis = H101;                               % Sunroof
body_Y_dis = H101/3;                                % Body
rearpass_Y_dis = H5_2;                              % Passenger     (rear)
motorr_Y_dis =  H156 + motor_cylinder_dia/2 ;       % Motor         (rear)
inverterr_Y_dis = H156 + inverter_box_h/2 ;         % Inverter      (rear)
gearboxr_Y_dis = H156 + gearbox_h/2 ;               % Gearbox       (rear)
differentialr_Y_dis = H156 + r;                     % Differential  (rear)
rearaxle_Y_dis = H156 + r ;                         % Axle          (rear)
batteryrear_Y_dis = H156 + battery_thickness/2 ;    % Battery       (rear)
cargorear_Y_dis = H156 + 0.436 + 0.225/2;           % Cargo         (rear)
framerail_Y_dis = H156 + h/2;                       % Frame Rail
```

Frame Rail optimization - intialization (comment out after finishing it)

% before activating add ; to stress and the deflection, and deactivvate % plots in SFBM frame_rail_optimization=zeros(0,6); %w,aspect ratio,t,area,stress, deflection aspect_ratio=1; i=0; for w=0.04:0.005:0.15 for aspect_ratio=1:0.5:5 h=w*aspect_ratio; for t=0.001:0.001:0.015 i=i+1; frame_rail_area=(w*h)-((w-(2*t))*(h-(2*t)));

Vehicle Mass Calculations

SI units have been used all through out the simulation unless otherwise stated

```
frame_rail_mass=3*2*frame_rail_area*L108*7800;          %kg (based on the frame rail design and 2D
inputs
body_mass=L108*H101*50;                                 %kg
mass=(4*passenger_mass)+cargo_mass+info_mass+sunroof_mass+unsprung_mass+total_gearbox_mass+total_differe
ntial_mass+axle_mass+body_mass+frame_rail_mass+battery_mass+total_motorinverter_mass;
```

Frame Rail calculations

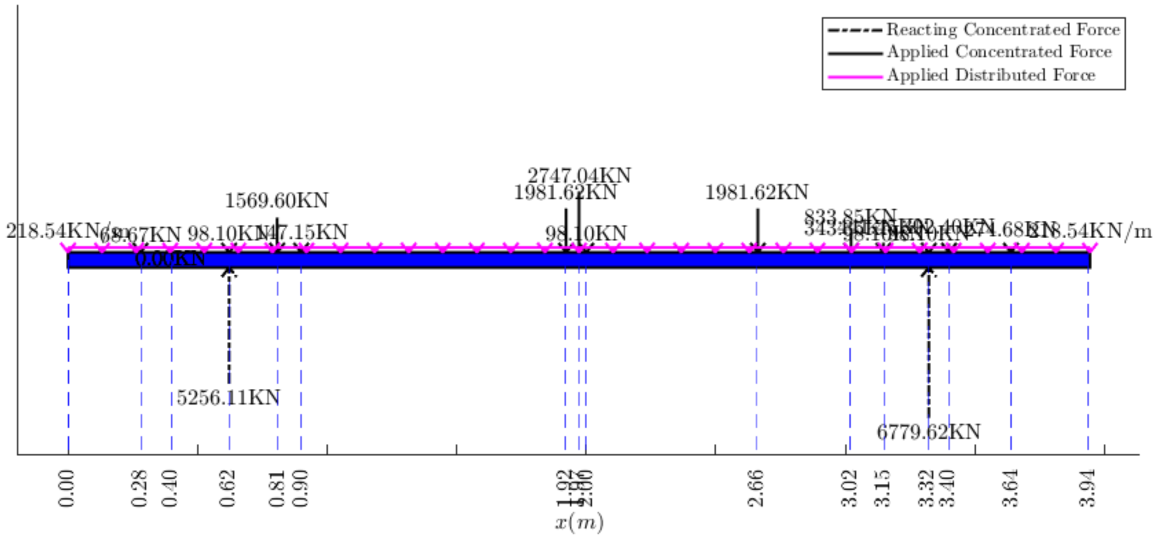
as per the assumptions in the report the below analysis was performed

```
% calculating the different loads (all of them in N)
load_cargof= -cargo_mass*g/4;
    if ((configuration =='RWD1') | (configuration =='RWD2'))
     load_motorf=0;
     load_inverterf=0;
     load_gearboxf=0;
     load_differentialf=0;
     elseif ((configuration =='FWD2') | (configuration =='AWD4'))
     load_motorf= -2*motor_mass*g;
     load_inverterf= -2*inverter_mass*g;
     load_gearboxf= -1*gearbox_mass*g;
     load_differentialf= -1*differential_mass*g;
     elseif ((configuration =='FWD1') | (configuration =='AWD2'))
     load_motorf= -1*motor_mass*g;
     load_inverterf= -1*inverter_mass*g;
     load_gearboxf= -1*gearbox_mass*g;
     load_differentialf= -1*differential_mass*g;
     end
load_axlef= -axle_mass*g/2;
load_batteryf= -battery_mass*g;
load_infotainment= -info_mass*g;
load_passf= -2*passenger_mass*g;
load_sunroof= -sunroof_mass*g;
load_body=-body_mass*g;
load_passr= -2*passenger_mass*g;
     if ((configuration =='RWD1') | (configuration =='AWD2'))
     load_motorr= -1*motor_mass*g;
     load_inverterr= -1*inverter_mass*g;
     load_gearboxr= -1*gearbox_mass*g;
     load_differentialr= -1*differential_mass*g;
     elseif ((configuration =='FWD2') | (configuration =='FWD1'))
     load_motorr=0;
     load_inverterr=0;
     load_gearboxr=0;
     load_differentialr=0;
     elseif ((configuration =='RWD2') | (configuration =='AWD4'))
     load_motorr= -2*motor_mass*g;
     load_inverterr= -2*inverter_mass*g;
     load_gearboxr= -1*gearbox_mass*g;
     load_differentialr= -1*differential_mass*g;
     end
load_axler= -axle_mass*g/2;
load_batteryr= -battery_mass*g/4;
load_cargor= -cargo_mass*g;
load_framerail= -frame_rail_mass*g/L108;


% The SFBM function
Name1 = 'Frame Rail';
LengthSupport1 = [L108,L104,(L101+L104)];       % Length and Supports
% Concetrated Loads

F1 = {'CF',load_cargof,cargofront_X_dis};
F2 = {'CF',load_motorf,motorf_X_dis};
F3 = {'CF',load_inverterf,inverterf_X_dis};
```

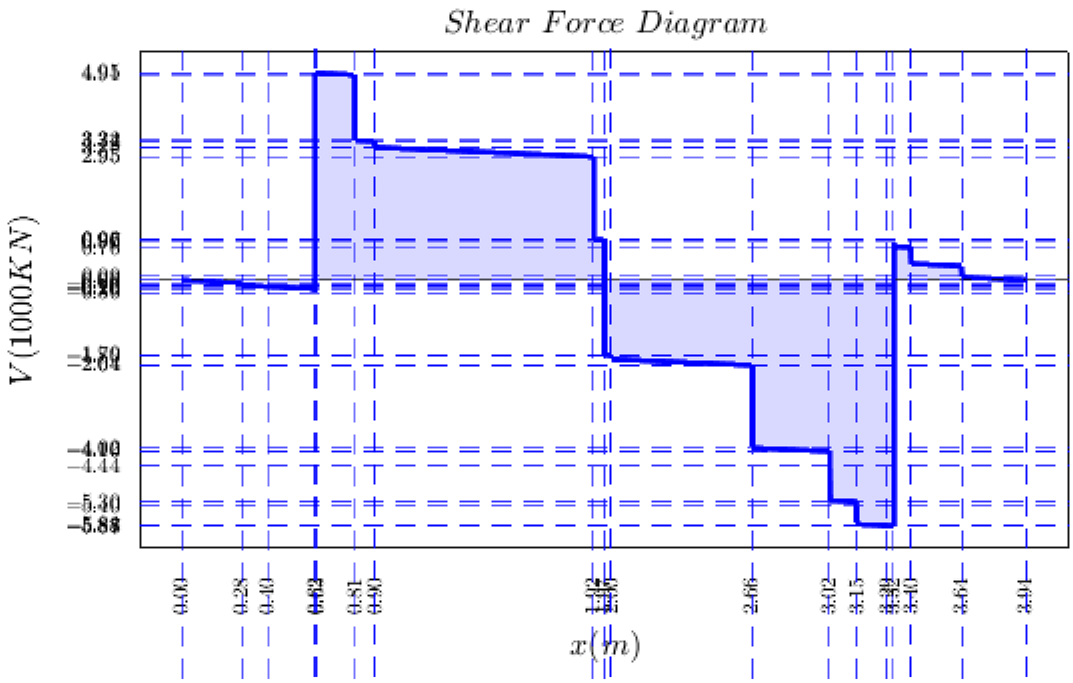
```
F4 = {'CF',load_gearboxf,gearboxf_X_dis};
F5 = {'CF',load_differentialf,differentialf_X_dis};
F6 = {'CF',load_axlef,frontaxle_X_dis};
F7 = {'CF',load_batteryf,batteryfront_X_dis};
F8 = {'CF',load_infotainment,infotainment_X_dis};
F9 = {'CF',load_passf,frontpass_X_dis};
F10 = {'CF',load_sunroof,sunroof_X_dis};
F11 = {'CF',load_body,body_X_dis};
F12 = {'CF',load_passr,rearpass_X_dis};
F13 = {'CF',load_motorr,motorr_X_dis};
F14 = {'CF',load_inverterr,inverterr_X_dis};
F15 = {'CF',load_gearboxr,gearboxr_X_dis};
F16 = {'CF',load_differentialr,differentialr_X_dis};
F17 = {'CF',load_axler,rearaxle_X_dis};
F18 = {'CF',load_batteryr,batteryrear_X_dis};
F19 = {'CF',load_cargor,cargorear_X_dis};

% Distributed Loads
D1 = {'DF',load_framerail,[framerail_X_dis_start,framerail_X_dis_end]};

% Call the function - output in the max bending mometn - N.m
bending_moment=SFBM(Name1,LengthSupport1,F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,F11,F12,F13,F14,F15,F16,F17,F18,
F19,D1);
bending_moment_max=max(abs(bending_moment));                            %max bending moment

% Calculating the max stress on the beam (the hieghest moment at the most
% far line of the section (the biggest stresses)
second_moment_area=((w*(h^3))/12)-(((w-2*t)*((h-2*t)^3))/12);
stress=(bending_moment_max*(h/2))/(second_moment_area*1000000)          %MPa

% Calculating the deflection
% rough assumption of having the vehicle weight affecting at the center
% distance betweeen the two wheels (the most sever condition)
% two supporters and one concentrated load at the middle equals (m.g)

deflection_max=(mass-unsprung_mass)*g*(L101^3)/(48*E*second_moment_area)
```

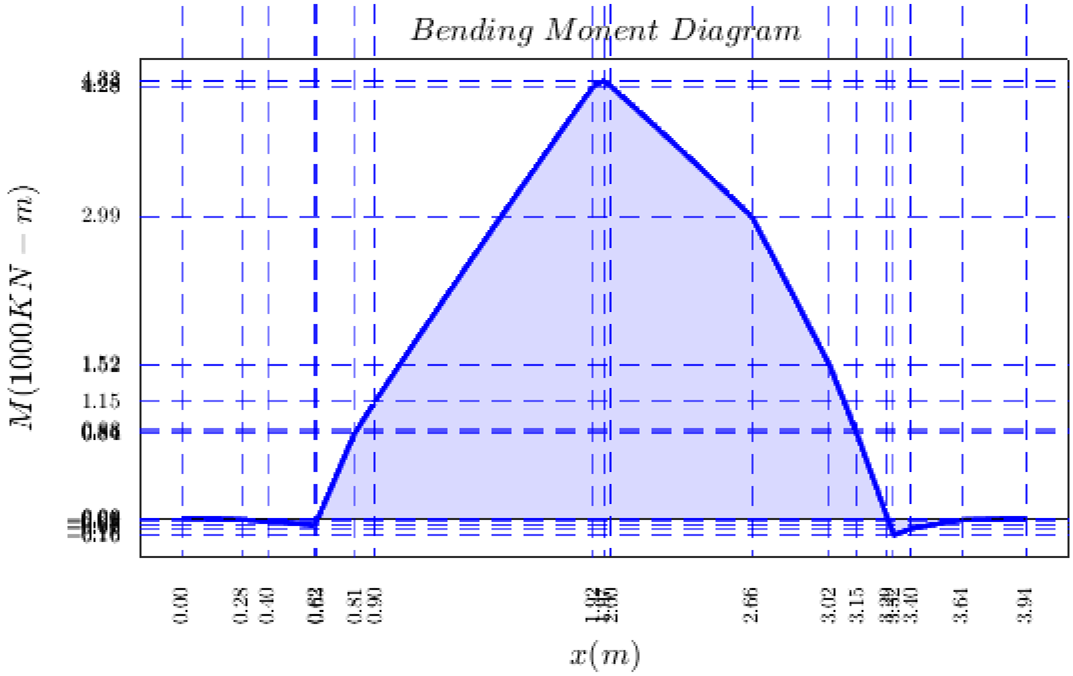
```
stress =

  207.7919


deflection_max =

    0.0146
```

Free Body Diagram



Range	*Equations of Shear Force :*	*Equations of Bending Moment :*
0 *to* 0.28	$-218.535x$	$-109.268x^2$
0.28 *to* 0.4	$-218.535x - 68.67$	$-109.268x^2 - 68.67x + 18.957$
0.4 *to* 0.622	$-218.535x - 68.67$	$-109.268x^2 - 68.67x + 18.956$
0.622 *to* 0.807	$-218.535x + 5089.341$	$-109.268x^2 + 5089.341x - 3185.292$
0.807 *to* 0.9	$-218.535x + 3519.741$	$-109.268x^2 + 3519.741x - 1922.43$
0.9 *to* 1.922	$-218.535x + 3372.591$	$-109.268x^2 + 3372.591x - 1790.109$
1.922 *to* 1.972	$-218.535x + 1390.971$	$-109.268x^2 + 1390.971x + 2016.044$
1.972 *to* 2	$-218.535x - 1356.065$	$-109.268x^2 - 1356.065x + 7433.198$
2 *to* 2.662	$-218.535x - 1454.165$	$-109.268x^2 - 1454.165x + 7629.553$
2.662 *to* 3.022	$-218.535x - 3435.785$	$-109.268x^2 - 3435.785x + 12897.204$
3.022 *to* 3.147	$-218.535x - 4612.985$	$-109.268x^2 - 4612.985x + 16455.292$
3.147 *to* 3.322	$-218.535x - 5152.535$	$-109.268x^2 - 5152.535x + 18152.165$
3.322 *to* 3.4	$-218.535x + 1528.984$	$-109.268x^2 + 1528.984x - 4035.949$
3.4 *to* 3.639	$-218.535x + 1136.584$	$-109.268x^2 + 1136.584x - 2701.122$





Frame Rail optimization - end (comment out after finishing it)

%w,aspect ratio,t,area,stress, deflection frame_rail_optimization(i,1)=w;
frame_rail_optimization(i,2)=aspect_ratio; frame_rail_optimization(i,3)=t;
frame_rail_optimization(i,4)=frame_rail_area; frame_rail_optimization(i,5)=stress;
frame_rail_optimization(i,6)=deflection_max;   end   end end

Center of Gravity

```
mass_x_dis = -1*(load_cargof*cargofront_X_dis ...
    + load_motorf*motorf_X_dis ...
    + load_inverterf*inverterf_X_dis ...
    + load_gearboxf*gearboxf_X_dis ...
    + load_differentialf*differentialf_X_dis ...
    + load_axlef*frontaxle_X_dis ...
    + load_batteryf*batteryfront_X_dis ...
    + load_infotainment*infotainment_X_dis ...
    + load_passf*frontpass_X_dis ...
    + load_sunroof*sunroof_X_dis ...
    + load_body*body_X_dis ...
    + load_passr*rearpass_X_dis ...
    + load_motorr*motorr_X_dis ...
    + load_inverterr*inverterr_X_dis ...
    + load_gearboxr*gearboxr_X_dis ...
    + load_differentialr*differentialr_X_dis ...
    + load_axler*rearaxle_X_dis ...
    + load_batteryr*batteryrear_X_dis ...
    + load_cargor*cargorear_X_dis ...
    +(load_framerail*L108)*(L108/2))/g;

mass_CG=mass-unsprung_mass;

CG_a = (mass_x_dis/mass_CG) - L104 ;  %L104: front overhang
CG_b = L101 - CG_a;
CG_location=CG_a/L101


mass_Y_dis = -1*(load_cargof*cargofront_Y_dis ...
    + load_motorf*motorf_Y_dis ...
    + load_inverterf*inverterf_Y_dis ...
    + load_gearboxf*gearboxf_Y_dis ...
    + load_differentialf*differentialf_Y_dis ...
    + load_axlef*frontaxle_Y_dis ...
    + load_batteryf*batteryfront_Y_dis ...
    + load_infotainment*infotainment_Y_dis ...
    + load_passf*frontpass_Y_dis ...
    + load_sunroof*sunroof_Y_dis ...
    + load_body*body_Y_dis ...
    + load_passr*rearpass_Y_dis ...
    + load_motorr*motorr_Y_dis ...
    + load_inverterr*inverterr_Y_dis ...
    + load_gearboxr*gearboxr_Y_dis ...
    + load_differentialr*differentialr_Y_dis ...
    + load_axler*rearaxle_Y_dis  ...
    + load_batteryr*batteryrear_Y_dis ...
    + load_cargor*cargorear_Y_dis ...
    + (load_framerail*L108)*framerail_Y_dis)/g;

CG_h = mass_Y_dis/mass_CG;
```

```
CG_location =
```

```
    0.5187
```

Vehicle Costing Calculations

```
body_cost=1000*W103*L108*H101;
frame_rail_cost=frame_rail_mass*2;
battery_cost=145*battery_capacity/1000;

cost=misc_electronics_cost+wheel_assembly_cost+(gearbox_cost*num_gearbox) ...
    +differntial_cost+axles_cost+(num_motor*motorinverter_cost)+body_cost ...
    +frame_rail_cost+battery_cost
```

```
cost =

   4.4419e+04
```

Vehicle Perfromance Model Setting and Parameters

```
%road load model
vehicle_frontal_area=H101*W103;

%Forces Distributer Model
front_on=0;       %0 means no forces to the front, 1 means forces are going to front
rear_on=0;        %0 means no forces to the rear, 1 means forces are going to rear
if ((configuration =='FWD1') | (configuration =='FWD2'))
    front_on=1;       %0 means no forces to the front, 1 means forces are going to front
    rear_on=0;        %0 means no forces to the rear, 1 means forces are going to rear
elseif ((configuration =='RWD2') | (configuration =='RWD1'))
    front_on=0;       %0 means no forces to the front, 1 means forces are going to front
    rear_on=1;        %0 means no forces to the rear, 1 means forces are going to rear
elseif ((configuration =='AWD4') | (configuration =='AWD2'))
    front_on=1;       %0 means no forces to the front, 1 means forces are going to front
    rear_on=1;        %0 means no forces to the rear, 1 means forces are going to rear
end

%Torque multiplication based on the number of engines per axle
torque_multi_factor=0;
if (configuration =='AWD2')
    torque_multi_factor=1;
elseif (configuration =='FWD1')
    torque_multi_factor=1;
elseif (configuration =='RWD1')
    torque_multi_factor=1;
elseif (configuration =='RWD2')
    torque_multi_factor=2;
elseif (configuration =='AWD4')
    torque_multi_factor=2;
elseif (configuration =='FWD2')
    torque_multi_factor=2;
end
```

```
%Electrical Motor Model
%Breakpoint values
omega_BP=max(size(EM_Omega_Map));
omega_max_BP=max(size(EM_Omega_Max));
torque_BP=max(size(EM_Torque_Map));
torque__max_BP=max(size(EM_Torque_Max));
%max Torqua and omega
omega_max=max(EM_Omega_Max);
torque__max=max(EM_Torque_Max);


%Drive Cycle Model
load US06_Drive_Cycle.mat
cycle(:,1)=t_cyc;
cycle(:,2)=v_cyc;
Kp = 0.5;
Ki = 0.03;
```

Max speed calculations (1 of 3 don't change the sequence)

```
sim_max=sim('Vehicle_Performance_Model_max.slx');
max_speed=max(sim_max.max_velocity_km_per_h)                  %(m/s)
```

```
max_speed =

  341.9806
```

0-100 km/h time calculations (2 of 3 don't change the sequence)

```
acceleration_time=max(sim_max.acceleration_time_to_hundred_kmh) %(s)
```

```
acceleration_time =

    6.1500
```

Range Calculations calculations (3 of 3 don't change the sequence)

```
sim_range=sim('Vehicle_Performance_Model_range.slx');
range_battery=max(sim_range.range_km)                  %(km)
```

```
range_battery =

  162.6164
```

Ascend a 12% slope speed (minimum at 30 km/h)

```
slope=0.12;                           %the road slope in %
sim_slope=sim('Vehicle_Performance_Model_slope.slx');

slope_difference_variation=sim_slope.slope_error.Data(end);

if abs(slope_difference_variation)<0.001
    slope_result='[Pass]'
else
    slope_result='[Fail]'
end
```

```
slope_result =

    '[Pass]'
```

Profit calculations

```
profit_input=[cost; range_battery; acceleration_time; max_speed];
[profit,demand,price] = profitPredict(profit_input)
```

```
profit =

   1.9341e+09


demand =

   9.5084e+04


price =

   6.4760e+04
```

Assesing the stability at 100 m radius with up to 0.8 g lateral accleration

```
Kus=(((1-CG_location)*mass)/(2*Caf))-(((CG_location)*mass)/(2*Car))

steeringScenario = 1;
steerAngle = 6;   % Ackermann steering angle in [deg]
enablePlotting = false;        % turn on the plotting features (off is faster!)
maxYawRate = 20;               % [rad/s] -- stop the simulation if yaw rate is >

%Intial values
V = 30;                      % velocity [m/s] (1 kmph is 40/3.6 m/s)
turning_data = sim('vehicle_dynamics_turning.slx',200);
Turning_data=[turning_data.X.Data turning_data.Y.Data];
[Lc,R,K] = curvature(Turning_data); % external function
```

```
radius=mean(R(50:end-1,:)); % excluding the first 50 points before the steady speed

%Determining the speed for 100 m diamter (50 m radius)
while (radius<49.5) || (50.5<radius)
    if radius<50
        V=V+0.5;
    elseif radius>50
        V=V-0.5;
    end
    turning_data = sim('vehicle_dynamics_turning.slx',200);
    Turning_data=[turning_data.X.Data turning_data.Y.Data];
    [Lc,R,K] = curvature(Turning_data); % external function
    radius=mean(R(50:end-1,:)); % excluding the first 50 points before the steady speed
end

% calculating the simulated turning radius
turning_data = sim('vehicle_dynamics_turning.slx',1000);
Turning_data=[turning_data.X.Data turning_data.Y.Data];
[Lc,R,K] = curvature(Turning_data); % external function
radius=mean(R(50:end-1,:)) % excluding the first 50 points before the steady speed

lateral_acceleration_max=max(abs(turning_data.lateral_acceleration.data))

if lateral_acceleration_max>=(g*0.8)
    turnning_result='[Pass]'
elseif lateral_acceleration_max<(g*0.8)
    turnning_result='[Adjust]'    %Increase the speed and steering angle
end

figure(1)
plot(turning_data.X.Data,turning_data.Y.Data);
grid on
title('Turning Radius')
xlabel('X')
ylabel('Y')
```

```
Kus =

    0.1541


radius =

   49.9736


lateral_acceleration_max =

    9.9976


turnning_result =
```

```
'[Pass]'
```

Spring Constant Calculations

```
front_corners_sprung_mass=(mass-unsprung_mass)*(1-CG_location)/2;
rear_corners_sprung_mass=(mass-unsprung_mass)*(CG_location)/2;

spring_rate_front=4*(pi^2)*(frf^2)*front_corners_sprung_mass*(MR^2)     % N
spring_rate_rear=4*(pi^2)*(frr^2)*rear_corners_sprung_mass*(MR^2)       % N
```

```
spring_rate_front =

   1.4404e+04


spring_rate_rear =

   2.2355e+04
```

Damping Coefficient

```
spring_damping_coeffiecient_front = 4*pi*damping_ratio*frf*front_corners_sprung_mass;   %N.s/m
spring_damping_coeffiecient_rear = 4*pi*damping_ratio*frr*rear_corners_sprung_mass;     %N.s/m
```

Published with MATLAB® R2020a