Chapter 5

MATLAB and SPICE simulations

Simulations performed in MATLAB and Spice are shown in this chapter. The simulations use models available in Simulink library and spice-models to build the PCU. There are 2 parts of the simulation design, MATLAB and LTspice. First one needs the environment to be setup along with variables, which is explained first and then followed by the model. LTspice uses inbuilt component models and spice directives. If any external file is used, it is mentioned in the text.
Virtual tests are test scenarios to which the above models are subjected to. The results obtained are then presented.
Stress tests are also performed, this includes subjecting the converter to, thermal stress simulation and also over-voltage scenario and collapse of voltage on DC-bus.

5.1 MATLAB setup

The MATLAB setup comprises of multiple modelling subsystems to make abstract models of various general systems including mechanical model, electrical model and environmental effects. 5.2 discusses the design further.

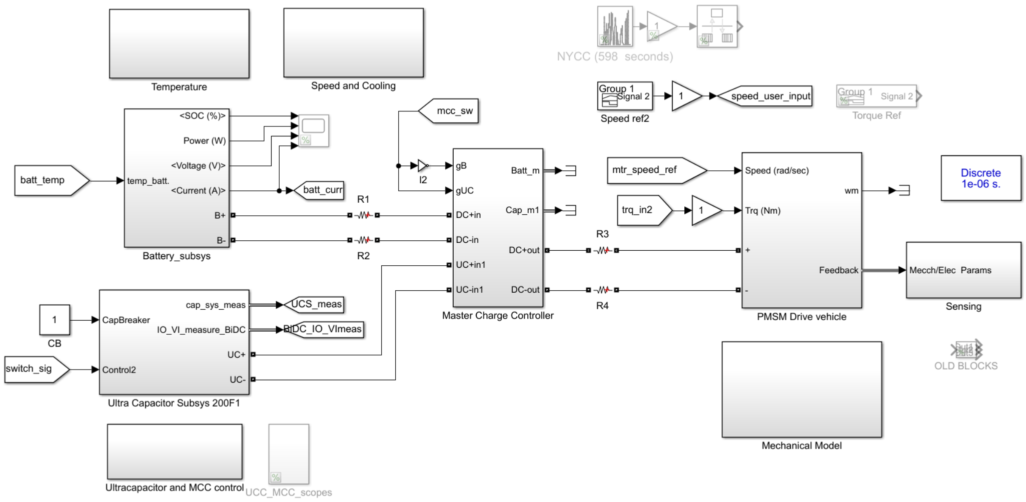

Figure 5.1: Main simulink setup. Full View of subsystems involved

5.2 Simulink setup

The deign comprises of following subsystems each discussed below. Fig. 5.1 shows the simulation setup and design

- Temperature
- Cooling
- Battery
- Ultracapacitor-200F
- Ultracapacitor and MCC control
- Main Charge Controller (MCC)
- PMSM drive
- Drive-sensing block
- Mechanical Model and Environmental-Model

Each subsystem is further explained below,

5.2.1 Temperature

Fig. 5.2 shows the temperature subsystem under the mask. The block consists of 2 parts. One to compute motor temperature and the other for battery. The cooling considered in these cases is both passive and active-cooling derived from vehicle rotor speed. The motor heating is primarily derived from the stator copper losses for current and scaled appropriately for enclosed spaces.

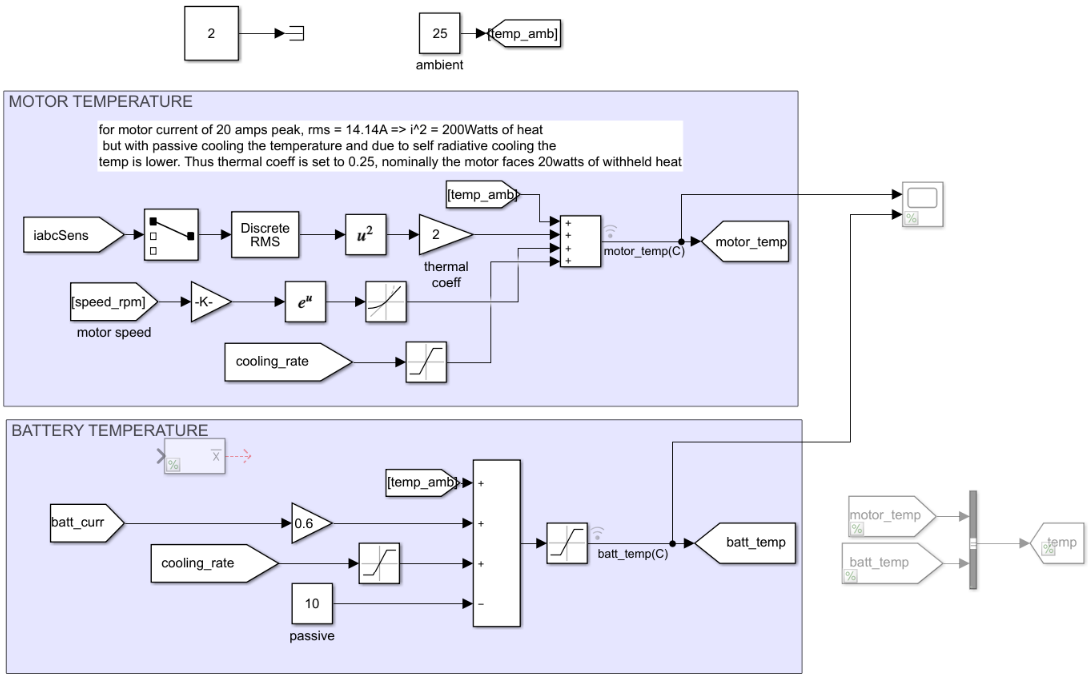

Figure 5.2: Temperature subsystem; Top Motor Temperature; Below Battery Temperature

The equations used are; 5.1 for battery and 5.2 for motor.

$$T_{ambient} + 0.6 * I_{batt} + R_{cooling} - 10_{passive} = sat(T_{batt}) \quad (5.1)$$

$$T_{ambient} + 2 * {I^2}_{motor} + R_{cooling} + 2 * e^{-0.01*rpm} - R_{cooling} = T_{motor} \quad (5.2)$$

where,

$T_{ambient}$ is the ambient temperature set at 25^oC
I_{batt} is the battery-current
I_{motor} is the motor-current in one phase
$R_{cooling}$ is a value determined from cooling subsystem
$10_{passive}$ is passive cooling decrease in temperature
T_{batt} is the battery temperature

rpm is the motor shaft rpm; used to determine cooling derived from the integrated fan.
T_{motor} is the motor temperature

5.2.2 Cooling

Fig. 5.3 shows the cooling equation model used here and the equation used is 5.3

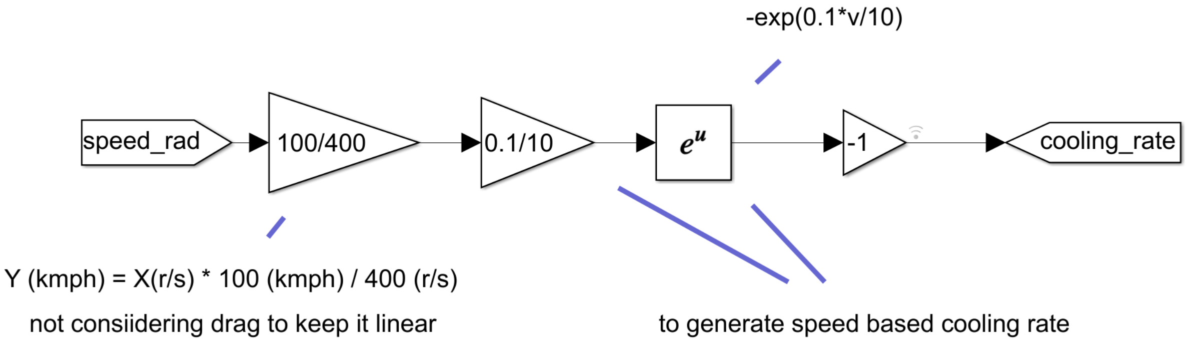

Figure 5.3: Cooling rate derived from motor speed

$$R_{cooling} = -1 * e^{0.1 * v_{mps}/10} * 0.25 \tag{5.3}$$

where,
$R_{cooling}$ is the rate of cooling
v_{mps} is the vehicle velocity in meters per second
0.25 is the nearest linear speed for the rotor rpm

5.2.3 Battery

Figs. 5.4- 5.6 shows the battery subsystem overview with the Management system used for protection/charge and discharge. The relay block is a simple 2 pole configuration with series RC snubbers rated at 100kΩ and 1μF. This avoids harsh switching transients.

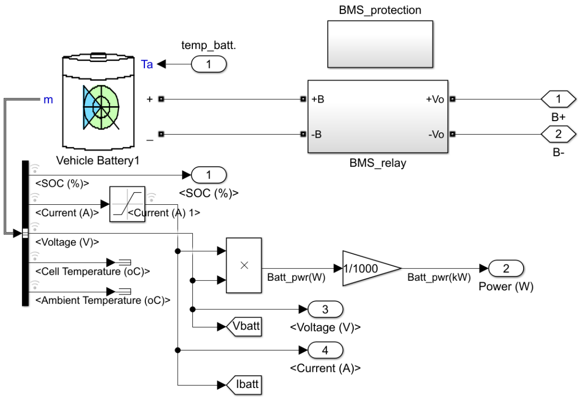

Figure 5.4: Battery subsystem

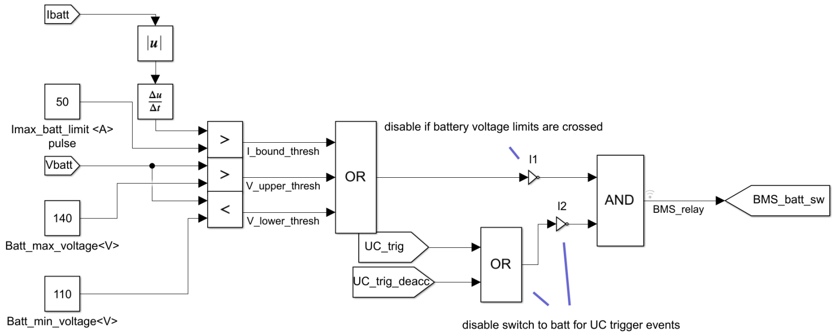

Figure 5.5: Battery-BMS

Note that the BMS module in fig. 5.5 protects the battery in case of multiple scenarios; Under-voltage $110V_{battmin}$, Over-voltage $140V_{battmax}$, Over-Current and Undercurrent whose rate of change is faster than 50A/sample.

The other task is to switch the battery on-to and off-of the DC-bus in case of ultracapacitor activation in MCC, which is here shown by "UC_{trig}" for acceleration events and "$UC_{trig-deacc}$" for deceleration-event.

The battery ratings and parameters are given in the following table;

The discharge characteristics of the battery at different current draws is shown in table 5.7

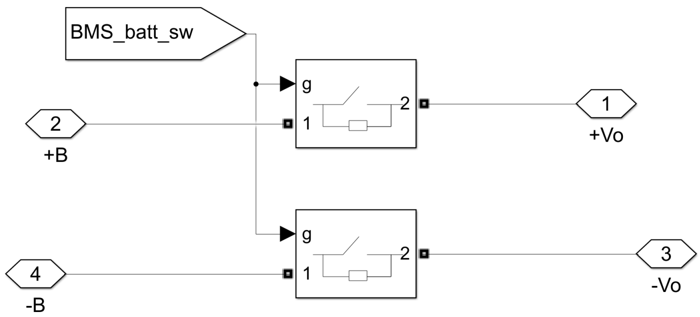

Figure 5.6: Battery-Relay (2-pole)

Battery Parameters and Ratings		
Batt. Char.	**Values**	**Typ.**
Type	Lithium Ion	-
Voltage	$140_{max}V$	130 V
Ah Rating	66 Ah	67 Ah
kWh Rating	8kWh	7.6kWh
Single Cell Voltage	4V	3.8V
Single Cell Ah rating	3Ah	2.9Ah
Series Connected	30	-
Parallel Connected	33	-
Weight of pack (kg)	50 kg	46 kg

Table 5.1: Battery Parameters and Ratings

5.2.4 Ultracapacitor-200F

This subsystem contains the main ultracapacitor and associated bidirectional buck-boost controller. The same is shown in the following figures.

The control signals for the buck-boost converter are supplied by the UC-control block which is discussed in the next section.

The parameters for ultracapacitor are given in table 5.2;

Ultracapacitor Ratings and Characteristics		
ultracap. Char.	**Values**	**Typ.**
Type	Ultracapacitor (Electrolytic)	-
Voltage	$24_{max}V$	20 V
Energy (kJ)	37 kJ	35 kJ
Internal resistance	$15m\Omega$	$25m\Omega$
Capacitance full-bank (F)	200	200
Single Unit Voltage	2.7 V	2.6
Single Unit Capacitance	500	500
Series Connected	8	-
Parallel Connected	4	-
Weight of bank (kg)	2.5 kg	3 kg

Table 5.2: Ultracapacitor Ratings and characteristics

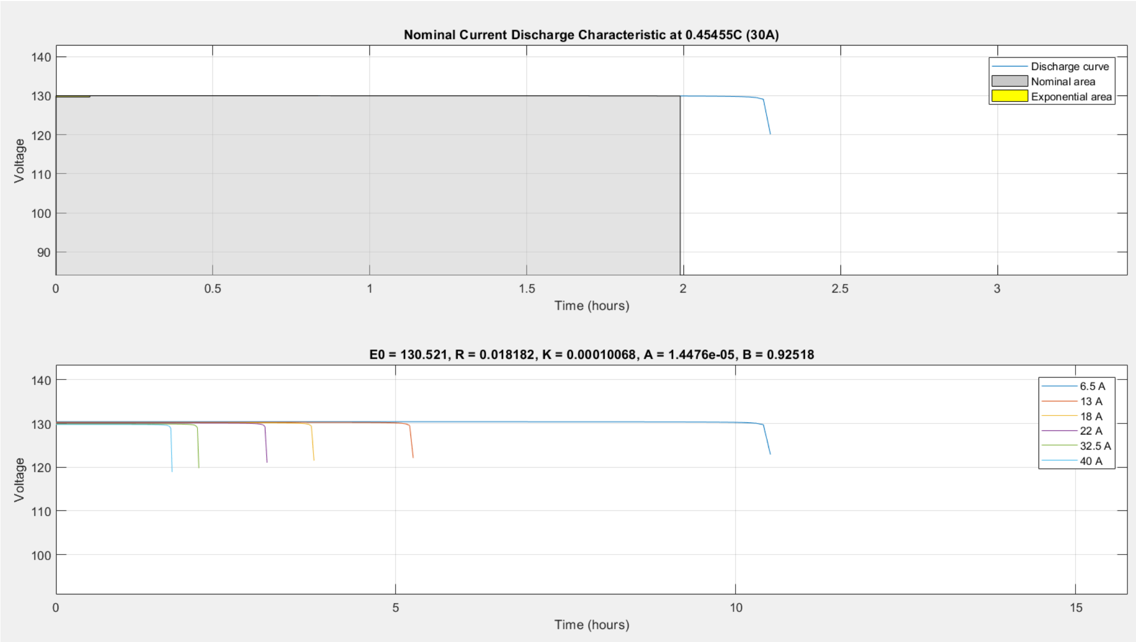

Figure 5.7: Battery discharge-characteristics for various current draws

5.2.5 Ultracapacitor and MCC control

This subsystem is one of the critical subsystems for this design to work properly. The control for MCC, Ultracapacitor relaying, and PWM signals for bidirectional buck-boost converter are generated here. These signals are generated in a closed loop based on feedback from several other subsystems and sensors. Fig. 5.10 shows the Simulink diagram for the same.

The system checks voltage bounds for the proper operation of the ultracapacitor. Lower cut-off threshold is 10V since the buck-boost converter has some stability issues below this value. Also the upper cut-off voltage is 20V since that is the maximum rating of the bank. The ultracapacitor is engaged using three signals, any of which can trigger the MCC switch to ultracapacitor from battery. The first two cases are the acceleration and deceleration triggers. The other is a case where the reverse torque due to any of the reasons; regenerative braking, cruise control etc. goes below 10Nm. These three triggers are also used to activate the buck-boost converter, control for which is shown in fig. 5.11.

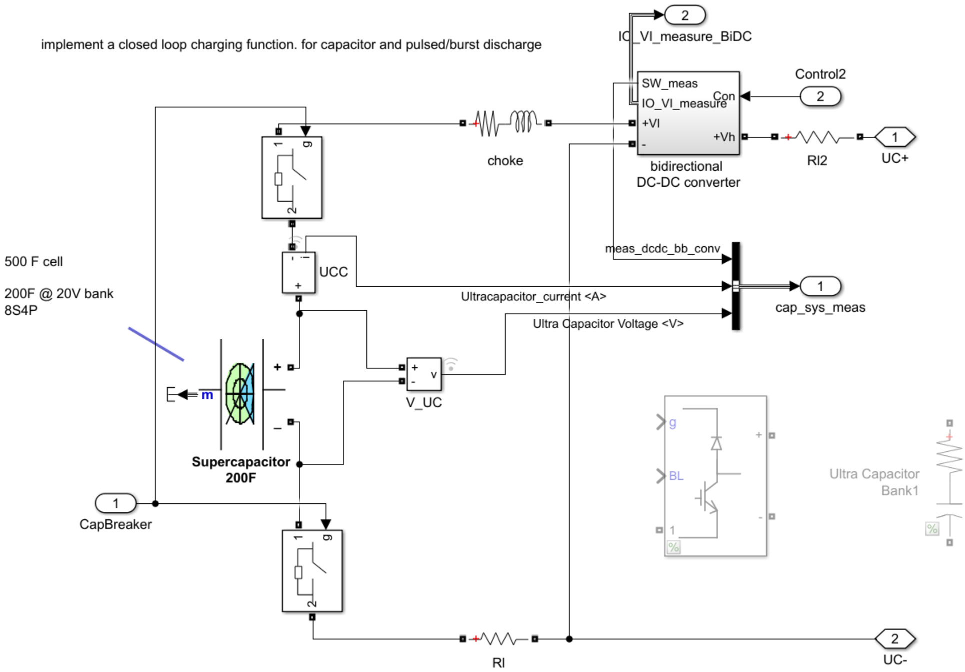

Figure 5.8: Ultracapacitor Subsystem

The duty cycle for the buck and boost blocks has been fixed in the simulation to keep it fast. Also the MATLAB function contains the code shown in fig. 5.12

5.2.6 Main Charge Controller (MCC)

This is a 4 pole double throw switching device which switches between battery and ultracapacitor during events triggered by the control mechanism. Fig. 5.13 shows the internal structure of this setup. It consists of 4 switches operated in pairs and also parasitic losses, a smaller DC link capacitor rated at 220uF and 600V. There are also surge filter capacitors rated at 1μF.

On the left are battery inputs (top) and ultracapacitor inputs (bottom). The output is on the right side.

5.2.7 PMSM drive

The PMSM drive is used because of the permanent magnets and performance characteristics. However a BLDC motor can be equally as good and will not require much modification in

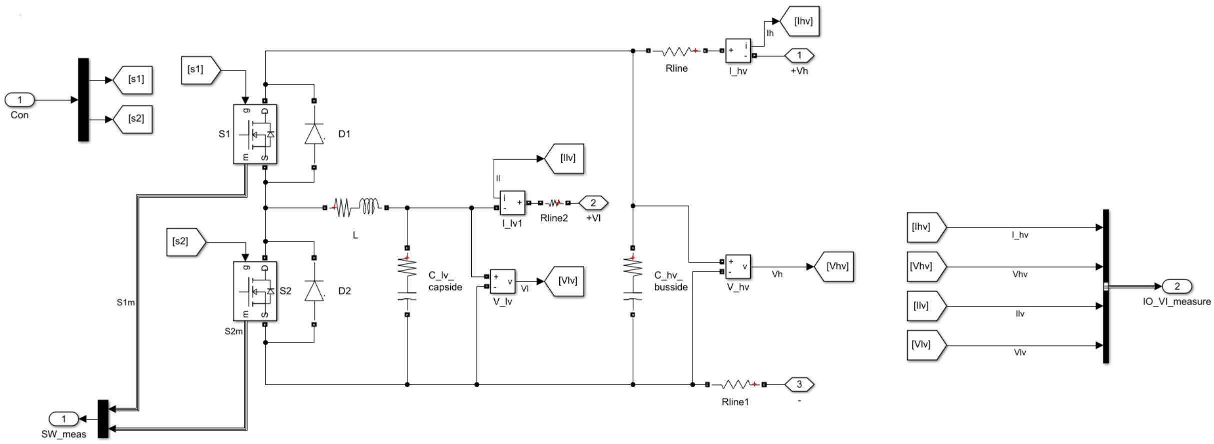

Figure 5.9: Bi-direction Buck Boost Converter Configuration

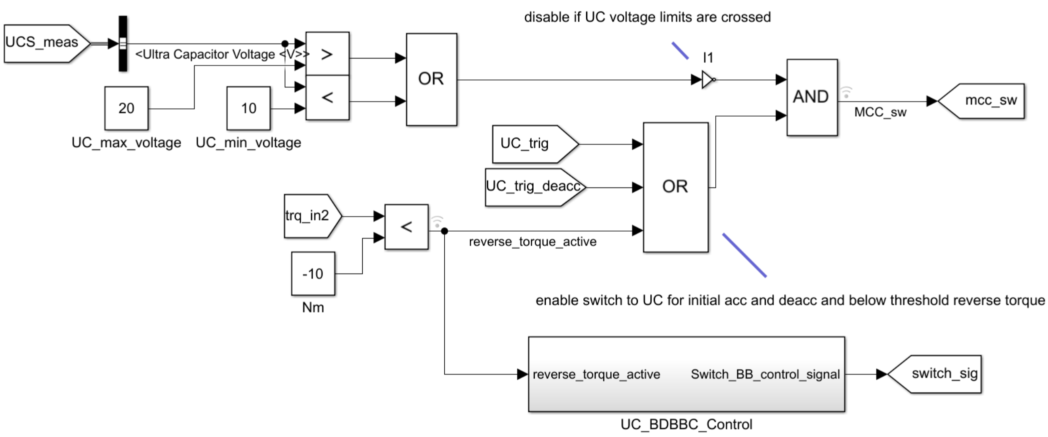

Figure 5.10: Ultracapacitor and Main Charge Controller (MCC) subsystem

terms of control. The block consists of the main drive block and an extension for DC bus over-voltage, under-voltage cut-off-control block. The drive itself consists of a FOC controlled PM synchronous motor. The diagrams for these are shown in figures 5.14, 5.16, 5.15.

FOC or Field oriented control is a technique which involves resolving the stator drive currents into 2 quadrature components I_d and I_q, direct and quadrature currents respectively. The actual torque is produced by the quadrature current and the more near it is to ideal torque angle of 90^o the better the output torque is. The technique involves forcing the direct current to become zero using simple PID control and then using reverse transforms to obtain back the main sinusoidal 3-phase

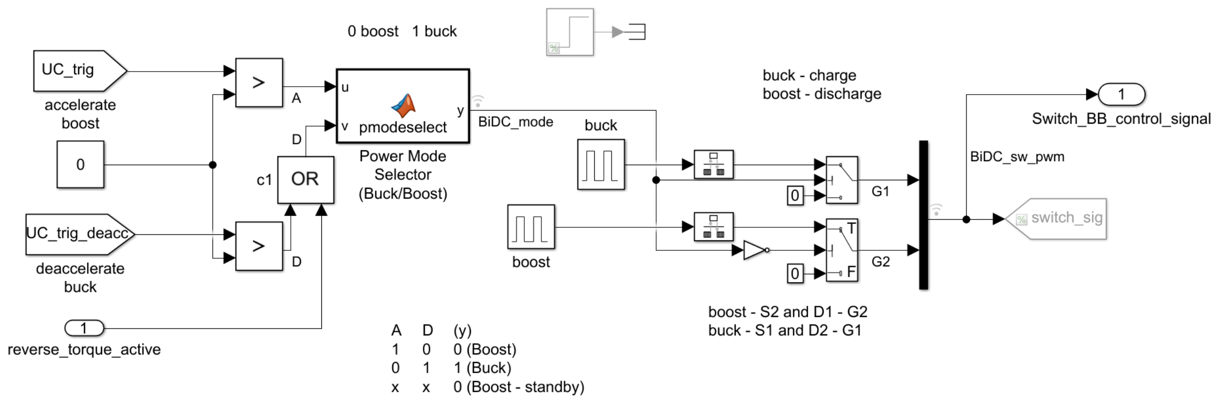

Figure 5.11: Ultracapacitor and Bi-directional buck boost converter (BDBBC) control with PWM generation and buck/boost task switching. Also shown is the truth table for the function logic. The system defaults to Boost or standby; which means the UC is ready to provide power to DC bus.

```
function y = pmodeselect(u,v)

% U - accelerate
% V - decelerate

if u>0 && v<1    %accelerating
    y=0; % boost
elseif u<1 && v>0    %decelerating
    y=1; % buck
else    %default
    y=0; % default boost or standby for power dump
end
```

Figure 5.12: Power Mode Selector Function Code

output drive waveform reference for the inverter. The input to such a control is usually speed from which the necessary torque is determined based on external torque-loading and from that internally computed torque a current reference is generated for the 2 axis obtained after the transforms. Transforms used in this case Clarke and Park transform and their respective inverses.

5.2.8 Drive-sensing block

This block contains signal demultiplexing and logging setup as shown in fig. 5.17

Also the second subsystem included here is shown below

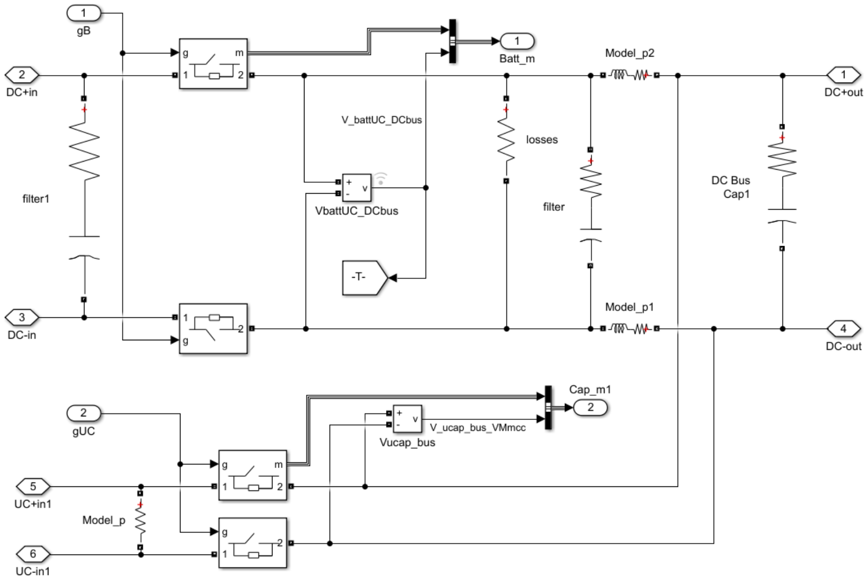

Figure 5.13: Main Charge Controller - MCC

5.2.9 Mechanical Model and Environmental-Model

This section describes the detail regarding simulation mechanical environment design and the vehicle sensing and parameter calculation.

The simulation environment here considers, effect of wind, drag force, gradient and the vehicle weight (maximum scenario is simulated). the mechanical model of the vehicle is also described in this section.

[1]

The full diagram of the mechanical model is shown in fig. 5.19

This subsystem is further divided into different parts which are given below; Each of these sub systems are described further in the following section.

1. Vehicle speed to motor-speed Reference generator

[1]**NOTE!:** It is important to note that this entire simulation is based for a 2-wheeler design which is preferred mode of commute in rural and more dominantly in urban areas due to its manoeuvrability and ease of handling. The model used here thus considers parameters for the 2-wheeler design and urban environment

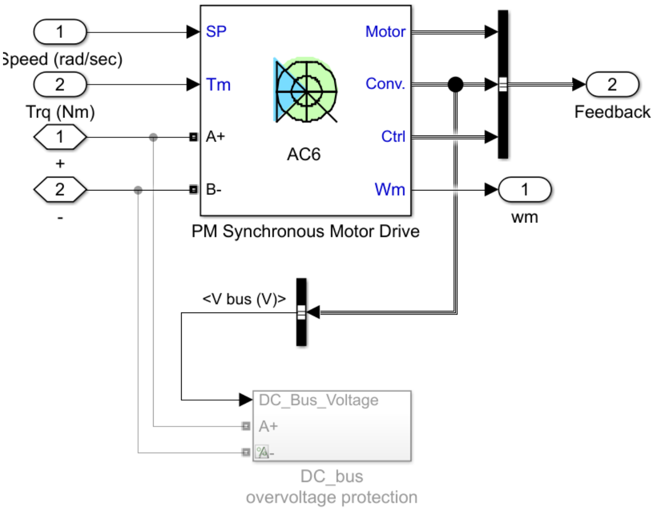

Figure 5.14: PMSM drive along with drive protection subsystem. The bus selector is also used to draw important characteristics from drive during operation.

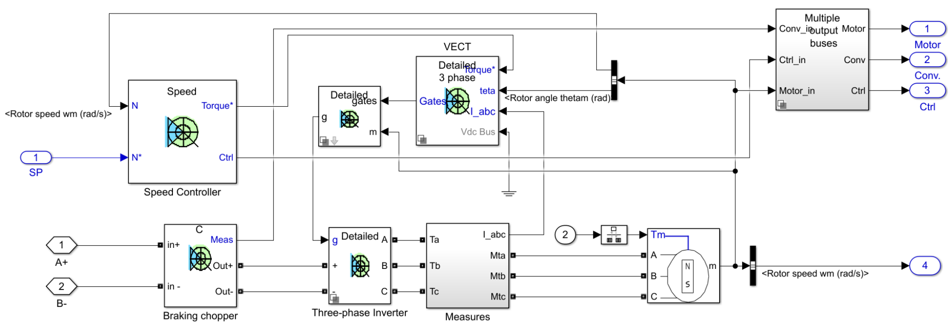

Figure 5.15: PMSM drive, inside. Torque reference generator, Vector controller (FOC-controller) with hysteresis control, PM synchronous motor and braking chopper.

2. Motor speed to Vehicle-speed compute

3. Mechanical Braking and Capacitor Event Triggering

4. Drag force

5. Gradient Effect

6. Toque reference generation

5.2.10 Vehicle speed to motor-speed Reference generator

This set comprises of computing the effective reference for the system given user-reference-input-speed (URIS) and braking level computed later. Also it computes the rotor reference

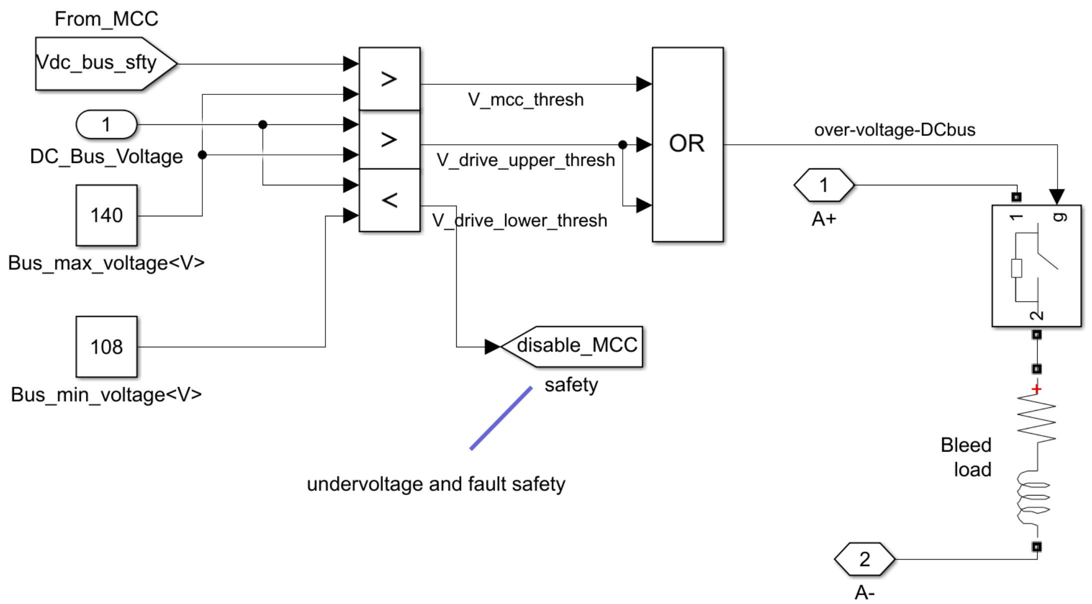

Figure 5.16: DC-bus voltage swing protection. Note the bleeder-load is used to dissipate the energy. It is not chopper based. rather a high dissipation resistive load. The limits here are different than that on the battery side.

speed that would be in radians per second for the drive from URIS and mechanical system constants.

The URIS is shown in fig. 5.20 and the block diagram for this setup is shown in fig. 5.21

The equations for blocks 1 and 2 are given below.

$$URIS - B_{mechanical} = v_{ref} \tag{5.4}$$

$$\frac{v_{ref}}{0.117} * 2.56_{GBratio} = v_{ref-motor} \tag{5.5}$$

Here,
$URIS$ is the user velocity input
$B_{mechanical}$ is the reduction in speed
v_{ref} is the reference velocity for the system

5.2.11 Motor speed to Vehicle-speed compute

The first mechanical constants are considered in these equations. The following table describes the constants and their

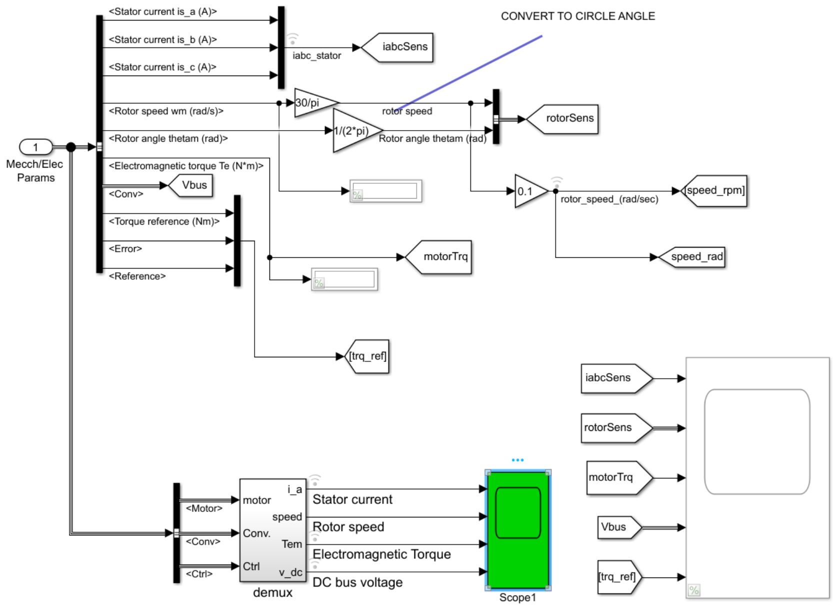

Figure 5.17: Drive output sensing mechanical and electrical parameters

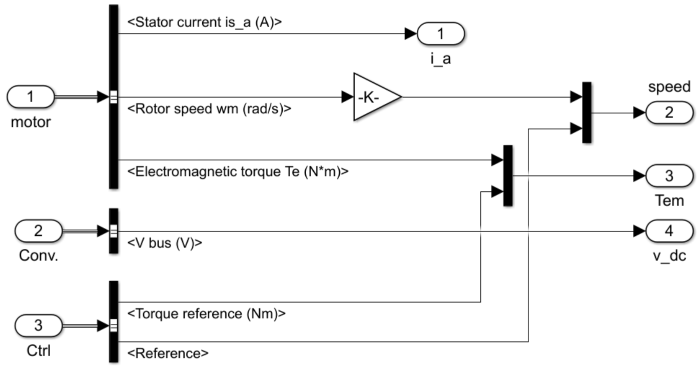

Figure 5.18: Sensing Subsystem 2

values used here. This computation is shown in third block of fig. 5.21.

5.2.12 Mechanical Braking and Capacitor Event Triggering

The mechanical braking and trigger event generator is a complex system which computes several parameters like detecting acceleration/deceleration, rate-of-acceleration or jerk ($\frac{da}{dt}$).

The capacitor MCC triggering events are generated based on detecting early acceleration and early deceleration. signal processing is done to make the pulses smooth and finally the

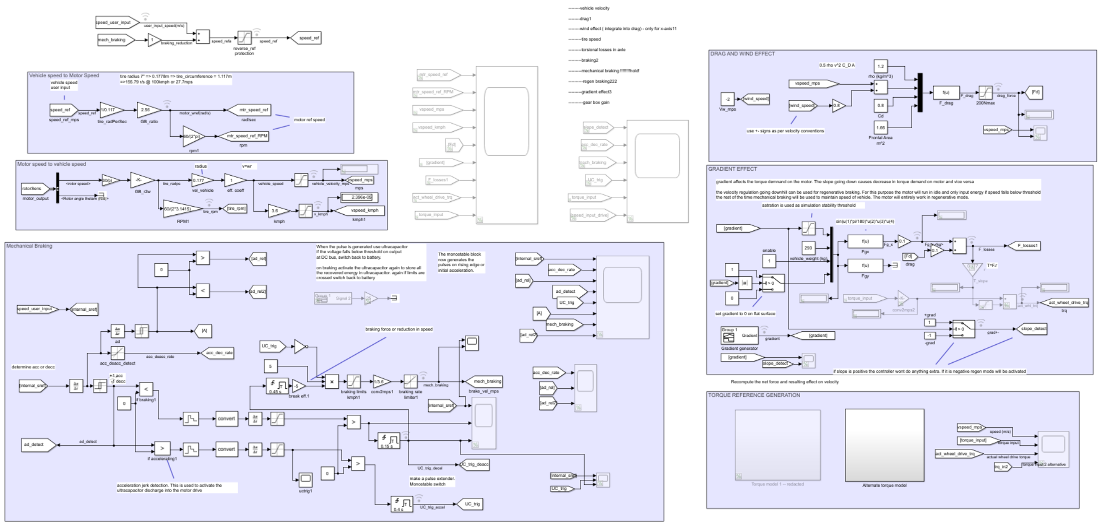

Figure 5.19: Mechanical Model Full. It further consists of 6 main parts

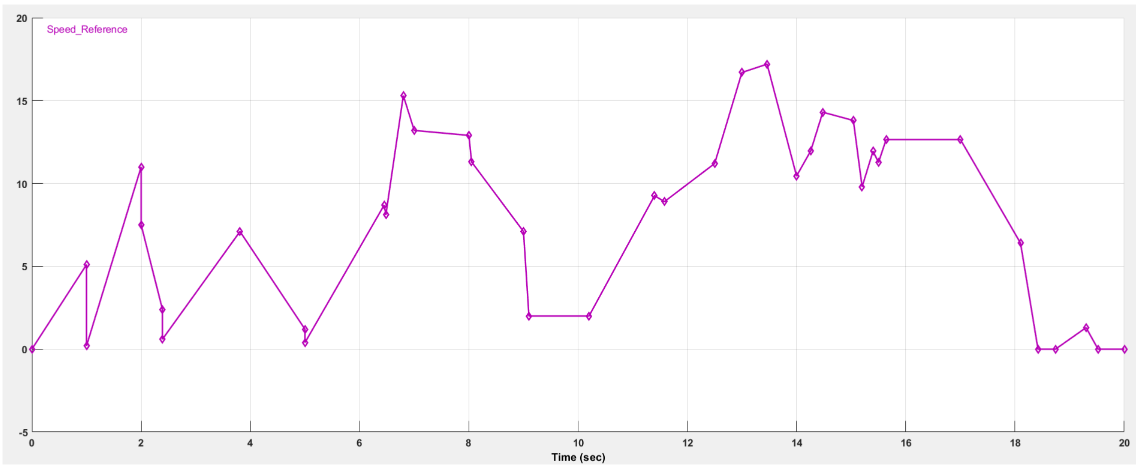

Figure 5.20: User reference input speed in meters/second. This input is derived from the throttle or speed control encoder often placed on the right side of a TEV handlebar.

duration of these triggered events is controlled by monostable timers. For acceleration the timers in the simulation are set to 0.15seconds and deceleration time is set to 0.4seconds. This is however for demonstrative purpose only. In a real world system the deceleration trigger will last as long as the reverse torque and deceleration is maintained, which is simulated in the UC control subsystem discussed earlier. Fig. 5.22 shows the block diagram for this section.

The equation best describing the trigger action for acceleration and braking are given as follows;

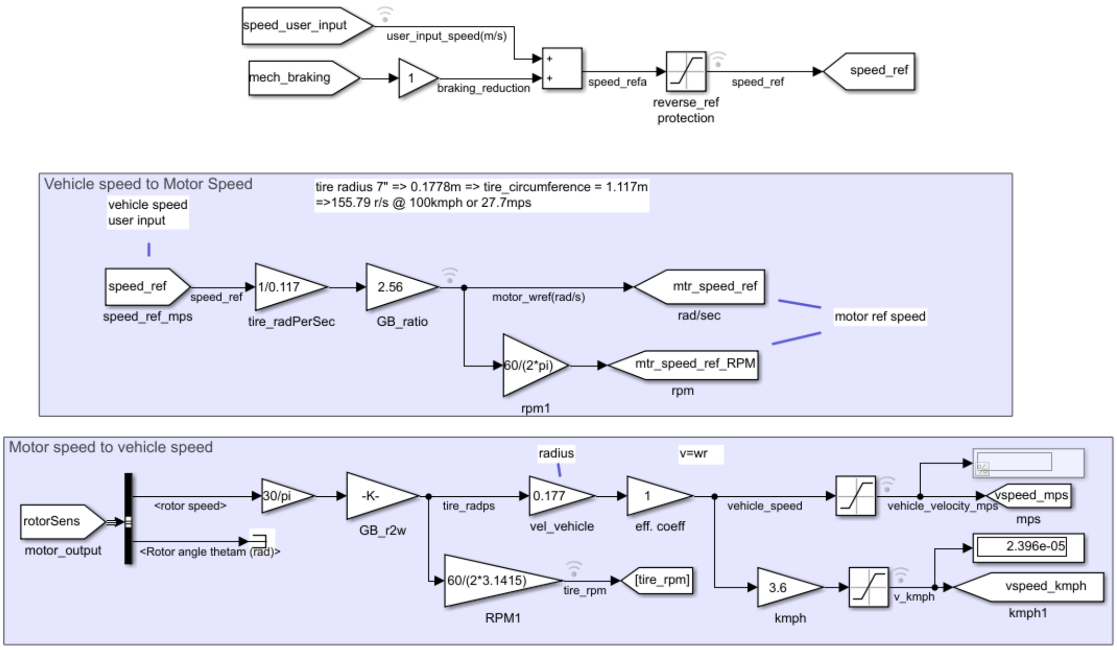

Figure 5.21: a) The error computation with URIS and braking. b) The second block is the calculation of rotor reference speed, the linear speed is converted to a fraction of tire speed using $velocity = \omega * tire_{radius}$, where ω is the angular speed. This is then scaled with the gearbox ratio and fed to the drive where it again internally scaled. c) Third block is used to calculate the actual vehicle velocity obtained based on the motor actual-speed output in rpm.

Vehicle Mechanical Constants		
Parameter	**Value**	**scale**
Tire diameter	14in (0.3556m)	1
Tire Circumference	43.97in (1.117m)	1
Gearbox ratio	2.56 (motor/tire)	0.1 (integrated P control)
Vehicle-Weight (curb)	160-180kg	1
Vehicle-Weight (2x 60kg passengers)	290kg (300kg)	1
Frontal Area	$1.66m^2$	$1.66m^2$

Table 5.3: Vehicle Parameters and Constants

$$a_{detect} = \frac{d(UREF)}{dt} \tag{5.6}$$

$$ad = \begin{cases} -1 & a_{detect} < 0 \\ 1 & a_{detect} \geq 0 \end{cases} \tag{5.7}$$

For braking and acceleration the equation then becomes;

$$de = [accel - or - decel] = \begin{cases} 1decel & ad < 0 \\ 1accel & ad > 0 \end{cases} \tag{5.8}$$

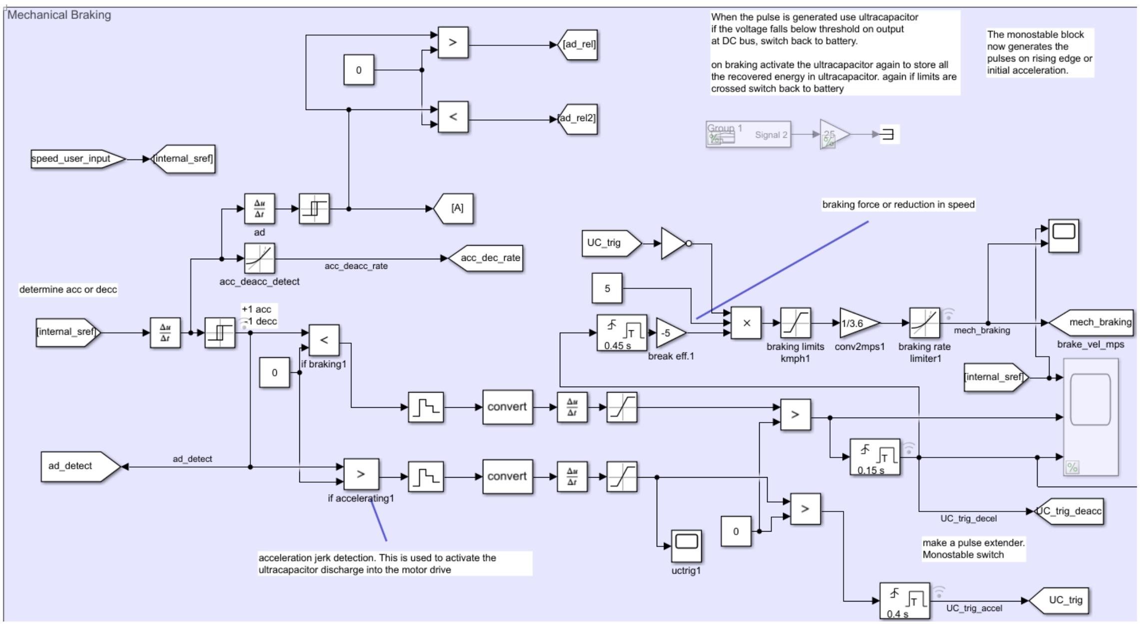

Figure 5.22: Mechanical Braking, Acceleration detect, Jerk-detect, UC trigger and monostable timers.

Here $1decel$ means that deceleration path becomes active and $1accel$ means that acceleration path becomes active.

$$trig = \begin{cases} UC_{decel} = 1, & if \frac{d(de_{1decel})}{dt} > 0 \\ UC_{accel} = 1, & if \frac{d(de_{1accel})}{dt} > 0 \end{cases} \tag{5.9}$$

The UC_{decel} and UC_{accel} then start the UC and MCC control. These 2 signals can be both active simultaneously, although generally offset in time, this does not hinder their performance since the controller gives preference to acceleration as discussed in sections prior. These signals activate the monostable timers.

The previous sections discussed the implementation of mechanical components and controls within the vehicle, following sections discuss the effects of environment on the vehicle.

5.2.13 Drag force

Objects in motion in air encounter dynamic resistance which depends upon several factors like density, temperature, shape of object etc. This resistance force is modelled as drag force or F_d. The equations for this are already discussed in earlier chapters. Here the implementation in the block-diagram is presented in fig. 5.23. This force can also depend on the directional of airflow with respect to a stationary reference such as ground, like wind. This is inculcated in the model.

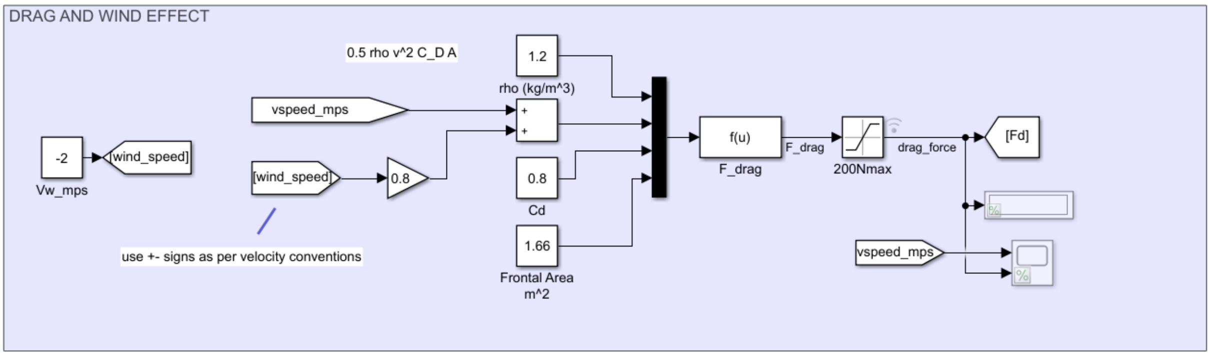

Figure 5.23: Drag force implementation along with wind-speed $v_w = -2m/s$ or in same direction of motion

The value of coefficients is apparent from the figure. It should also be noted that the frontal area is also mentioned in table 5.3.The drag coefficient is for a typical two-wheeler obtained from a standard chart. The output is saturated at 200N for simulation whereas in real-life can go up-to 2000N or 20kg.

5.2.14 Gradient Effect

The gradient effect concerns the mass of vehicle and the torque to be generated by the motor to keep vehicle in motion. The case for mass is kept extreme here; considering full load capacity operation at 300kg.

The equation corresponding to the block diagram in fig. 5.24 is given as follows;

$$Fg_x = sin(\alpha) * m * 0.1kg \tag{5.10}$$

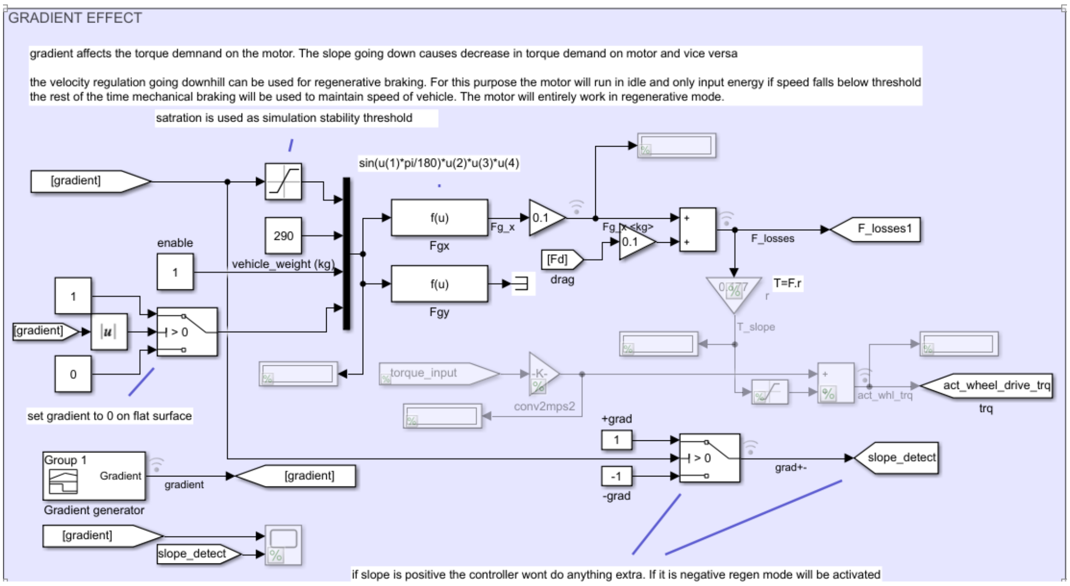

Figure 5.24: Effect of gradient calculation and overall loss computation.

$$F_{losses} = F_d + Fg_x \tag{5.11}$$

The first equation is simple resolution of mass a vector on a slope-of-angle or gradient α. Also $m = 300$ is the mass of the vehicle in kg'. The 0.1 at the end is for converting newtons to kg'.

Fig. 5.25 shows the gradient input to the block. Here it is fed directly. In an elaborate scenario this would be computed based on the distance covered.

Note that it is given only till 10 seconds. Since the gradient is considered 0 thereafter. The α varies between $[-1.6, 5]$ degrees.

5.2.15 Torque reference generation

The torque input on the motor drive is computed from the losses and the speed reference demanded by the user-input. This is computed and scaled in real-time and the gearbox consideration is taken in the simulation. Fig. 5.26 shows the block diagram for the torque input to motor. Also the equation for the same is given below as well.

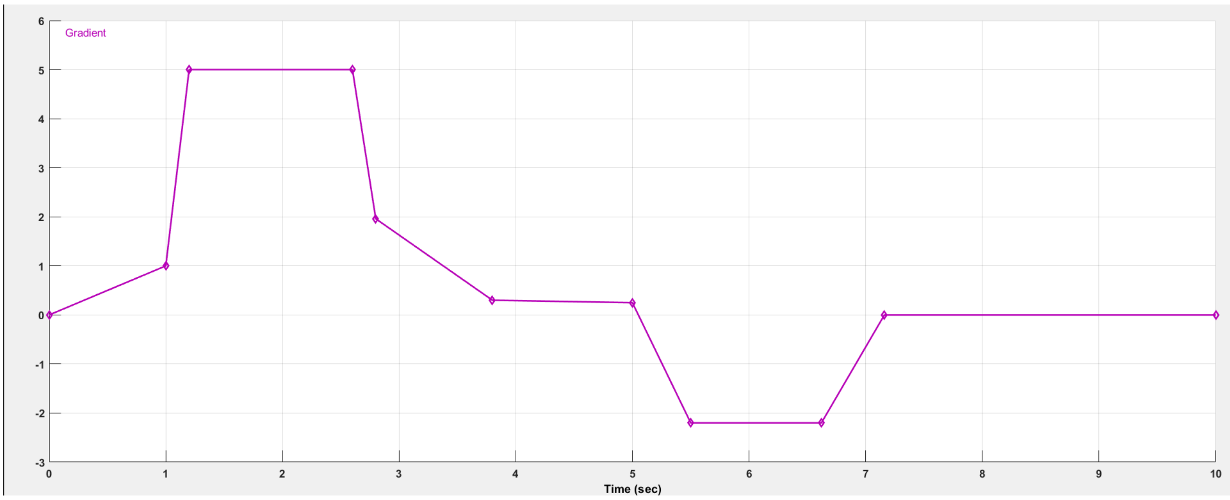

Figure 5.25: Gradient input as environment variable.

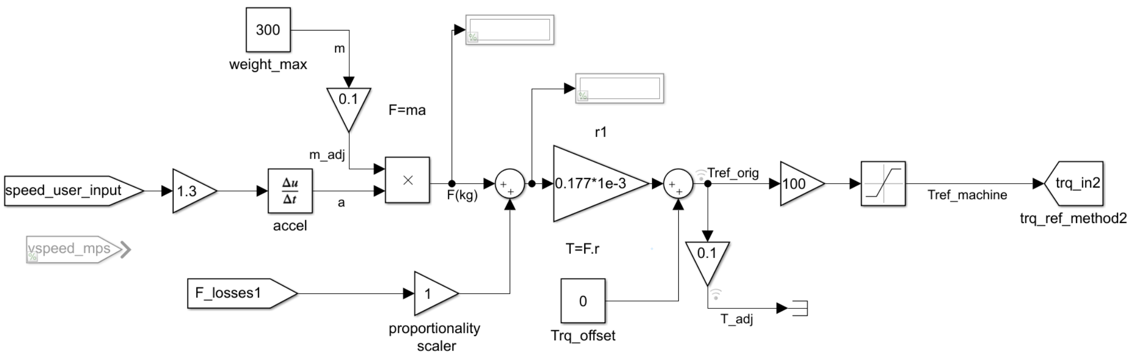

Figure 5.26: Torque reference generation for the PMSM drive.

The driving force is computed in real time here as well. Although in a real-world scenario the mass of the vehicle is not recorded in real-time and will thus be changed with alternatives which can be indirectly used to sense the torque load, like speed error and motor current draw.

$$\tau_{ref} = (((m*1.3*\frac{d(UREF)}{dt}*0.1)+F_{losses})*\frac{r_{tire}}{1000}+\tau_{offset})*100 \tag{5.12}$$

Here it also worth a mention that the characteristics of this motor drive are based on a real world TEV motor rated at 5kW nominal and 8kW peak power. The machine is thus restricted to 25Nm torque maximum and 20Nm torque nominal as per

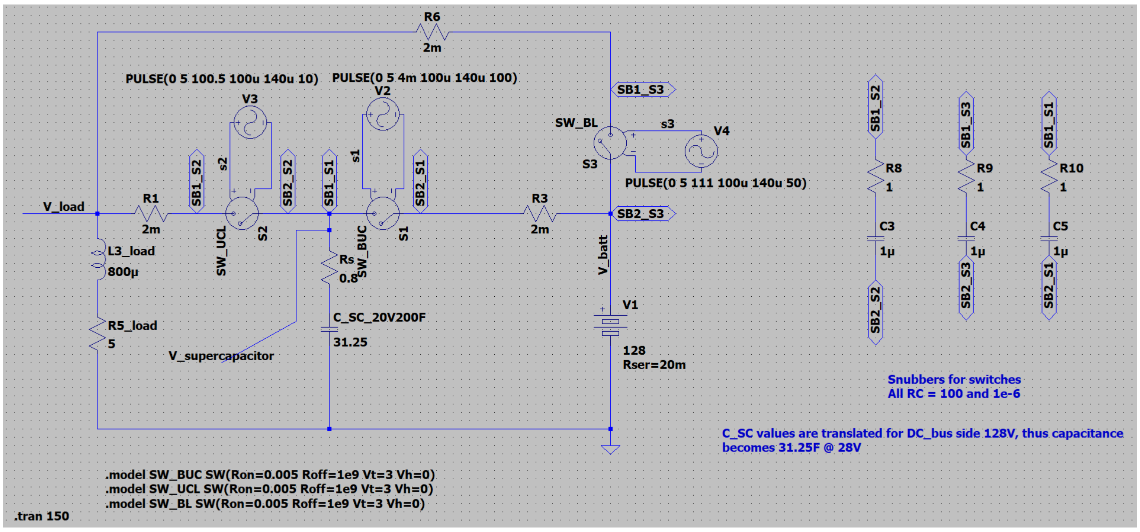

Figure 5.27: The schematic drawn in Ltspice software. The component values are written besides themselves, The three switches are operated using signal blocks in order S1 then S2 followed by S3. S1 is on for 100s to charge the SC. After this S1 is turned off and S2 is turned-on supplying power to load for 10s. Finally the S2 is turned off and with S1 being still off, S3 is turned-on, the load now receives power from battery directly.

the real world reference. Also the values for variables are the same throughout the model such as radius $r_{tire} = 0.177m$ and $m = 300kg$. Rest all constants are for scaling and control

5.3 SPICE setup

The simulation in LTSPICE corresponds to simulating charging and discharge of capacitor bank and also for switching transient on DC bus. The following figures show the setup and results generated.

The results of SPICE run are shown in fig. 5.28. The circuit diagram and SPICE directives are also shown in fig. 5.27. The value for capacitance is adjusted for 128V to be 31.25F. This is done to ensure that the absence of BDBBC does not affect the data. This process preserves the total energy stored in the capacitor or charge conservation. Since $200F * 20V = 31.25F * 128V$

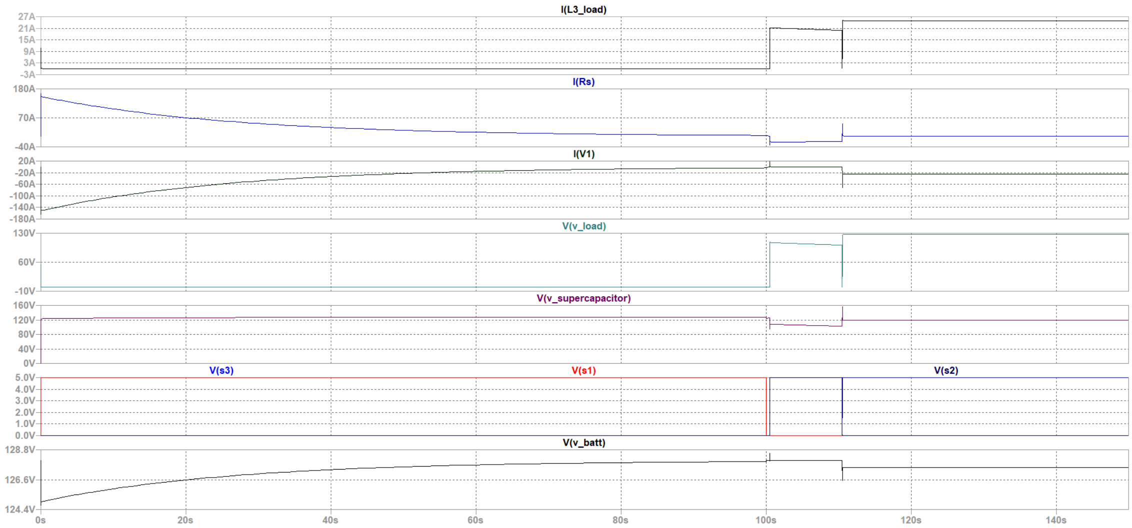

Figure 5.28: The data plot shows from top to bottom; Load current, SC current, Battery Current, Load voltage, SC voltage, switching signals for sources and SC charging (S1 - battery to SC, S2 - SC to load, S3 - battery to load), last is the battery voltage. Note that snubbers were used in the circuit suppressing any significant transients. This data is in contrast to MATLAB/Simulink presented in chapter 7 where transient-phenomena are significant.

5.4 Virtual Tests

The first test is the simulation itself being subjected to user input for speed, the second test is the range calculation obtained from a best fit scenario for simulating SoC down to 10

5.5 Stress tests

The stress tests are also integrated into the simulation, however are generally handled well by the control system. This input data does contain two scenarios corresponding to a realistic stress-simulation, where the user command shows acceleration but the brakes are also engaged. In that case the torque generation struggles and the velocity obtained as well. This is further discussed in the chapter 7 - *Results*.