Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

University of Oviedo
Gijon Polytechnic School of Engineering, Gijon

Simulation of Hybrid and Electric Vehicles (SIMUHEV)
Course 2021/2022

Final Coursework

Simulation of BMWi3 Model

By
Pawan Kumar Dhakal
UO286375@uniovi.es

Abstract:

This coursework report is about the Multiphysics simulation of BMWi3 vehicle which is a requirement of final coursework for the SIMUHEV course 2021/2022. The simulation has three main blocks namely Cruise control block, electrical car block and the sensors block. The simulations are done using a model-based approach. The cruise control of the car is performed putting together the electrical components, mechanical components, and the corresponding control loops. The simulations are performed in MATLAB/Simulink environment and the results are presented. Finally, the conclusions are made based on the results obtained.

Contents

1.BMWi3 Parameters

The BMWi3 model parameters that are considered for simulating the same are as follows.

1.1 Battery

The BMW i3 Li-ion battery pack is composed of 8 modules in series, and each module consists of 12 cells in series [1].

Parameter	Value	Unit
Battery nominal capacity	60	Ah
Battery nominal voltage	353	V
Rated energy	22	kWh
Maximum battery current	409	A
Maximum battery power (assumption same as motor)	125	kW

1.2 DC/DC Converter

The BMW i3 data suggest that the car model is fed directly from the battery but to match the proposed topology, the following DC/DC converter parameters are considered.

Parameter	Value	Unit
DC-link voltage	600	V
Switching frequency	10	kHz
Inductor value	1	mH
Capacitor value	2375	uF
Inductor Series resistance	1.8	mOhm

1.3 Electrical Machine

The following electrical machine parameters are considered [2]. The BMWi3 eDrive technology consists of hybrid synchronous electric motor with integrated power electronics, charging unit and generator function for energy recuperation. In simulation an interior permanent magnet (IPM) machine is used.

Parameter	Value	Unit
Machine type	IPM	
Maximum torque	250	Nm
Maximum power	125	kW
Base speed @75kW	4800	RPM
Maximum speed	11400	RPM
Maximum current	400	A
Nominal voltage	400	V
Pole pairs	6	

d-axis inductance	0.0858	mH
q-axis inductance	0.24	mH
Stator series resistance (per phase)	0.005225	Ohm
PM flux	0.0663	Wb
Machine inertia	0.0666	Kgm2

1.4 DC/AC Converter

The following inverter parameters are considered for simulation.

Parameter	Value	Unit
Switching frequency	20	KHz
Modulation technique	Space vector pulse width modulation (SVPWM)	

1.5 Mechanical Parameters

The following mechanical parameters are used to simulate the vehicle body [3], [4].

Parameter	Value	Unit
Vehicle mass	1270	kg
CoG (z-axis)	0.4699	m
Wheelbase	2570	mm
Frontal area	2.38	m2
Drag coefficient	0.29	
Maximum load	1620	Kg
Air density	1.25	Kh/m3
Rolling resistance coefficient	0.013	
Wheel radius	0.19	m
Maximum speed	150	km/hr
Transmission gear ratio	5.46	

2.Cruise Control Model

To perform the cruise control, a velocity/speed reference of the car is given either directly using a step generating function or from the different speed profiles that are available in different MATLAB Simulink toolbox. The following block models are considered to perform the cruise control.

2.1 Torque Generator

The torque generator block generates the torque reference to be fed to the torque sharing block. The following mathematical equation is used to model the torque generator block.

$$T_{em} = T_{Load} + \frac{Jdw_r}{dt} + Bw_r$$

Where, T_{em} is the electromagnetic torque and T_{Load} is the load torque. J is the moment of inertia, B is the friction coefficient and w_r is the rotor speed in $\frac{rad}{s}$. Normally the friction coefficient is neglect while modelling the PI controller which generates the torque reference. Usually the given voltage/speed reference has a unit of $\frac{km}{hr}$ and to get the rotor angular speed, we can use the following formula.

$$a = r_{wheel} \times \alpha$$

Where, a is the linear acceleration in $\frac{m}{s}$, r_{wheel} is the radius of the wheel in meters and α is the angular acceleration in $\frac{rad}{s^2}$. Thus, the PI controller is designed accordingly considering all above-mentioned equation and considering the proper bandwidths to tune the controller response. Since the controller outputs the reference torque, it is then saturated between the positive and negative maximum torque. To obtain the smooth starting torque command for the next stage, the torque reference output from the PI controller is then multiplied with a ramp function.

2.2 Torque Sharing

This torque sharing block model then shares the torque from the torque generator into actual machine torque and the mechanical braking torque. This mechanical braking torque is then applied on wheels based on the brake command. The machine torque is applied to the motor control model block. To share the torque, we can monitor the input torque reference. When the brakes are applied, the torque reference is negative, thus a part of it goes to machine as regenerative torque and part of it goes to the brake pad. Eventually a function block can be programmed to share the input torque based on the battery state of charge, battery current and motor current etc.

2.3 MTPA and FW

The motor current control is performed using the maximum torque per ampere (MTPA) and field weakening (FW) techniques. MTPA control is used for the motor speed until the base speed and for the higher speeds, field weakening technique is used. Both MTPA and FW techniques are explained in brief here.

We know that the torque equation of a IPM machine is given by the following equation.

$$T_e = \frac{3p}{2}(\Psi_m i_q + (L_d - L_q) i_d i_q)$$

Where, p is the pole pairs, Ψ_m is the permanent magnet flux, $L_d \& L_q$ are the d and q-axis inductances respectively, $i_d \& i_q$ are the d and q-axis currents respectively.

The total current i can be written then as follows.

$$|i| = \sqrt{{i_d}^2 + {i_q}^2}$$

These currents can be represented by a phasor diagram as follows.

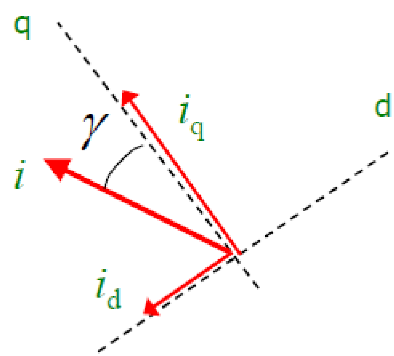

Figure 1: Phasor diagram of d and q-axis currents

Now, we can write the d and q-axis current as,

$$i_d = -|i|\sin\gamma$$

$$i_q = |i|\cos\gamma$$

Now, if we substitute these d and q-axis current equation back in the torque equation and then calculate the maximum torque, we can find the maximum value of the angle (γ) between the q-axis and the demand current, i. The maximum angle is thus given by the following equation.

$$\gamma_{\max T} = \sin^{-1}\left\{\frac{-\Psi_m \pm \sqrt{\Psi_m{}^2 + 8|i|^2 \Delta L^2}}{-4\Delta L|i|}\right\}$$

Where, $\Delta L = L_d - L_q$, is the saliency. Now, with this angle information, we can recalculate the value of the d and q-axis currents for the maximum torque.
To realise this in the simulation, we can build a look up table for d and q-axis currents for a given torque value. Similar approach is considered in the simulation.

Now, when we try to achieve the higher speed above the base speed, the back emf increases in proportionality with the speed. So, to counter this increase in back emf, we need to decrease the d-axis current i.e., increase the d-axis current negatively to produce the counter back emf. As d-axis current is responsible for producing the field, and we can reduce it, this is so called the field weakening. FW are done by several methods. Field weakening by injecting direct negative d-axis current is used as shown in the Figure 2. In direct i_d method, we generate the d and q-axis current reference even in field weakening region (Look up tables are used in simulation), and the voltage is compared with the limit value which then is fed to a PI controller to produce a negative d-axis current to be injected directly in the d-axis. Doing so, the q-axis current the must be limited as the maximum current limit must be maintained.

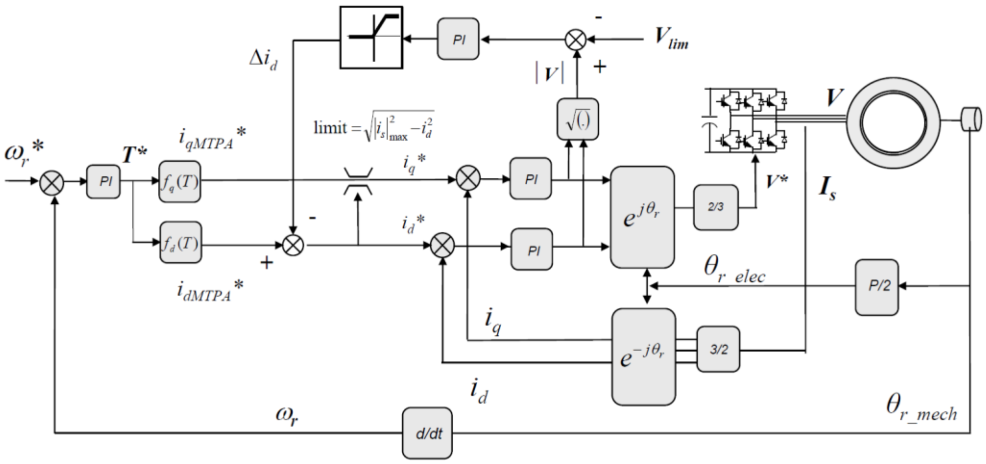

Figure 2: Field Weakening by direct Id method

The field weakening can also be done by generating the current references which are calculated by using the current circle and the voltage ellipse. The equation governing both are as follows. The voltage limit equation can be written as,

$$\left(L_q i_q\right)^2 + \left(L_d i_d + \lambda_{pm}\right)^2 \leq \frac{V_{\max}^2}{\omega_e^2}$$

And the current limit equation can be written as,

$$i_q = \sqrt{i_{max}^2 - i_d^2}$$

Now the voltage limit equation can be rewritten as,

$$\left(L_d^2 - L_q^2\right) i_d^2 + 2\lambda_{pm} L_d i_d + \lambda_{pm} + L_q^2 i_{max}^2 - \frac{V_{\max}^2}{\omega_e^2} = 0$$

Now, the values of d and q axis currents in the field weakening region can be calculated as,

$$i_{dfw} = \frac{-\lambda_{pm} L_d + \sqrt{\left(\lambda_{pm} L_d\right)^2 - (L_d^2 - L_q^2)(\lambda_{pm}^2 + L_q^2 i_{max}^2 - \frac{V_{\max}^2}{\omega_e^2})}}{L_d^2 - L_q^2}$$

$$i_{qfw} = \sqrt{i_{max}^2 - i_{dfw}^2}$$

And finally, the torque in the field weakening region can be thus calculated as,

$$T_{fw} = \frac{3}{2} p(\lambda_{pm} i_{qfw} + \left(L_d - L_q\right) i_{dfw} i_{qfw}$$

2.4 Motor Current Control

To perform the motor current control, the following dynamic equations of the motor in dq reference frame are used.

$$V_d = i_d R + L_d \frac{di_d}{dt} - \omega_r L_q i_q$$

$$V_q = i_q R + L_q \frac{di_q}{dt} + \omega_r L_d i_d + \omega_r \psi_m$$

Where, V_d & V_q are the d and q-axis voltages respectively, ω_r is the rotor speed, R is the stator phase resistance.
Now, these mathematical equations can be used to design the current PI controller both in d and q-axis. To model the controller, only the following terms are used.

$$V_d = i_d R + \frac{L_d di_d}{dt}$$
$$V_q = i_q R + \frac{L_q di_q}{dt}$$

And the rest of the terms are used as feed forward terms.
For a fixed allowed maximum overshoot (M_p%) and the controller bandwidth, the d and q-axis current controller gains can be calculated as follows.

$$K_p = 2\pi BW * \zeta * L_{s(dq)} - R$$
$$K_i = (2\pi BW)^2 * L_{s(dq)}$$

The damping factor ζ can be calculated from the chosen maximum overshoot as follows.

$$\zeta = \cos\,[\tan^{-1}(-\frac{\pi}{\log(M_p)}]$$

A maximum overshoot of 2% is considered and the chosen bandwidth is 150Hz.

2.5 DC-DC Converter Control

As we know that the battery voltage varies based on the state of charge of the battery. To supply a constant dc voltage to the inverter which drives the motor, we need to control the voltage input from the battery to make it more stable. The dc-link voltage control of the DC-DC converter is as done as per the block diagram shown in Figure 3 . As it can be seen, the load current is feed forward. The inductor voltage is used to calculate the duty cycle.

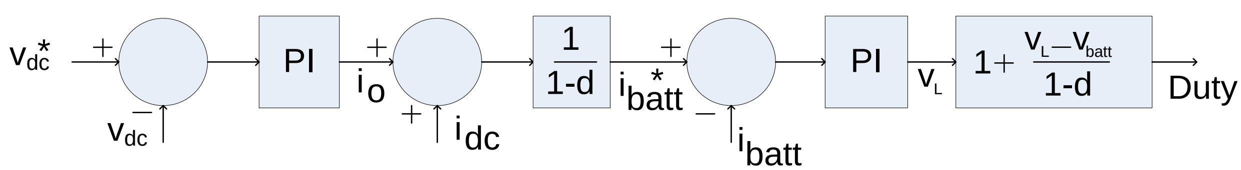

Figure 3: DC-DC Converter control block diagram

As it can be seen in the dc-dc converter control block diagram, we have two controllers: Voltage controller and the Current controller. The battery side inductor current controller is design using a standard pole-zero cancellation technique. The current controller gains are calculated as follows.

$$K_p = 2\pi BW * L_{dc}$$
$$K_i = 2\pi BW * R_{L_{dc}}$$

Where, L_{dc} is the input side inductor and $R_{L_{dc}}$ is the series resistance of the inductor. The bandwidth (BW) chosen is 100Hz.
The DC-link voltage controller is designed considering a fixed overshoot and a fixed bandwidth. The controller gains are calculated as follows.

$$K_p = 2\pi BW * \zeta * C_{dc}$$
$$K_i = (2\pi BW)^2 * C_{dc}$$

Where, C_{dc} is the dc-link capacitor and ζ is the damping factor calculated using a fixed overshoot of 2% as already discussed in the motor current control section.

3. Electrical Car Model

The electric car model block consists of the following block models.

3.1 Inverter

The inverter is used to convert the DC link voltage into the three phase ac voltages which are then supplied to the electric motor. The modulation of inverter becomes a very important factor to be considered here in this block. There are various modulation techniques however a Space Vector Modulation (SVM) technique is used as it allows higher voltage limits. The voltage limit using SVM technique is $\frac{V_{dc}}{3^{\frac{1}{2}}}$ without using any overmodulation. As this is the peak value, the root mean square value then becomes $\frac{V_{dc}}{6^{\frac{1}{2}}}$. The voltages obtained from the motor current control block (MTPA+FW) are limited to this voltage limit. This voltage limit is achieved when the motor speed reaches the base speed and after that in field weakening region, it remains constant at this limit.

3.2 Electric Machine

Permanent magnet (IPM) motor is used to produce the required torque. The motor parameters are considered as per the table in the parameters section above. The torque from the machine is then fed to the drivetrain for transmission to the wheels. Three phase voltages from the inverter are then fed to the electric machine. In simulation, an interior

3.3 Drivetrain

The drivetrain consists of gearbox with the gear ration as already indicated in the parameters section above. A gearbox already available with the Simulink toolbox is used to simulate.

3.4 Car Frame

The vehicle body available in the Simulink library is used and a rear wheel drive is selected. The vehicle parameters already indicated in the parameter section are considered.

3.5 Mechanical Braking System

A mechanical brake already available in the Simulink library is used which puts the brakes in the tyres based on the braking torque command from the torque sharing model block. A disc brake or normal clutch brake can be used to simulate. The brake action can be shared between both front and rear wheel with appropriate gain.

3.6 Battery

A standard Li-ion battery model available in the Simulink library is used and the battery parameters as already indicated in the parameters section above are considered. The battery model gives the battery current, battery voltage and battery state of charge information. The battery voltage is then fed to the DC-DC converter model block to control the DC-link voltage.

4.Sensors

The sensor block consists of various sensors such as vehicle speed sensor, electric machine rotor sensor block, battery current sensor, dc-link voltage sensor, motor current sensor etc which are used to feed back to the various block models to calculate the real time control signals.

5. Simulation Results

The simulation results are analysed both in mechanical and electrical domain. Three separate models: mechanical, average model, and switching model are analysed.

5.1 Mechanical Model

The acceleration test is done through a mechanical model. PI controller to produce the torque throttle using speed refence and a lookup table to obtain the machine torque based on the vehicle speed are considered to model the mechanical system. The manufacturer's specification on acceleration performance of 0-100km/hr in 7.2 sec is well achieved at 6.6 sec. Moreover, the acceleration performance of 80km/hr-120km/hr in 4.9 sec is achieved in 4 sec. These performances are based on the controller values designed based on the vehicle specification. The simulation results are as shown in the figure below.

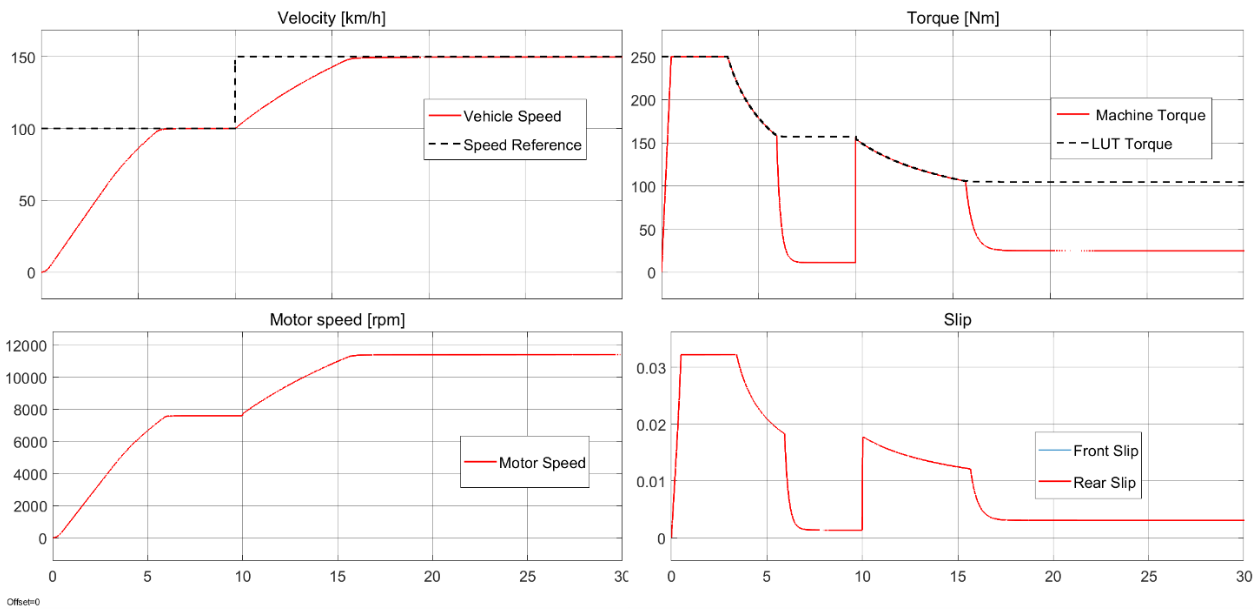

Figure 4: Vehicle mechanical model simulation results

5.2 Average Model

The average model consists of a driving profile block, cruise controller model block, and the electrical car model block. The feedback sensors are used to measure the values to be fed back to the controller to compute the control actions. These blocks have several subblocks computing the different control action which are already explained. The average model is simulated using a built-in MATLAB model for DC-DC converter and the inverter. The same control strategies as taught in during the class are used. The model is tested by using a step up and step-down speed reference and a standard speed profile (WLTP3) from MATLAB library. The following results as shown in the figures below are obtained.

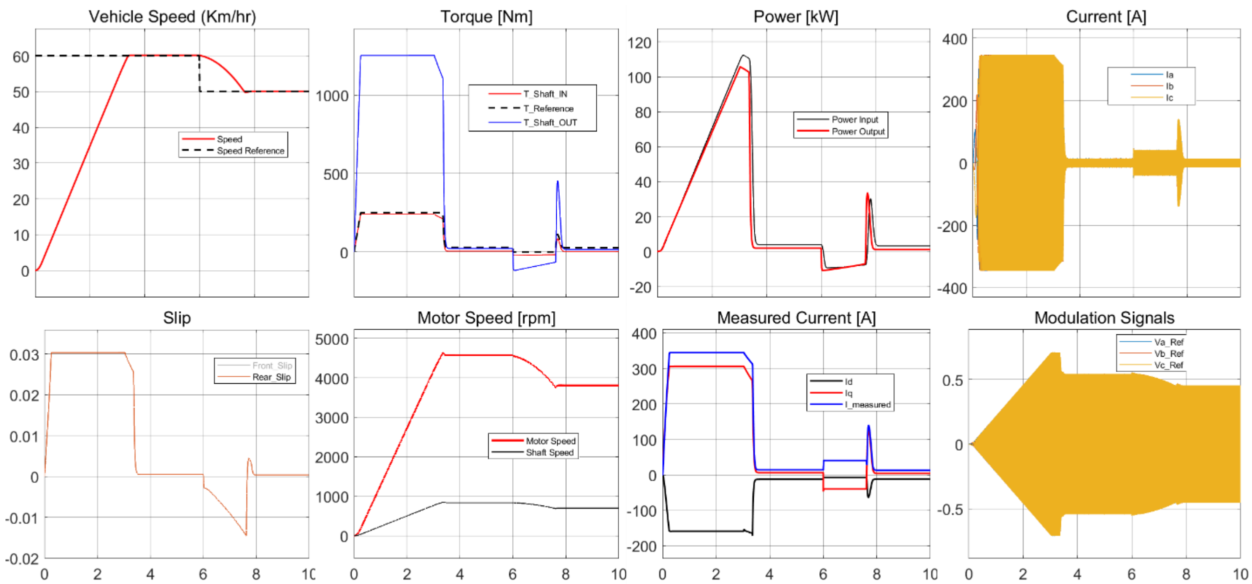

Figure 5: Vehicle performance results for step-up and step-down speed profile

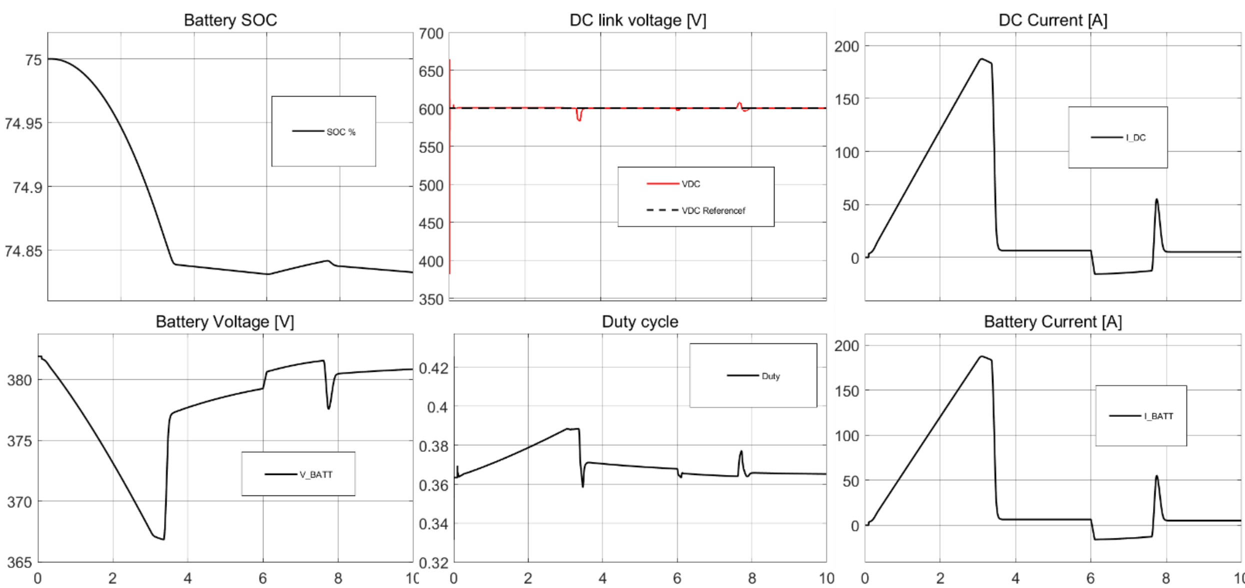

Figure 6: DC-DC converter performance results for step-up and step-down speed profile

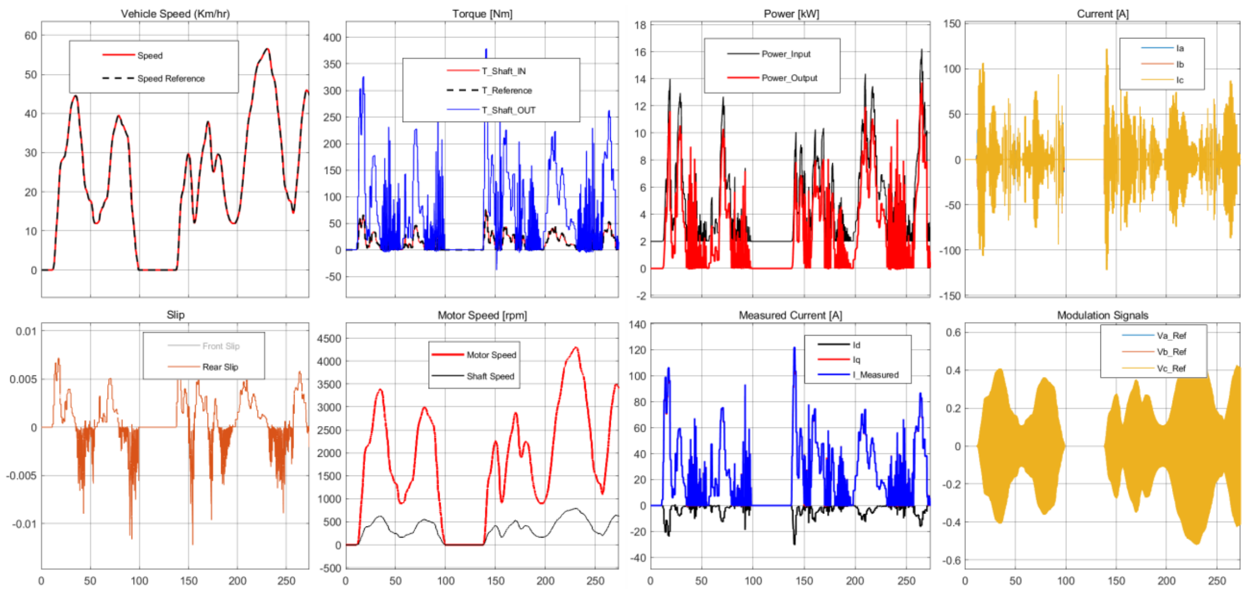

Figure 7: Vehicle performance results with WLTP 3 speed profile

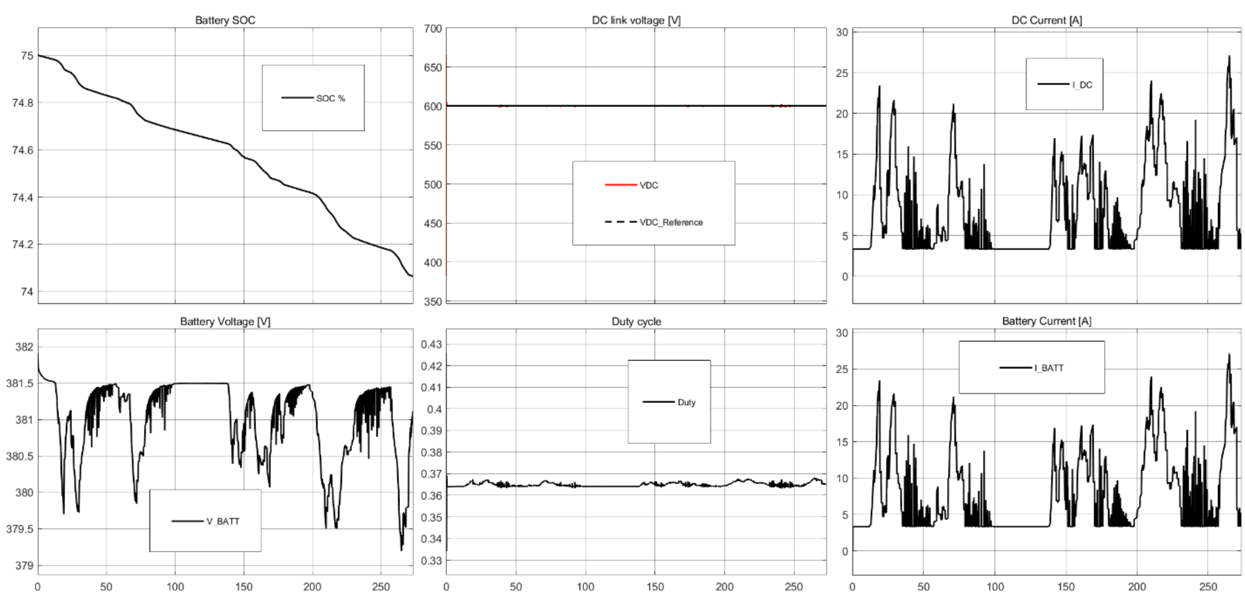

Figure 8: DC-DC converter performance results with WLTP 3 speed profile

5.3 Switching Model

The switching model consists of all the blocks that the average model consists of and in switching model, the DC-DC converter and the inverter are not the average models, but they consist of the circuital connection of high frequency switch (MOSFET) which are fed with the switching pulses from the gate driver modules available in the MATLAB library. The switching frequency of 20kHz is used to simulate the switching blocks and the sampling time is chosen accordingly. The switching model is tested by giving the step-up speed reference. The results obtained are as shown below in the figures.

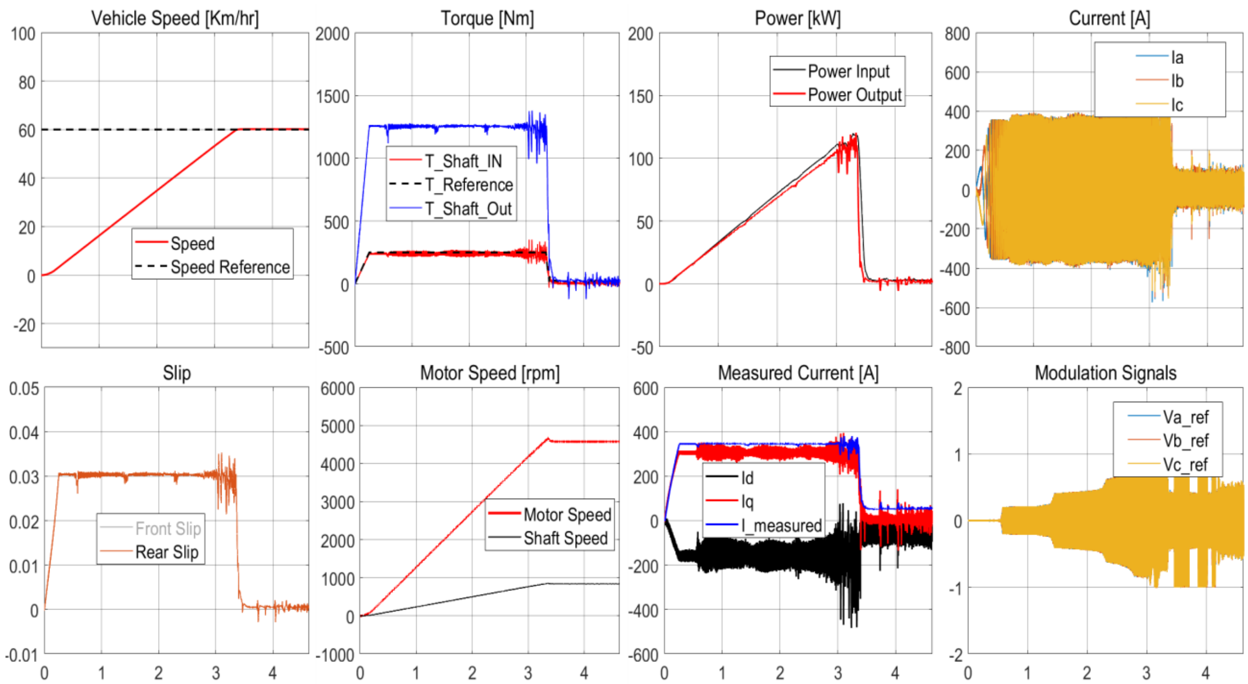

Figure 9: Vehicle performance results for step-up speed profile

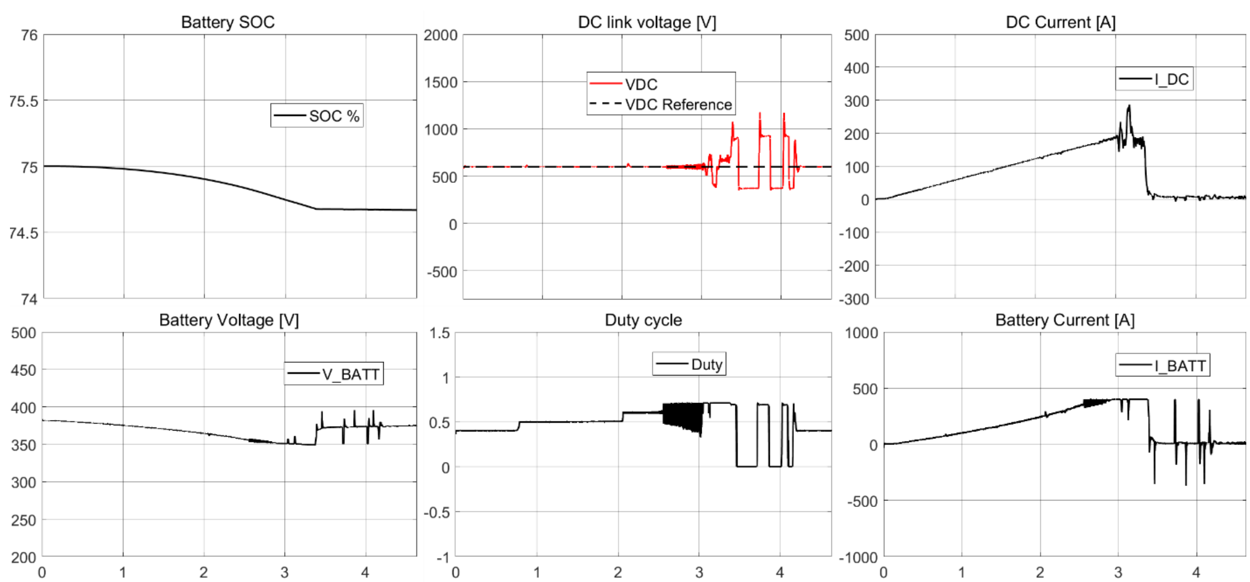

Figure 10: DC-DC converter performance results for step-up speed profile

6. Conclusions

The multidomain (electrical, mechanical, electro-mechanical) simulations of an electric car (BMWi3) with mechanical model, average full model and the switching full model are simulated, and results are presented. Some of the parameters like vehicle speed, power, torque and battery voltage and current are checked against the manufacturer's specifications and the result seem to be closely matched if not exactly with that of data provided by the manufacturer. The control algorithms are developed based on the manufacturer provided electrical and mechanical specifications. The controller used are the proportional integral which are a good choice in terms of reference tracking and the load rejection capability. The controller gains are calculated and tuned for the dynamic performance, steady state error, disturbance rejection etc. The anti-windup scheme is adopted to prevent the integrator windup problem of the controller. The mechanical model verifies the vehicle acceleration against the manufacturer's data of 0-100km/hr. The average model is checked by giving the step-up and step down, and the WLTP 3 speed profile from the MATLAB. The switching model is checked only by giving the step-up speed reference. It is evident that the switching model takes long time to simulate for the given sampling time and the switching frequency thus the simulation times have been reduced and thus the model is checked by giving the lesser speed reference.

Appendix

```
%MATLAB SCRIPT%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all;
% Battery - Li-ion/8mod*12cells
Cn_bat = 60;  %Ah
Vn_bat = 353; % V
E_bat = 22; % kWh, 18.8 kWh net capacity. Other models of the BMWi3 have higher
capacity batteries.
Imax = 409; % A,
https://wiki.aalto.fi/download/attachments/91692283/high_voltage_batteries_of_bmw_veh
icles.pdf?version=1&modificationDate=1398446470505&api=v2
% On some simulations this was increased to 900 A  explore the capabilities of the
% control scheme
Pmax_bat = 125; %kW, (assumption, same as motor)
Initial_SOC = 75;   % [percentage]
Bat_response = 10; % [sec]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% DC/DC Converter - Values designed for the appplication, not part of BMWi3 data
%The data suggests this car model is fed directly from the battery, but to
%match the proposed topology, one can be introduced, with a DC voltage of
%at least the maximum achieved by the battery

Vdc = 600; %V

%For the same reason, there is no data about a dc/dc converter interfacing
%with the battery, so the design available from ESHEV will be used

fsw_boost = 20e3; %Hz
L_boost = 1e-3; %H
R_Lboost = L_boost*250; %ohm
C_boost = 5*475*1e-6; %F
R_Cboost = 1.8e-3; %ohm

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%  Boost converter controller parameters
% Method thought in the Energy Storage subject
% Current controller params
% Zero-pole cancellation
BW_i =    100;  %[Hz] Closed-loop bandwidth for IL  before 50 [100 Hz for Avg, 30 Hz for SW]
Kp_idc = 2*pi*BW_i*L_boost;
Ki_idc = 2*pi*BW_i*R_Lboost;
% Voltage controller params
% Free selection of parameters
BW_v =  20;  %[Hz] Closed-loop bandwidth for Vo [20 Hz for Avg, 5 Hz for SW]
Mp = .2;                    %[pu] Overshoot
Wn_v = 2*pi*BW_v;           %[rad/s] Natural frequency
xi = cos(atan(-pi/log(Mp)));
Kp_vdc = 2*xi*Wn_v*C_boost;
Ki_vdc = Wn_v^2*C_boost;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% DC/AC Inverter

%Modulation scheme->sine-triangle with zero harmonic injection (assumption)

fsw_inv = 10e3; %Hz % From "Adopting MOSFET Multilevel Inverters to Improve the Partial Load Efficiency of Electric Vehicles"; F. Chang, O. Ilina, O. Hegazi, L. Voss, M. Lienkamp;
fsw_inv1 = 10e3;%average model switching frequency
fsw_inv2 = 20e3;%swithing model switching frequency
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%fsw_inv = 1e3; %for WLTP3 speed porfile test, average model
%Tsim = 1e-4; %sample time for WLTP3 speed profile test, average model
Tsim = 1e-5; % Average model sample time
Tsim_2 = 5e-6; % Switching model sample time


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Machine Parameters
% Machine - hybrid synchronous

% Performance
% V range = 250 - 400 V
Tmax_mach = 250; %Nm % https://www.trader-
media.ie/templates/bmw2015/forms/brochure_pdfs/E229404_BMWi.pdf
Pmax_mach = 125*1e3; %W
wb_mach = 4800*pi/30; %rad/s, @75kW
nmax_mach = 11400; %rpm % https://evobsession.com/bmw-i3-details/
wmax_mach = nmax_mach*pi/30; %rad/s

% Electrical data and lumped parmeters

In_mach = 240*sqrt(2); %A;
Imotor_max = 400; % A phase;
Vn_mach = 400; %V
p = 6; % pole pairs %http://hybridfordonscentrum.se/wp-
content/uploads/2014/05/20140404_BMW.pdf
Ld_mach = 0.0858*1e-3; %H
https://ssl.lvl3.on24.com/event/13/49/76/5/rt/1/documents/resourceList1488311786346/w
ebinar_power_point.pdf
Lq_mach = 0.24*1e-3; %H
R_phase_mach = 0.005225; %ohm
J_mach = 0.0666; %kgm2
% Xd_mach = 0.615; %ohm
% Xq_mach = 1.72; %ohm
ke_mach = 41.67; %V/krpm
% kt_mach = 0.523; %Nm/A
% n_noload_mach = 8950; %rpm
flux_mach = 200/(6*4800*pi/30); %Wb
deltaL_mach = Ld_mach - Lq_mach; %H

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Motor Current Controller
% d axis
Bw_c = 150*2*pi;   % Bandwidth of current controller
Mp_c = 0.02;       % Maximum overshoot 2%
xi_c = cos(atan(-pi/log(Mp_c)));
Kp_id = 2*xi_c*Bw_c*Ld_mach-R_phase_mach;   % Proportional gain id controller
Ki_id = Bw_c^2 * Ld_mach;     %  Integrator gain id controller
% q axis
Kp_iq = 2*xi_c*Bw_c*Lq_mach-R_phase_mach;   % Proportional gain iq controller
```

```
Ki_iq = Bw_c^2 * Lq_mach;      % Integrator gain iq controller

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% MTPA Values of id and iq
% Values used for look-up table
I_step = 0.1;    % [A]
I_demand = -Imotor_max-1e-3 : I_step : Imotor_max-1e-3;
beta = acos((-flux_mach + sqrt(flux_mach ^2 + 8* (Ld_mach-Lq_mach)^2 * I_demand.^2))
./ (4*(Ld_mach-Lq_mach)*I_demand));
id = I_demand .* cos(beta);
iq = I_demand .* sin(beta);
T_dq = (3/2)*(p)*(flux_mach*iq+(Ld_mach-Lq_mach)*iq.*id); % Mechanical torque
figure(1)
plot(I_demand,T_dq);
xlim([-500,500]);
grid on;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%torque speed characteristics
Tmax = 250; %%Nm
Pmax = 125; %Kw
Wbase = 4800; %%rpm
Wmax = 11400; %%rpm
w = 0:1:Wmax;
T = zeros(1,length(w));

%%use of logical indexing to access the index of the vectors%%
wbase = Pmax*1e3/Tmax; %%rad/s
wbase_rpm = wbase*30/pi;
T(w<=wbase_rpm) = Tmax;
T(w>wbase_rpm) = Pmax*1e3./(w(w>wbase_rpm)*pi/30);
P = w.*T*(2*pi/60);
figure(2)
plot(w,T,'-'); grid on;
ylim([100,350]);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Wheels - 155/70 R19 84 Q

r_wheel = (15+2*70/(100*175))*0.0254/2; %m

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Mechanical transmission

%single speed transmission
vmax = 150; %km/h
nmax_wheel = vmax*1e3/(60*2*pi*r_wheel); %rpm
k_gear = nmax_mach/nmax_wheel;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Vehicle body % rear wheel drive

mass = 1270; %kg
%50:50 weight distribution
CoG_x = 0;
CoG_y = 0;
CoG_z = 18.5*0.0254; %m % http://www.mybmwi3.com/forum/viewtopic.php?t=1751
wheelbase = 2570;
A_front = 2.38; % m^2
C_drag = 0.29;
%https://www.press.bmwgroup.com/global/article/attachment/T0143924EN/222601
max_load = 1620; %kg
wheel_span = 2.570; %m
rho=1.25;% air density, kg/m3
fr=0.013; % rolling resistant coefficient
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Performance to be expected

% 100km, NEDC, combined -> 13.1 s
% Range (NEDC) -> 359 km
% top speed -> 150 km/h
% Acceleration 0-100 km/h -> 7.2 s
% Acceleration 80-120 km/h -> 4.9 s

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

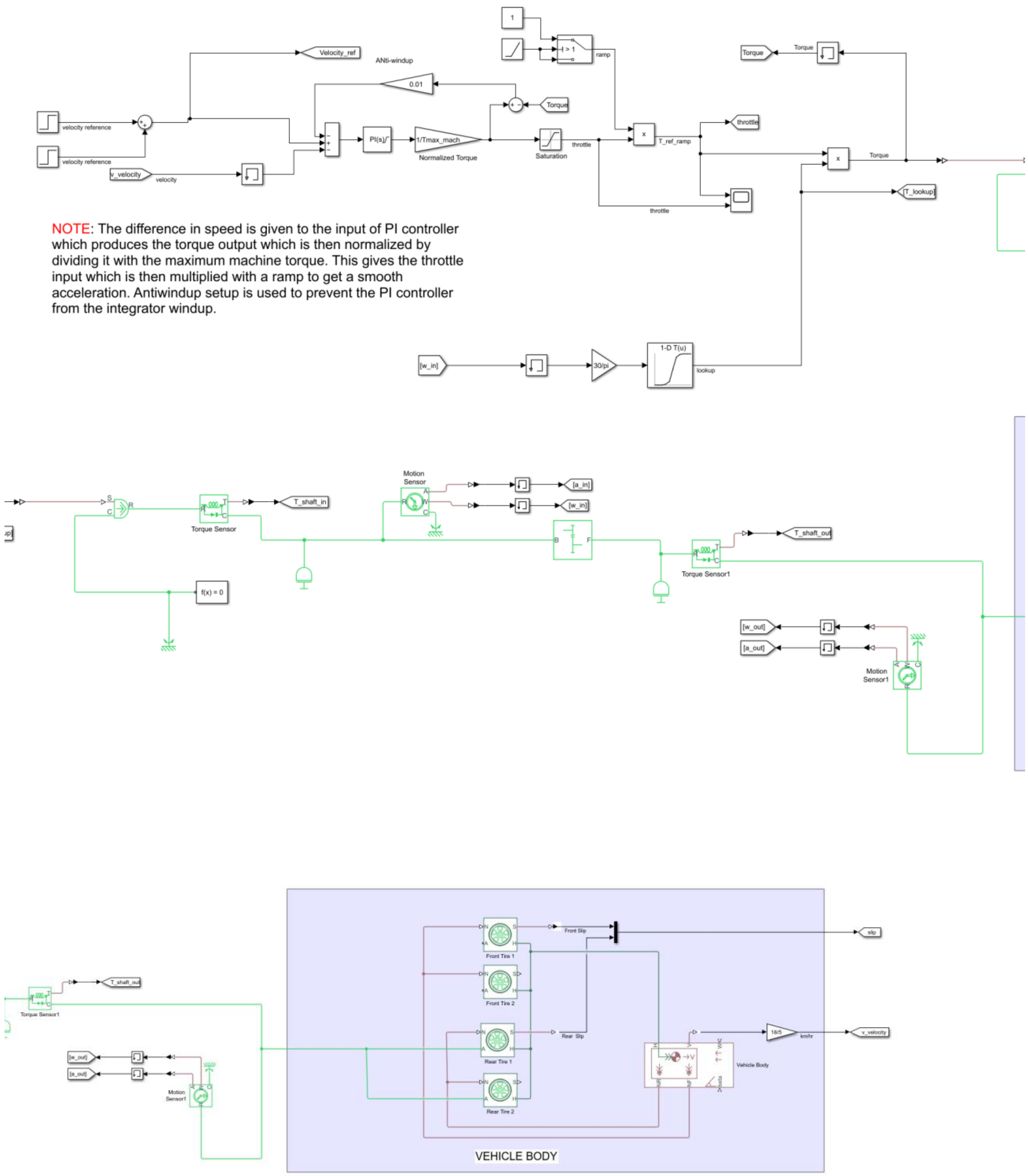

Figure 11: Mechanical Model

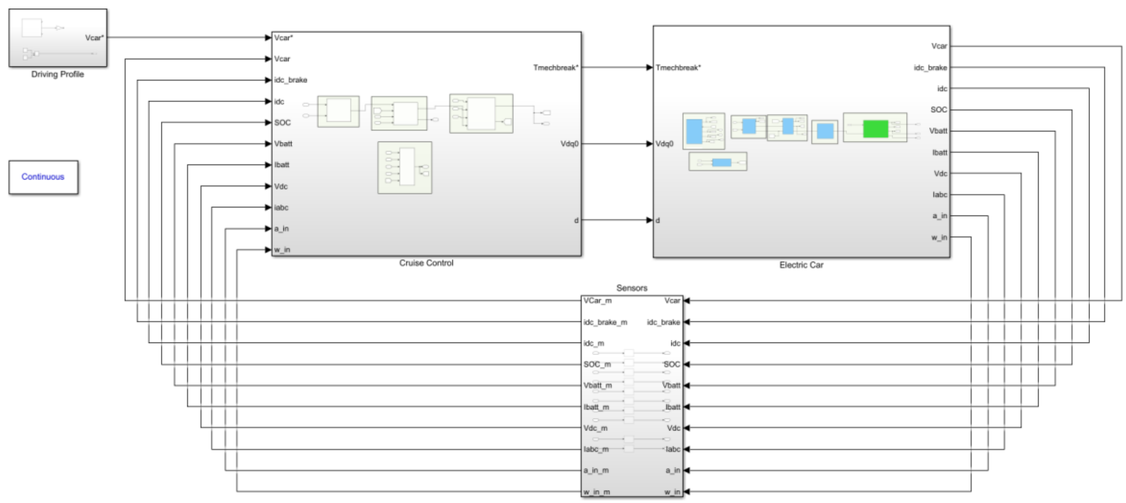

Figure 12: Complete Model Block

References

[1] A. Pan et al., 'The applications of echelon use batteries from electric vehicles to distributed energy storage systems', IOP Conf. Ser. Earth Environ. Sci., vol. 354, p. 012012, Oct. 2019, doi: 10.1088/1755-1315/354/1/012012.

[2] 'BMW i3 Details –'. https://evobsession.com/bmw-i3-details/ (accessed Jun. 05, 2022).

[3] 'The all-new BMW i3.' https://www.press.bmwgroup.com/usa/article/detail/T0149790EN_US/the-all-new-bmw-i3?language=en_US (accessed Jun. 05, 2022).

[4] 'Used 2014 BMW i3 Specs & Features', Edmunds. https://www.edmunds.com/bmw/i3/2014/features-specs/ (accessed Jun. 05, 2022).