



***INSPIRED*** **KARTERS**

DESIGN REPORT

# Table of Contents

1. [Introduction](#)
2. [Lap Time Simulation](#)
3. [Powertrain](#)
  - 3.1. [Motor Selection](#)
  - 3.2. [Motor Controller](#)
  - 3.3. [Electronic Torque Control \(APPS\)](#)
  - 3.4. [Motor Mounts](#)
  - 3.5. [Drivetrain](#)
4. [Accumulator](#)
  - 4.1. [Cell Chemistry and Model Selection](#)
  - 4.2. [Electrical Parameter Calculations](#)
  - 4.3. [Cell Configuration](#)
  - 4.4. [Battery Modelling](#)
  - 4.5. [Battery Management System](#)
  - 4.6. [Container and Module Design](#)
  - 4.7. [Charger](#)
  - 4.8. [Battery Aging Analysis](#)
  - 4.9. [Thermal Management](#)
5. [Safety and Control Circuits](#)
  - 5.1. [Central Network](#)
    - 5.1.1. [Drive Control Module](#)
    - 5.1.2. [Data Acquisition System](#)
  - 5.2. [Safety Circuits and Sensors](#)
    - 5.2.1. [Brake Pedal Plausibility Circuit](#)
    - 5.2.2. [Shutdown Circuit](#)
    - 5.2.3. [Pre-Charge - Discharge](#)
    - 5.2.4. [TSAL - AIL](#)

# 1 . Introduction

## Formula Bharat

Formula Bharat is an inter-collegiate competition which provides the opportunity for students from various universities across the country to design and fabricate a formula-style open wheeled race car. The event also gives the students a chance to contest with one another in areas of design, performance, cost analysis, business model, etc. Formula Bharat has a subsidiary event called the FSEV Concept Challenge

## FSEV Concept Challenge

This report is the combination of multiple design reports pertaining to the electrical package of the car, along with the development and results of a lap time simulation which was self developed

This report aims at taking the reader through the design process of the team Inspired Karters Electric. The report extensively covers the challenges and constraints encountered in the designing of the electrical package for the upcoming edition of Formula Bharat. Further, the team's solutions are also presented comprehensively.

## 2.Lap Time Simulation

### 2.1 Introduction

The design of a Formula Student Vehicle hinges on the ability to model what happens in reality beforehand to predict and modify the design to suit our needs. A lap-time simulator is a primary analysis tool that helps evaluate design and guide the whole process.

#### 2.1.1 Need for a Lap-Time Simulation

In an FSAE competition, the primary aim is to maximize the amount of points scored. Thus, the design of the vehicle and every design decision must be geared towards optimizing the amount of points gained. A lap time simulator is a simulation of the vehicle (which is abstracted into a mathematical model) as it goes around the racetrack. The primary output of a lap-time simulation is the time taken to complete the track, which is the primary parameter upon which points in all the events, especially Endurance, is awarded. So any design decision made by the team that directly affects the performance of the car can be quantified through the lap-time simulator in terms of the difference in Endurance time or acceleration time which in turn can be directly converted to points gain through the scoring rules(you can use the previous year event's scores to predict the score as the scoring is relative).Thus a Lap-time simulator is indispensable to every team.

It is useful to also decide which design improvement gives the most improvement of one's scores for the same amount of material resources, time and money involved. This is especially useful when a team is trying to improve on the previous year's design and wants to get "the biggest bang for their buck".An example decision would be whether to work for a 10kg mass reduction and thus spend extensive time in designing the car or reduce the designing phase and increase the time for testing thereby making the car more reliable in the event.This can't be directly compared with the points scored on track. This suggests that just lap time can't be an influencing factor while taking decisions. There needs to be some metrics which look Performance parameters and reliability to be able to understand the decision properly.

#### 2.1.2 Performance parameters

Although lap time is the main parameter through which scoring is done (Fuel/Energy efficiency is also taken into account), it is not the only performance parameter that one must be concerned with while designing. Although it is one of the main parameters, we must understand that the lap-time simulation is an idealized simulation which is a highly abstracted model of reality.

One of the major abstractions is the human element of the driver driving the vehicle. The lap-time simulation assumes a driver that can use the car's potential to the fullest and then produces the consequent lap-timing. This idealization means that the lap-time produced by the simulation is generally the best possible lap time that is possible. But we need to create a design that also takes into account the ease of driving, which cannot be directly captured by just a single measure of lap time. To understand all of the parameters of driving, we can look at the steering response-quantify the dynamic understeer and understeer gradient-thereby be able to quantify the ease of driving(there are multiple test/standards to also achieve this). This is just one of many things that Lap-time as a parameter doesn't capture. So there exists many performance parameters that although doesn't directly affect the lap time in the simulation, it affects the performance in reality. Therefore, the lap time-simulation must be extended to include simulations of other components of vehicles to give other performance parameters like the steering response to help guide design of those corresponding components. Thus, the extended lap-time simulation is vital in design decisions of every level: from the complete vehicle level to subsystem level to component level. Thus, the aim is to model the complete vehicle during the lap.

Levels of Simulation:

## Simulation Type Flowchart

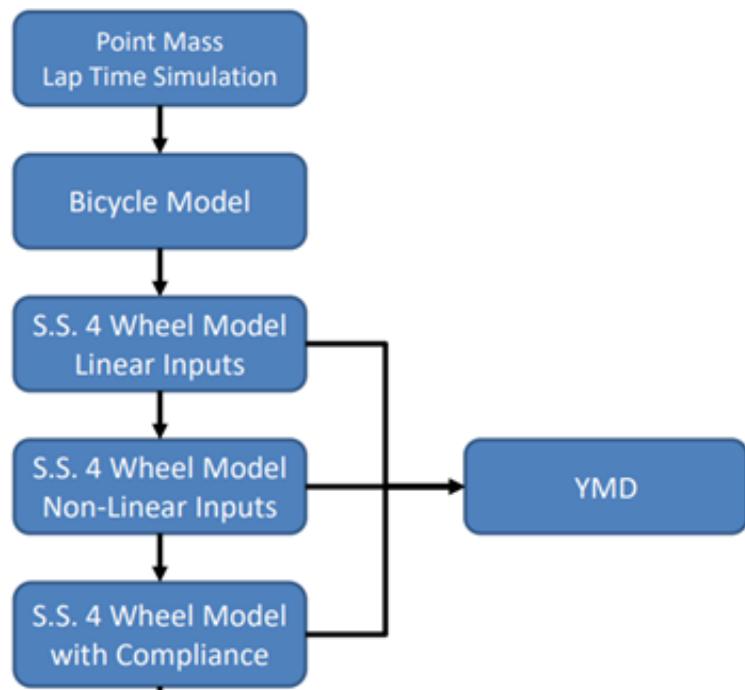


Fig 1 Different levels of Modelling [2]

The figure above shows the different levels of modelling a car with increasing complexity. As we go lower down, the complexity increases and returns in terms of ability to take design decisions for modelling decreases. Each model has a set of underlying assumptions that tries to capture a certain behaviour of the car and its corresponding influence on the performance parameters.

### **2.1.3 Motivation for Development of Indigenous Simulation**

The first model (the point mass model) assumes a point mass in the place of the car and simulates it around the lap. This is the simplest model and commercial(academic) software called Optimum Lap with this model is available to the team. This is a very good software that helps with making major design decisions and is able to predict the lap time with a deviation of 10% from reality. Despite the ready availability we opted to create our own full body vehicle lap-time simulation for several reasons:

1. Subsequent increase in complexity of modelling increases the accuracy of the lap time predictions.
2. OptimumLap is not an open source software-therefore it is difficult to derive the performance parameters which is very important to design a lot of components.
3. Various models can be further built upon a lap-time simulation -basically a full vehicle simulation-which needs the code to be built ground up.
4. OptimumLap is developed for a CV while we required a simulation for EV.
5. We found a bug in the Optimum Lap software which made it significantly overestimate the energy/power required. This led to serious issues with regards to battery pack sizing.

Due to these reasons, we decided to develop a lap time-simulation starting from the point mass model and expanding it with increasing complexity to guide various design decisions. Currently we have developed the model up to the level of the Bicycle model while incorporating components of EV vehicles like battery modelling and motor modelling. The following slides from Claude Rouelle [4] summarize the inputs, outputs and the design decisions that can be taken from the implementation of point-mass modelling and Bicycle model.

## Point Mass - Lap Time Simulation

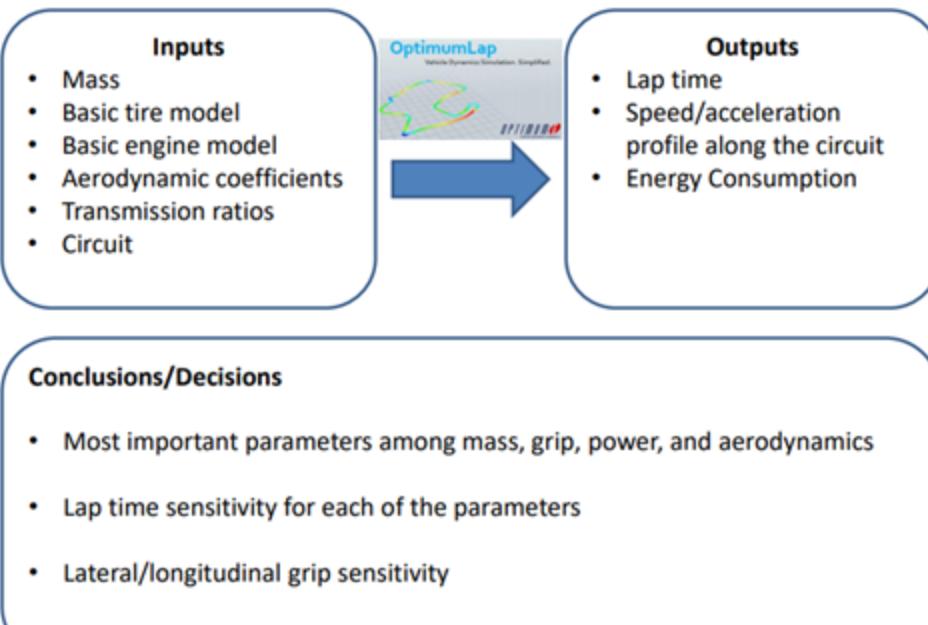


Fig 2 Point mass model [4]

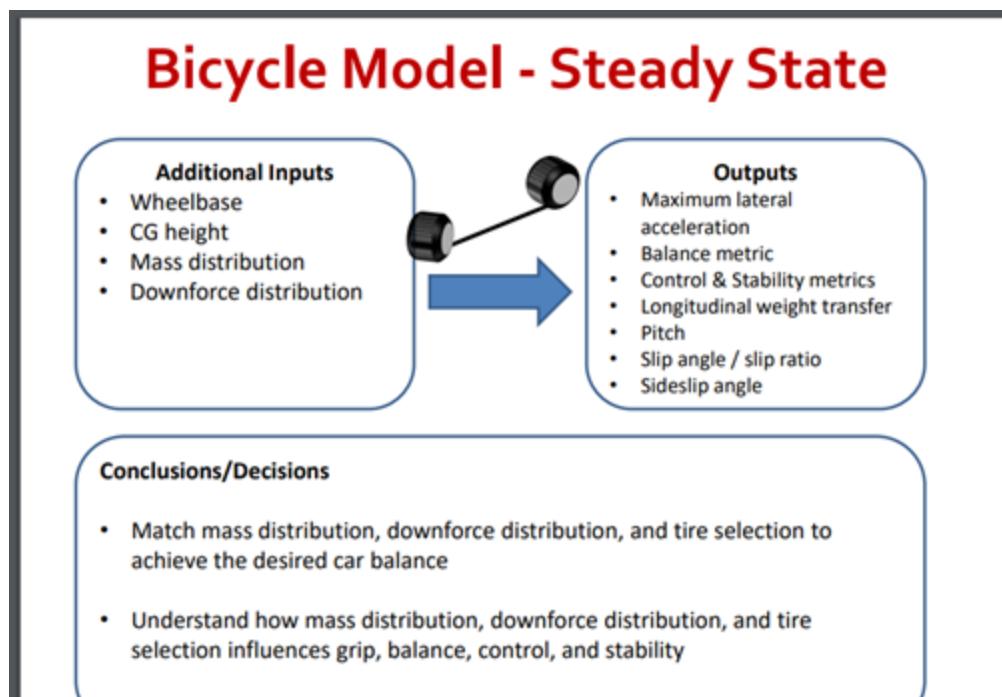


Fig 3 Bicycle model [4]

## 2.2 Implementation of Point Mass Modelling:

A point mass model contains the following assumptions/abstractions:

1. The whole car is abstracted to a singular point mass. Thus, we can directly apply Newton's laws to this point mass based on the forces acting on the car to simulate its motion around. This assumption is later relaxed in the bicycle and the 4-wheel model.

2. The effects of the following forces are considered:

- Traction Force from the Tire: This is the main driving force. This is determined based on the powertrain, drivetrain and tire characteristics (specifically the friction ellipse). The powertrain and drivetrain are modelled separately which constrain the traction force on the tire.
- Rolling resistance force on the tire.
- Braking force on tire -determined by the friction ellipse [3]
- Aerodynamic forces of drag which affects the vehicle's longitudinal movement and downforce which transforms the friction ellipse into the g-g-v diagram (a 3-D plot of gg diagram at different velocities)

3. Importantly it assumes that the car drives at its maximum capability. This means that the car has the ability to instantly change acceleration as required while being constrained by the g-g-v diagram. This basically removes the driver out of the equation and assumes a race-controller which has all required information and also has exact knowledge of the track so as to run an "optimum" lap around the track, thereby giving the best lap time.

An optimum lap is the lap that gives the lowest lap-time. The major vehicular constraint that dictates the performance is the g-g-v diagram. The g-g-v diagram [5] is basically the performance envelope of the car in question. It shows the limits to which the car can perform. It is experimentally obtained by extensively testing the used tire to generate tire data. This tire data must then be processed to then provide the final gg diagram.

The optimum lap is decided on the basis of the g-g diagram. Since acceleration can be changed instantaneously, it can be intuitively said that the optimum lap is the one where the vehicle is always at the surface of its performance envelope (the g-g-v diagram). This can also be mathematically proved. This forms the major basis for the design of the simulation.

The simulation basically tries to choose the acceleration on the g-g-v diagram such that it obeys the laws of physics that constrain it. In terms of the current simulation, the major constraint is the lateral acceleration required by the car while negotiating any turn. There are other constraints also that are incorporated. The discussions made in sections 2.3, 2.4 and 2.6 are based on a research

paper[2] by Tomas Novotny. Basically, the lap time model is based on this research paper with changes made according to our car and our requirements.

### 2.3. Understanding the concept [2]

Simulink is a software package that enables to model, simulate, and analyse systems whose outputs change over time. Such systems are referred to as dynamic systems and mathematically described in form of differential equations. The simulation is then performed by numerical integration methods (e.g. Euler) from a specified start time to a specified stop time.

From this point of view, the timeline could be understood as a kind of control input. However, the problem is that in case of racing simulation, the time value is also an unknown variable. In order to solve this issue, suitable transformation procedures have to be found.

So, it means to get the information of driven distance the velocity has to be integrated over a specified time limit, but finding this time limit is the objective of lap time simulation. In order to reach this period and hereby the real racing time, the integration process has to be stopped at the right moment. In this case, the stop-point is specified by the overall circuit length. When this milestone is reached, the final time can be read. (Fig4)

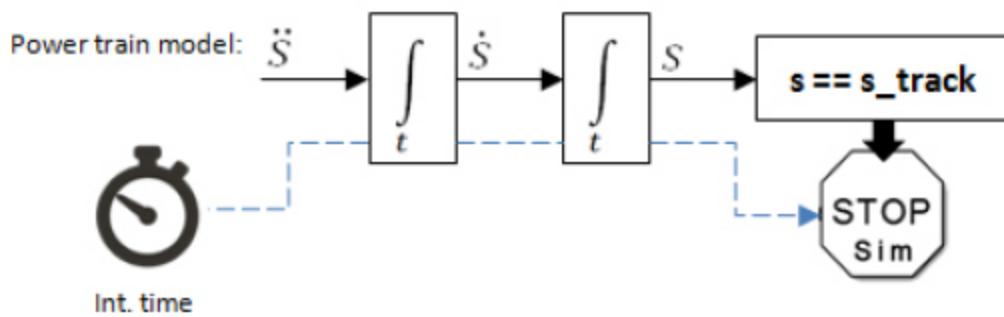


Fig 4 Transposition Method [2]

So, the time dependent input should be transposed to track dependent input which means that the vehicle controller could operate on the track information for example if it is known that there is a turn at 500m from start then the braking manoeuvre will automatically be performed at 500m. For this kind of a system a base velocity profile needs to be generated based on the maximum vehicle adhesion capabilities over the entire track. In simple words this profile will tell the operating region of the actual velocity and this operating region is based on maximum tire adhesion capability of the car and no other constraints.

So, based on this base velocity profile would be the final velocity profile which would be dependent on the adhesion limit as well as the powertrain of the vehicle. The figure below gives the schematic of the entire simulation process

No.	Calculation method	Input	->	Output
<b>1 Stage I Pre-calculation</b>				
	- Track analysis	[x,y]	->	[v_boundary]
	- Script boundary vel.	[x,y]	->	[1/R(s)]
		[1/R(s)]	->	[v_boundary(s)]
<b>2 Stage II Power train simulation</b>		[v_boundary]	->	<b>Results</b>
	- Controller	[v_boundary(s)]	->	[throttle/brake(t)]
	- Power train model	[throttle/brake(t)]	->	[x_dotdot]
	- subsystems			

Fig 5 Simulation outline [2]

As it can be seen in Fig 5 the first part is the track analysis which means converting the x,y coordinate system based track input to an array of curvature values at every distance step this implies that the value of curvature is known at every 1m (considering 1m to be the distance step) distance of track from a given starting point and is stored in the form of an array. So, a track of 500m will have the array length of 500 if the distance step is considered to be 1m. This is required because, to find a track dependent velocity profile the track data should be interpreted in the form which could determine the velocity at that point of track.

## 2.4. Pre-calculation[.]

In accordance with the simulation workflow at first, the trajectory is analysed. Then, comes the main phase with calculation of initial speed profile. Then the adhesion limits in the form of vehicle performance envelope will be discussed.

### 2.4.1 Track analysis

The functionality of track analysis will be explained on the path section between points  $P_i$  and  $P_i + \Delta$  that are both building the trajectory line. Fundamentally, the method is based on geometrical properties of red and blue highlighted triangles in. Both of them are determined by step delta  $\Delta$  (Fig 6).

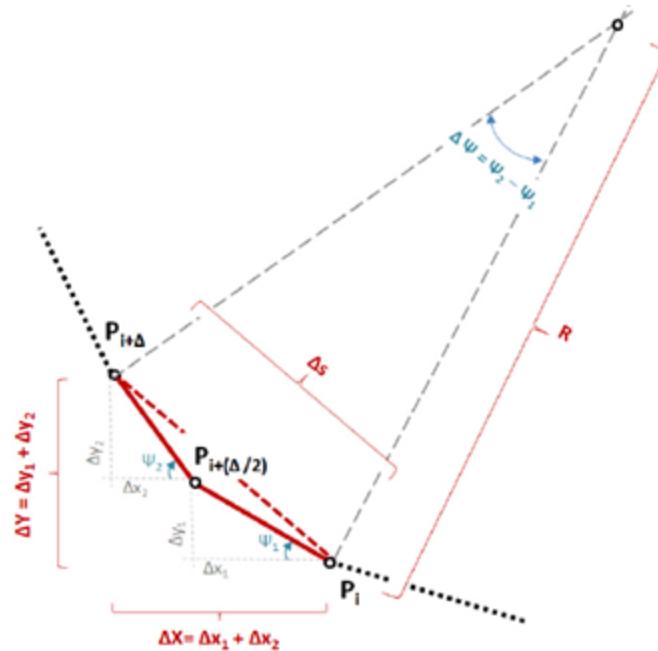


Fig 6 Track analysis [2]

The change of course angle  $\Delta\Psi$  comes geometrically from difference

between  $\Psi_1$  and  $\Psi_2$ . Therefore, the step size  $\Delta$  must be defined only as an odd number higher than 2 in order to guarantee a point in the middle for construction of two same sized elements that allow the identification of difference  $\Delta\Psi$ .

The values of  $\Psi_1$  and  $\Psi_2$  and the curvature value  $\rho$  could be found with the following relations.

$$\psi = \text{atan} \left( \frac{\Delta y}{\Delta x} \right)$$

$$\rho = \frac{\Delta\Psi}{\Delta s}$$

Fig 7 Mathematical relations [2]

Using different curve fitting tools and optimizing the step size the final track data could be generated in the required form.

#### 2.4.2 Base velocity profile

Speed profile generated represents the maximal feasible velocity that satisfies the adhesion limitation. This speed profile will be denoted as initial. Proposed calculation method comes from

a quasi-static approach investigating the maximal allowed speed for every track segment by the help of predefined vehicle performance envelope - the g-g diagram. By considering only adhesion limits, the applied performance envelope can be substantially simplified – the limitation of realistic accelerations will be simulated later on by the addition of the powertrain part. Basically, for initial speed profile, there is only two information needed – maximal lateral and longitudinal capability. The values of maximum acceleration limits corresponding to a given velocity were taken from the g-g table for the car; the table itself could be optimized by considering the aerodynamic forces into consideration.

#### 2.4.2.1 Initial guess

Without considering the acceleration and braking phase the maximal velocity at every point n of the track is constrained by the parameter Vmax (see fig 8).

$$v_{step1(n)} = \min \left( v_{max}, \sqrt{\frac{a_{y\_max}}{\rho(n)}} \right), \quad n = 1 \dots N$$

Fig 8 [2]

The aero downforce would increase the lateral capacity ( $a_{y\_max}$ ) for the car and therefore the acceleration itself is a speed dependent variable so the lateral acceleration was calculated iteratively at every point n of the track and the values updated in g-g table as mentioned earlier. A new velocity profile could be generated by using the values from the optimized g-g table. Figure 9 shows the initial guess of velocity profile with and without the aerodynamic considerations.

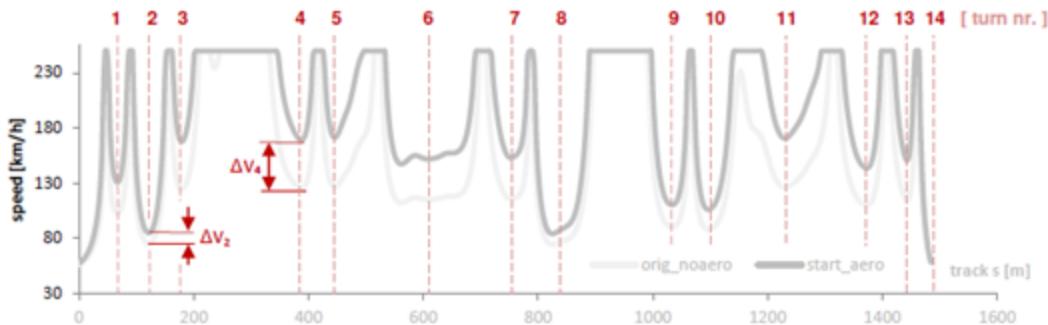


Fig 9 Initial speed guess [2]

The functionality is proved by the displayed speed difference between  $\Delta v_3$  and  $\Delta v_4$  in the above figure.

It can be seen that there are extremely high gradients (acceleration/deceleration) of velocity so now this profile could be optimized according to the maximum accelerating and braking capability of the car which basically limits the available  $a_x$ .

#### 2.4.2.2 Braking phase

The regions of interest are limited to the local minimums of the profile in fig 6 where initially the braking takes place till the exact local minimum point (intersection of red lines with graph in fig 9) and then acceleration. For this phase the algorithm could be run backwards because the points at which the car is decelerating are basically the points at which the car would be accelerating if we are moving backwards.

At the local minimum it is assumed that all the available grip is used for lateral acceleration and  $a_x$  is equal to zero ( $a_y = a_{y\_max}$  and  $a_x = 0$ ). If this point on track is considered to be  $S(i)$  then the velocity could be found at this point using the simple relation

$$V(i) = (a_{y\_max})^2 / \rho(i)$$

now using this velocity for the next distance step  $S(i-1)$  which means

$$V(i-1) = V(i)$$

the lateral acceleration could be found, now since the curvature value  $\rho(i-1) < \rho(i)$

The lateral acceleration value would be less than  $a_{y\_max}$ , therefore the remaining acceleration capacity is utilized in longitudinal direction and it gives the  $a_x$  value between the  $S(i)$  and  $S(i-1)$  step (or the braking/deceleration value if interpreted in forward direction). This  $a_x$  could now be used to calculate the final velocity at the point  $S(i-1)$  i.e.  $V(i-1)$ . The entire algorithm and functional realization are given in fig 10.

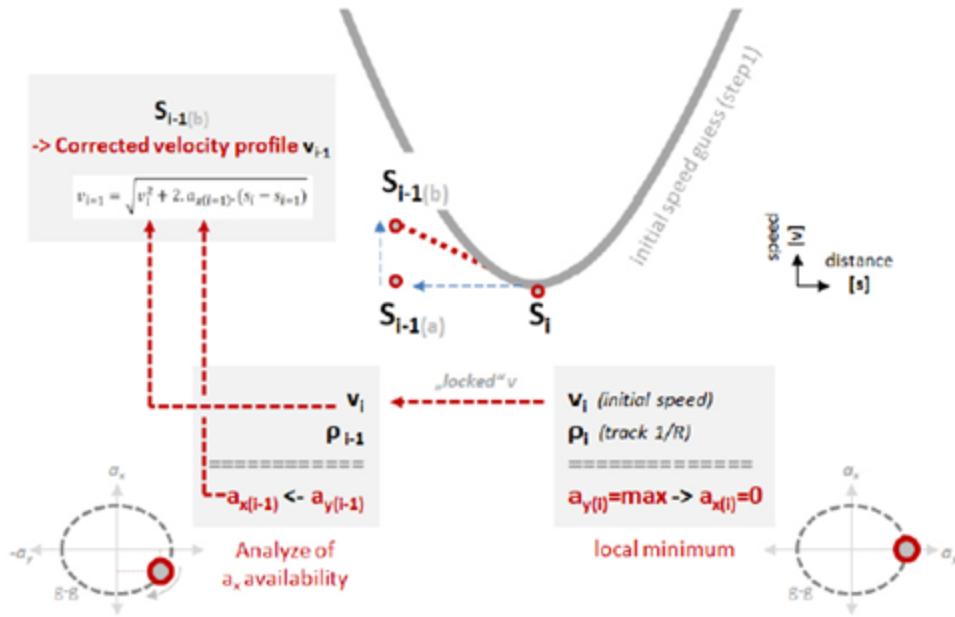


Figure 4.9: Braking Phase (Step 2) - Algorithm Realization

for  $i = 1 \dots N$

$$a_{y(i-1)} = v_{step1(i)}^2 \cdot \rho_{i-1} \quad (4.5)$$

$$a_{x_{brake}(i-1)} = \sqrt{\left(1 - \frac{a_{y(i-1)}^2}{a_{y,max}^2}\right) \cdot a_{x,max}^2} \quad (4.6)^{75}$$

$$v_{(i-1)} = \sqrt{v_{step1(i)}^2 + 2 \cdot a_{x_{brake}(i-1)} \cdot \Delta s} \quad (4.7)$$

Fig 10 Mathematical Calculations and visualization [2]

Same as earlier, in MATLAB script, the whole calculation process is modified for varying influence of aero down forces.

Running this script, the previously obtained velocity profile could be modified according to the braking phase (see Fig 11).

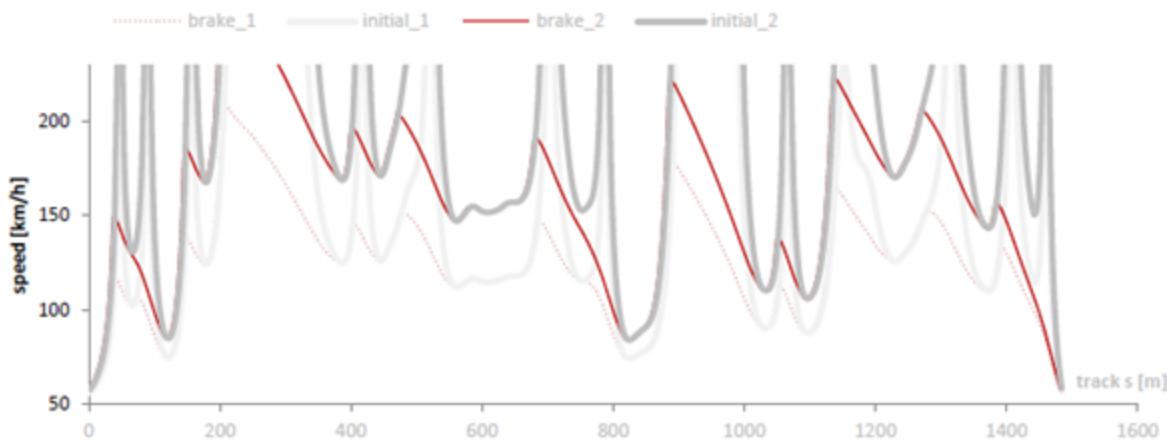


Fig 11 corrected velocity profile [2]

The dotted red lines are the profile without the aero downforce considerations; the final profile is the solid red line.

#### 2.4.2.3 Accelerating phase

Correction of initial guess profile for acceleration manoeuvre is based on the same principle as for the braking manoeuvre or the braking phase. The only difference is that instead of backwards calculation, this type of correction is processed by running the algorithm in forward direction. So, running the same script in forward direction gives the following profile as shown in fig 12.

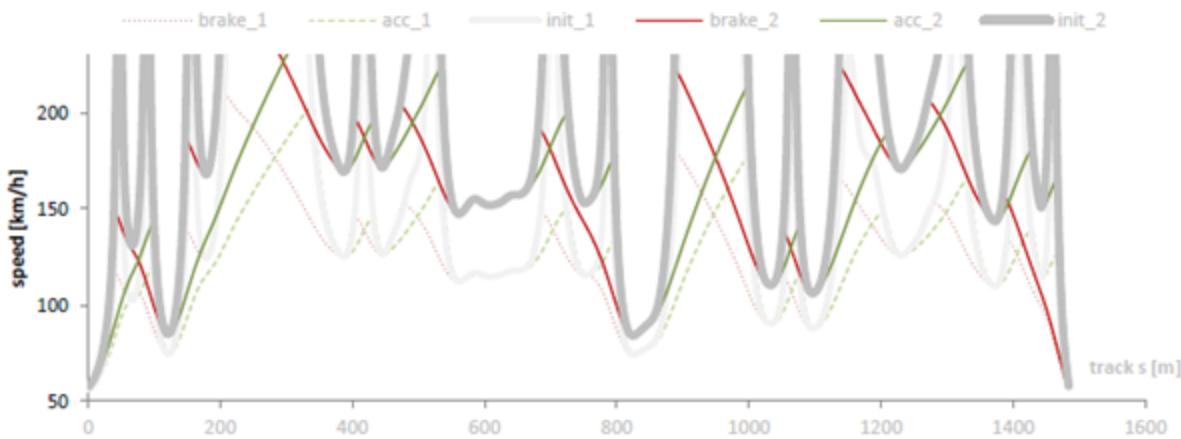


Fig 12 corrected velocity profile [2]

Using the optimization of both the accelerating and braking phase the final velocity profile based on the adhesion limits of the car could be obtained as shown in fig 13

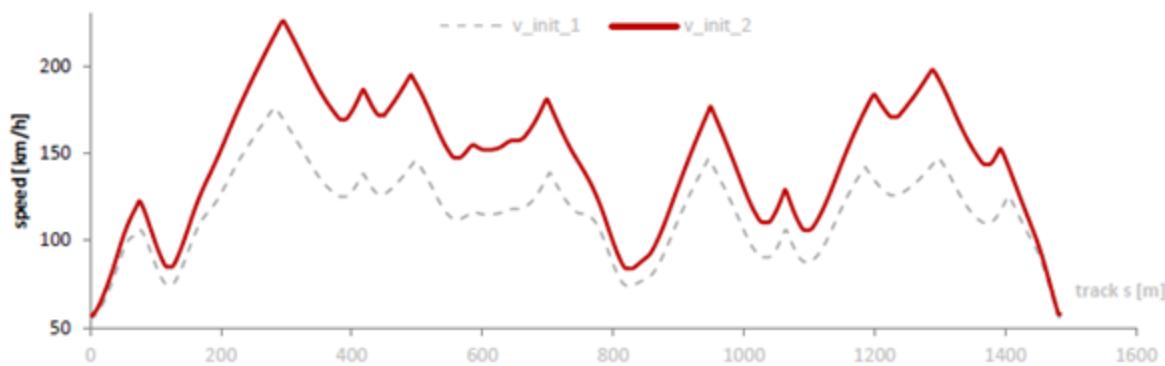


Fig 13 final velocity profile based on adhesion limits

The dotted profile is without the aerodynamic downforce consideration while the red profile is with the consideration of aerodynamics downforce and the maximum available grip of the vehicle. This Velocity profile and the curvature profile of the track will be the primary variables to be used for creating the lap-time simulation model and the bicycle model.

Before moving into further discussions of powertrain simulations and bicycle models, a brief description of a PID controller is necessary since it has been used in the models and is a crucial part of it.

## 2.5. PID controller

PID stands for the Proportional-Integral-Derivative control Fig 14 shows the elaborative display of a PID controller the three blocks (P, I and D) and the summation block together form the PID controller.

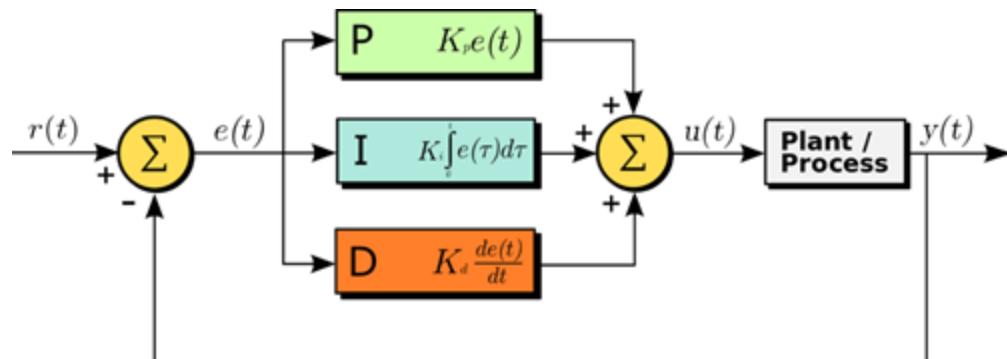


Fig 14 PID controlled model [1]

The model for which we want to generate input through the PID controller is called the plant. In the figure above the plant takes the input signal  $u(t)$  and evaluates it to calculate the output signal  $y(t)$  now if we have a reference signal  $r(t)$  to which we want our output signal  $y(t)$  to match with then a PID controller can be used to generate the input for the plant so that the error ( $e(t)$ ) between the reference signal ( $r(t)$ ) and the output signal ( $y(t)$ ) of the plant is minimum or zero.

This could be understood with an example: -

Consider the plant to be a motor and let  $u(t)$  is the current and  $y(t)$  is the output rpm of the motor generated now if we want the motor to run at a particular rpm say 500 rpm then initially with zero current the error will be of 500 rpm (reference 500 minus 0 rpm) then the controller in order to reduce it will give a new value of current (based on the proportional factor P) and again the difference will be taken and this process will be continued until the final 500 rpm is achieved.

The functionality of each part or constant (P,I and D) of the PID controller would not be discussed in detail however the basic functionality of each of them could be discussed. The proportional constant is what generates output with the help of error term and tries to reduce it. The integral and the differential constant keep a check on the signal to prevent it from exceeding from the correct value and reach the correct value in a controlled way.

## 2.6. Race Controller Strategy [2]

### 2.6.1 Concept

Till now our model is able to capture only the constraints imposed by the tire. The base velocity profile will be the velocity profile for just a single tire (with corresponding mass and aero forces) that is powered by an infinite power source. So, to actually capture the constraints of power and acceleration imposed by our powertrain, we need to refine our model. We could also capture other constraints such as longitudinal/lateral load transfer etc which also have an effect on the lap time. As explained before (Fig 1) this can be done by subsequent additions in modelling as required.

All of the above effects are dynamic and this is the reason that they can't be directly captured into our initial model. For example if our motor was to produce a constant torque at all rpm, we could just directly modify our g-g ellipse into a top cut off ellipse(impose a maximum longitudinal accel). To capture the dynamic effect(which means that the force/constraint depends upon the velocity/any other dynamic variable),we need to use the PID controller previously described.

## 2.6.2 Implementation:

The implementation of this is shown schematically in the Figure 15

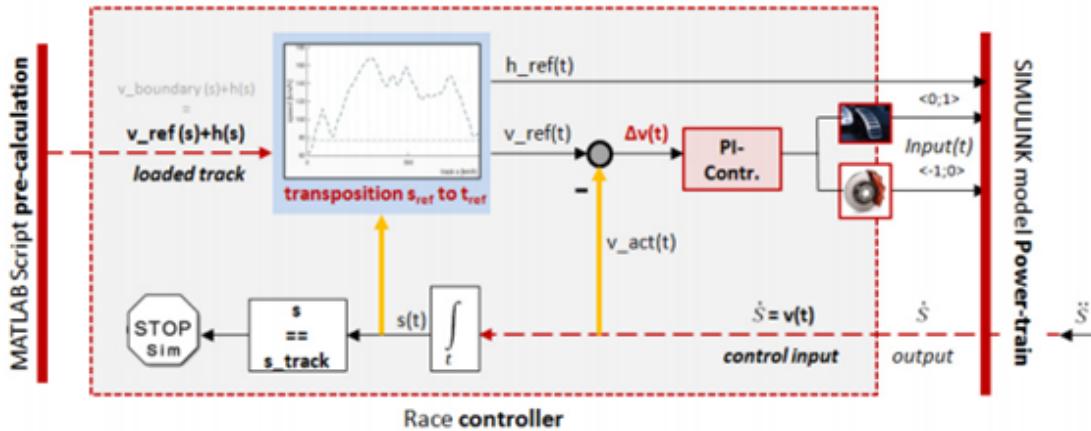


Fig. 15 Implementation of the PID controller to impose powertrain restrictions. [2]

This schematic is only for the powertrain restriction imposition, we could also (as previously mentioned) constrain the lateral acceleration based on load transfer etc. The output from the MATLAB script is  $v(s)$  which we will call  $V_i$  (for V ideal). The problem is directly complicated due to the fact that we have calculated the  $V_i$  as a function of distance along the track. This is unsuited for the dynamic nature we are trying to capture as the PID and the nature of the effects we are trying to capture works on the basis of time(also Simulink is a time-based solver primarily). So we need to first transpose  $V_i(s)$  to the corresponding  $V_i(t)$ .  $V_i(t)$  is the reference velocity that we want our controller to strive towards. There is a small quirk in this transposition of  $V_i(s)$  to  $V_i(t)$ . This cannot be done directly for this case and is not a direct transposition. The base velocity profile we generated ( $V_i(s)$ ) tells us what the best speed (given tire restrictions) is at a particular point in the track. A direct transpose wouldn't work as the actual profile will be much slower so the reference that a direct transpose gives wouldn't match with the required point on track.(For example let's assume  $V_i(s=2)=5$  and  $T_{ideal} @s=2$  is 4secs,now a direct transpose means that the reference is directly  $V_i(t)$  .For the actual run at  $s=1$ , $T$  will be 4 ,so the direct transpose would mistakenly take  $V_i=5$  as reference which is wrong.) To fix this a lookup table that takes in an  $s$  value and outputs the corresponding  $V_i$  value based on the function  $V_i(s)$  was used. So, this serves as the input signal for the PID error block

Now the PID is a tool that guides the actual velocity profile towards the reference velocity profile by taking their error as input and giving specific design parameters that would be input into a block which gives the actual velocity profile. The block contains the restriction/physical model you want to capture. It directly constraints the velocity through some equation. In our case it is

multiple complicated differential equations culminating in a first order Differential equation w.r.t. velocity i.e. Newton's second law for the car. So, the PID controller essentially controls the actual velocity profile by giving some design input, while at the same time conforming to the restrictions imposed by the Differential equations. The design input in our case is the throttle position (both brake and acceleration). So the PID controller here acts like a driver driving the car in the best possible way, trying to conform to Videal by giving the best possible throttle input to do so the Powertrain block will be in detail in the next section.

Now the actual velocity profile will be output (timestep by timestep: see working of Simulink). So, the actual velocity profile at any time step is then taken and integrated to get the distance covered. Now this is important because of 2 reasons:

1. Our ideal velocity profile is obtained with respect to track position. As explained the feedback loop is generated by taking this integrated position and feeding it to a look-up table. This output the  $V_i$  (current timestep) which is then used to check for the error input to the PID controller

2. We need to know when to stop. Since we are only generating a velocity profile and running the simulation in a loop (timestep after timestep), we need something to end the loop when the car completes the track. Therefore, a check is initiated and the simulation is stopped when  $s=s_{\text{track}}$ .

The settings of the PID controller are also important. The PID controller is really good for a linear plant. But depending on the complexity added (like bicycle modelling, motor modelling, etc), the plant can become nonlinear and the PID controller may not be able to handle the problem. In our experience for the point mass and subsequent bicycle modelling, we found that PI controller with a clamping function to stop the integrator windup worked really well and was able to simulate pretty accurately (Check the subsequent Validation section). The model also worked on just a P controller as well.

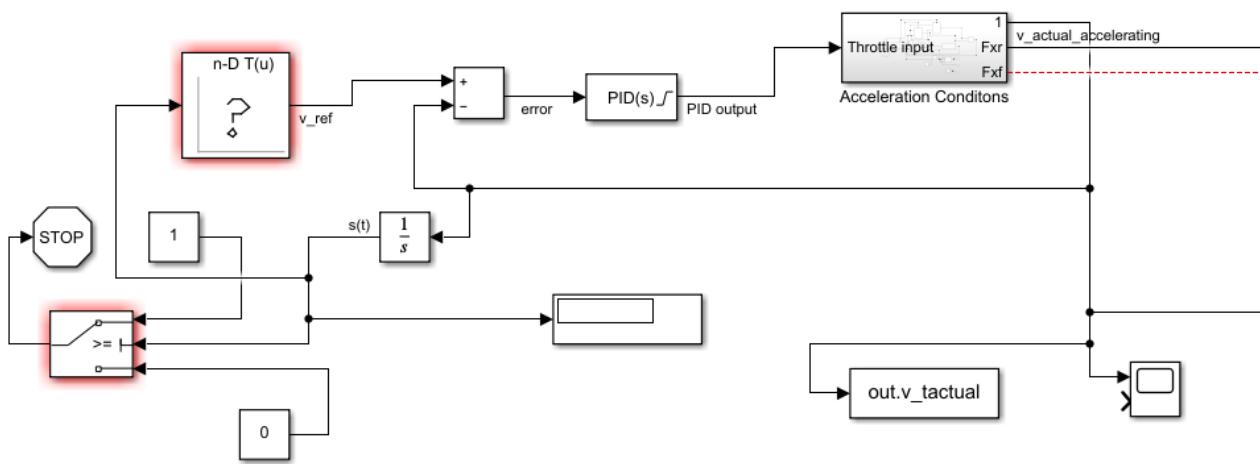


Fig 16: Snapshot of the Simulink Model

### 2.6.3 Powertrain Modelling:

The current model considers the effect of 3 forces on the caritative force, aero forces and rolling resistance force. All of these forces are considered in the longitudinal direction only. The point mass lateral dynamics was already taken care of in the base velocity profile. Any other influence on lateral dynamics can also be modelled (like the bicycle model).

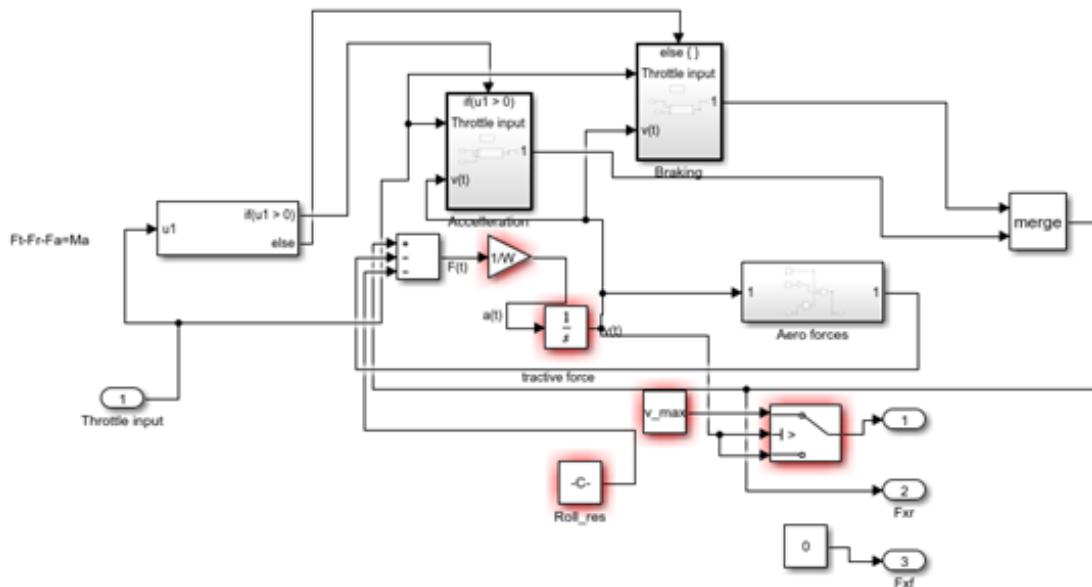


Fig 17 Overview of the Powertrain Block

So the tractive force is of main concern here as the rolling resistance and the Aero downforce and drag have already been modelled in the base-velocity profile (we still have to model it over here also). The tractive force can either be an accelerating force produced by pressing the accelerator or a braking force through the brake system. Both of these are separately modelled as two different blocks. The acceleration reactive force contains a model for the motor and a modelling of the drivetrain that the car uses. Electric drivetrain typically has a single gear system, so drivetrain modelling is easy. The braking system is also modelled and with the introduction of the bicycle model, we can predict the dynamic brake force distribution and its resulting effects also. Newton's second law is then constructed through these different forces and the differential equation is then modelled on Simulink. This is a first order differential equation with Vinitial set to zero. The actual model is intuitive to understand from the figure itself

All of the forces can be extensively modelled upon requirement and each of them can impose a constraint on the Lap Time simulation. For example, the battery can also be modelled to understand what would be the discharge cycle during a race and how it would affect battery health, the heat generated, any other battery related effects that could impose a constraint on the performance. But the decision of investing time to model depends upon the returns in terms of design. We believe a bicycle model gives us accurate results with 5-10% error in lap time which is acceptable at the current stage. The modelling of other components is mostly decided by the design utility it provides in understanding/predicting other performance parameters.

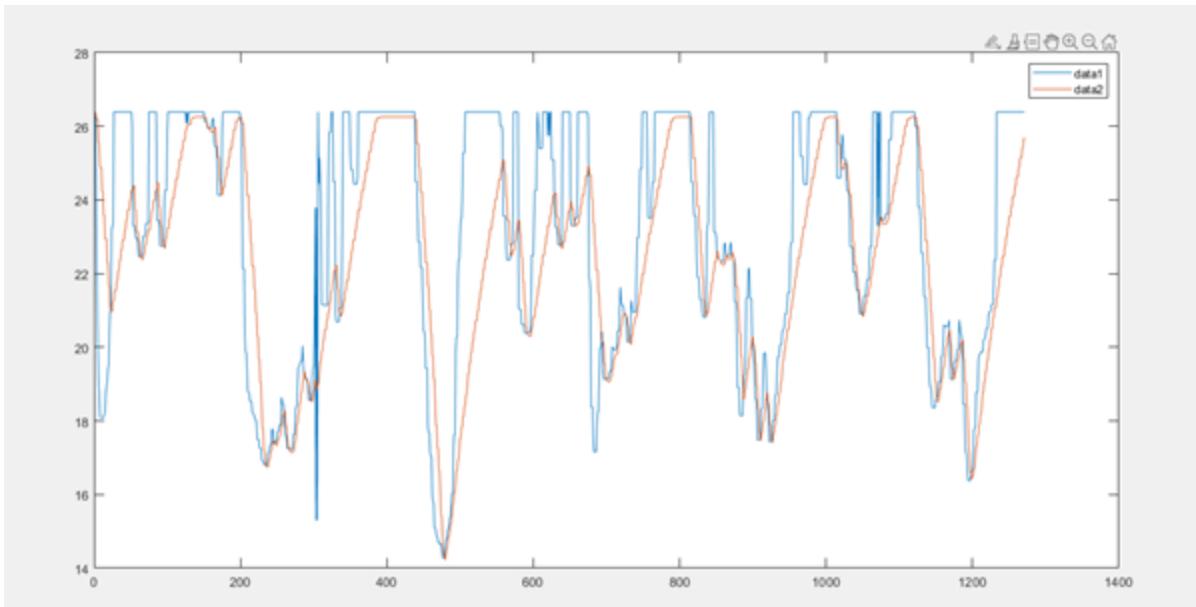


Fig 18: A comparison of base velocity profile (as a function of distance) and the actual velocity profile as a function of distance. The red line corresponds to the actual velocity profile and the blue line to the base velocity profile

Note 1: The initial velocity is not zero because this was simulated for the second lap of endurance.

Note 2: For the most part, the red line stays under the blue line which shows its conformity to the constraints

## **2.7. Validation:**

The validation of the Lap-time simulation was carried in two methods. The first method was the comparison of the results obtained from our Lapsim across the commercial software of Optimum Lap for a point mass model. The second method was to use our previous car's DAQ data to compare a real time run across our simulation. The former one tried out and the results were generated for different tracks from the optimum lap software and our lap time simulation model. The results matched satisfactorily however the second method could not be done yet due to the problem with a proper DAQ data of the vehicle and challenges such as extracting the track equation of the path that vehicle covered and also the conversion of many other parameters such as the velocity and acceleration, in the form which could be compared with the lap sim results is still under progress.

## **2.8. Bicycle Model**

As discussed earlier the results given by a point mass model have some limitations of its own and also it has some error of around 10% in the output. So, it is in interest of the team to upgrade to a bicycle model from a point mass model. The process of updating and improving a model is a continuous and long process because apart from bicycle model there could be huge scope of improvements leading to a better design but bicycle model gives a major breakthrough for this since, it incorporates many vehicle parameters which are necessary for the inception of a vehicle design, such as the wheelbase, track width measurements cg location and many other parameters. So, the first task is to build a bicycle model and then incorporate it into the lap time simulation model to view the effects of those vehicle parameters on the lap time. The bicycle model itself can provide many information such as the steering profile, yaw profile etc.

### **2.8.1. Simulink model**

Modelling such types of systems on Simulink gives the developer a huge advantage and makes it very easy for him. Simulink has predefined block sets which are basically a mathematical model in itself confined to an input and output block. These blocks work on the basis of all the required equations to work upon the given input and produce the desired output so what a developer has to

do is to give the required input to the block in the correct form and for that the knowledge of the equations being used in the block is paramount.

In this case the block used is a part of the vehicle dynamics block set library of the Simulink which is the vehicle body 3DOF single track block (fig 19)

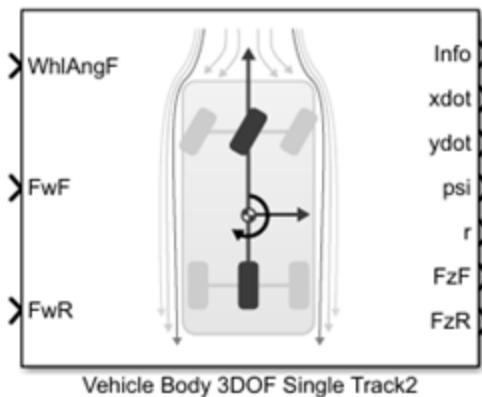


Fig 19 Simulink block

Single track means that the tires across the axle have been compressed and considered as one tire which is basically the bicycle model of a vehicle. It can be seen that the block demands three inputs and gives several outputs. The equations used by the block and how it uses the inputs to do the calculations could be seen in the documentation of the block. The **WhlAngF** is the front wheel steering angle of the vehicle while the **FwF** and **FwR** are the tractive force on the front and rear tyres respectively. The output port **xdot** represents the longitudinal velocity and **ydot** represents lateral velocity, **psi** is the yaw angle and **r** is the yaw rate, **FzF** and **FzR** are the vertical forces on the front and rear tyres respectively

The block requires many parameters which are user defined such as the cg location, wheelbase, track width length, tire cornering stiffness etc. The effect of these parameters could be seen directly by noting the change in outputs of the block with the change of these parameters. The block settings could also be varied to include external parameters such as wind etc.

So, the block requires two basic inputs to function the tractive force  $F_x$  and the steering profile. The generation of tractive force  $F_x$  is already discussed in (2.6) so the primary objective is to obtain a steering profile for/from this block. Before moving into further discussion, we should note that from earlier discussions we already have the track information on the basis of curvature and the velocity profile all over the track which will be required for the profile generation. The steering angle or the steering profile could be of two types one is the kinematic steering and the other one is dynamic steering. Kinematic steering means that the vehicle is taking constant velocity turns and the steering angle could be obtained based on the curvature and the vehicle

parameters only using the mathematical formulae for the Ackerman geometry. Whereas the dynamic steering could involve either acceleration/deceleration or constant velocity turn this means that it would be dependent on more than just the curvature and vehicle parameters.

Initially equations of vehicle dynamics were used which could give the steering profile using the lateral acceleration (was determined from the velocity profile and curvature profile) and the vehicle parameters but the block itself uses those equations considering the steering profile as the independent variable (hence the input). Using the same equation to produce steering profile considering it as the dependent variable gave errors and wrong profiles because what we were doing was using a pre assumed value of output based on the information we have (velocity and curvature profile) and using that we were calculating input which in turn would give the output which is wrong. To check whether a profile is correct or not the output longitudinal velocity could be compared with the reference velocity profile, also the steering profile could be plotted with the curvature profile and it could be checked whether there are non-zero values of steering angle corresponding to non-zero curvature points. These comparisons were not satisfying for the above case.

So, it was concluded that the profile could not be pre generated and given as input. This means the steering profile generation would be generated simultaneously with the output and this could be done with the help of a PID controller. What a PID controller does has already been discussed earlier. So, the idea is that using some controlling/reference parameter or profile made using the already available velocity profile and the curvature profile the steering profile could be generated.

Since the PID controller works in a loop and the two inputs of the controller are the reference profile and the output port of the block, this implies that the reference profile could be of a yaw rate, yaw angle, longitudinal or lateral velocity which are basically the name of output ports of the block. Fig 20 shows a schematic of how the model would look if we could give a reference lateral velocity profile.

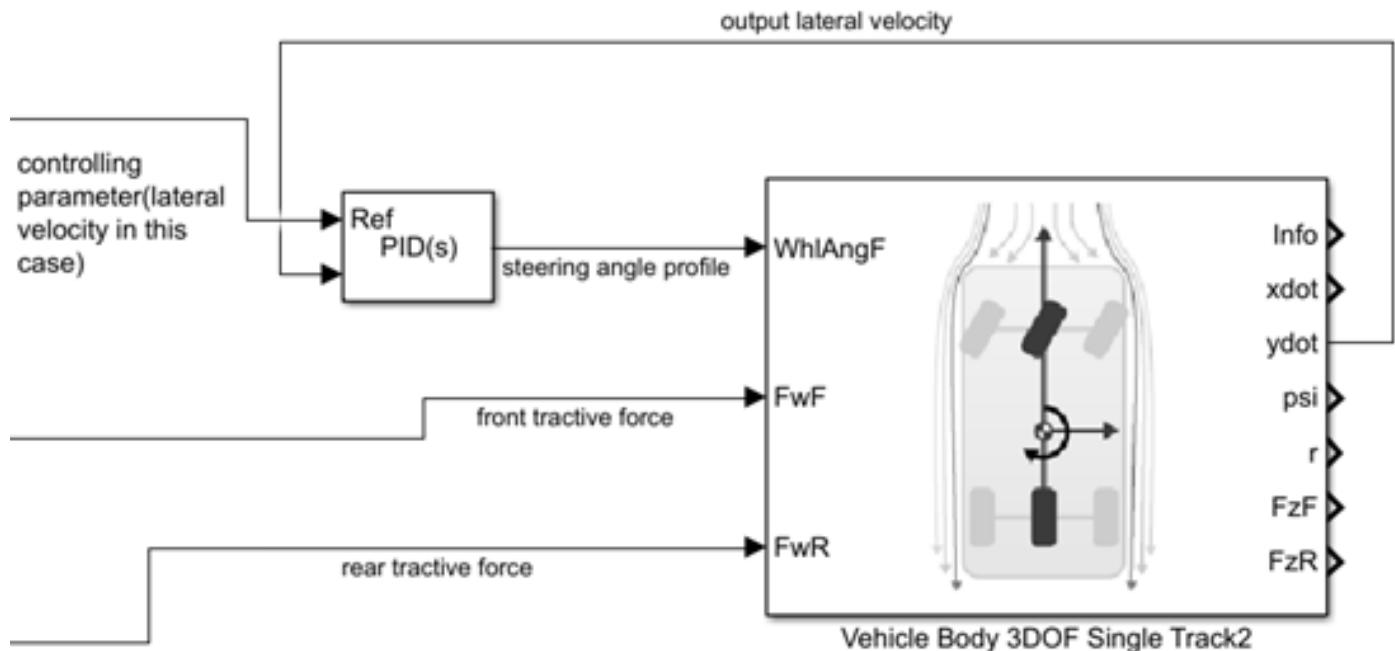


Fig 20 PID controller model

In the discussions further the parameter which controls the PID output would be referred to as the ‘controlling parameter’ for example in fig 2 the lateral velocity would be called the ‘controlling parameter’.

The PID controller has limitations of its own and also all the parameters could not be pre-determined using the given velocity and curvature profiles so these two facts shape the opinion on which parameter should be used as the controlling parameter. To find out the parameter another model was built as shown in fig 21.

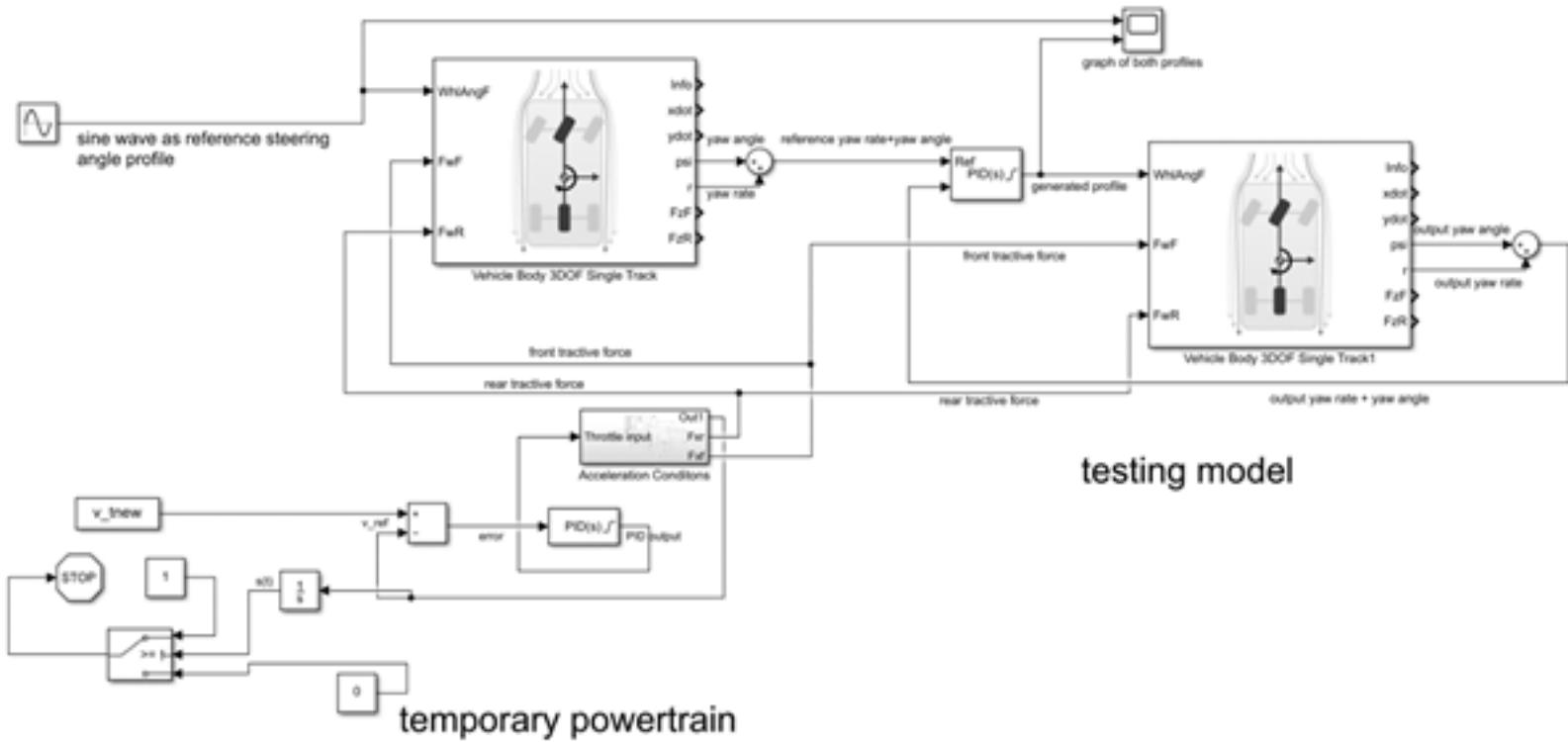


Fig 21 Testing model

The lower left portion (till acceleration condition block) of the figure above is a temporary framework from which the tractive force could be determined. The main testing model however is the one above it.

Model explanation – two ‘vehicle body 3DOF single track’ were used; the left block basically completes the left half of Fig20 which means that this block has been used to generate a controlling parameter profile for the block on the right. The tractive force input for both the blocks is the same. The entire model is based on the fact that if a known steering profile is given for the left block (sine wave in this case) and the left block’s output are used as the reference/controlling parameter for the right block then the steering profile generated for the right block should be same as the known profile.

The figure above is the final model in which the sum of yaw rate and yaw angle has been used as the controlling parameter however there were many iterations performed before arriving at this final model and to conclude that this could be the controlling parameter. So, one by one the graphs were generated for different controlling parameters and the one giving the profile closest to the sine wave was taken. The graphs for each one is discussed below. The PID controller i.e. the blue line represents the generated steering profile for each of the following figures.

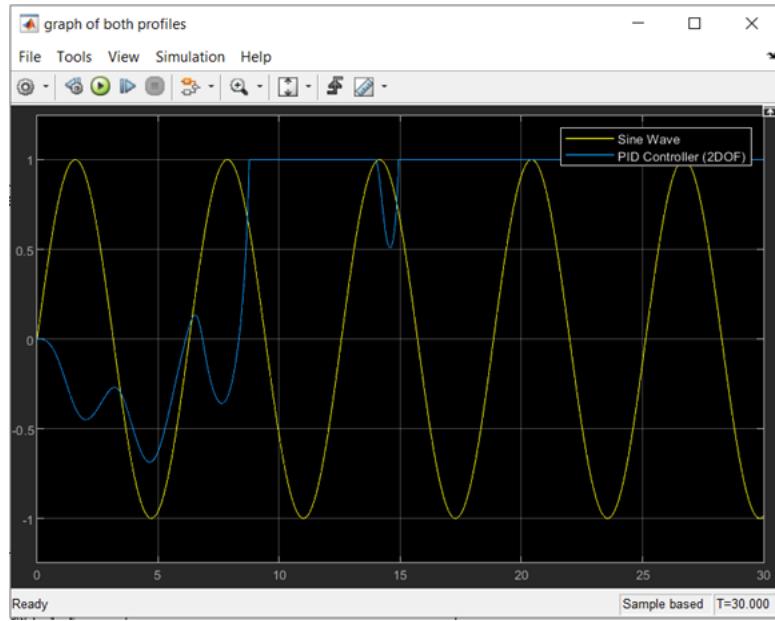


Fig 22 profile based on longitudinal velocity

As can be seen from figure Fig4 that there is no match of the profile generated if the longitudinal velocity is taken as reference, the possible reason being that this output is directly dependent on the other input and that is the tractive force. So longitudinal velocity could not be used as the controlling parameter.

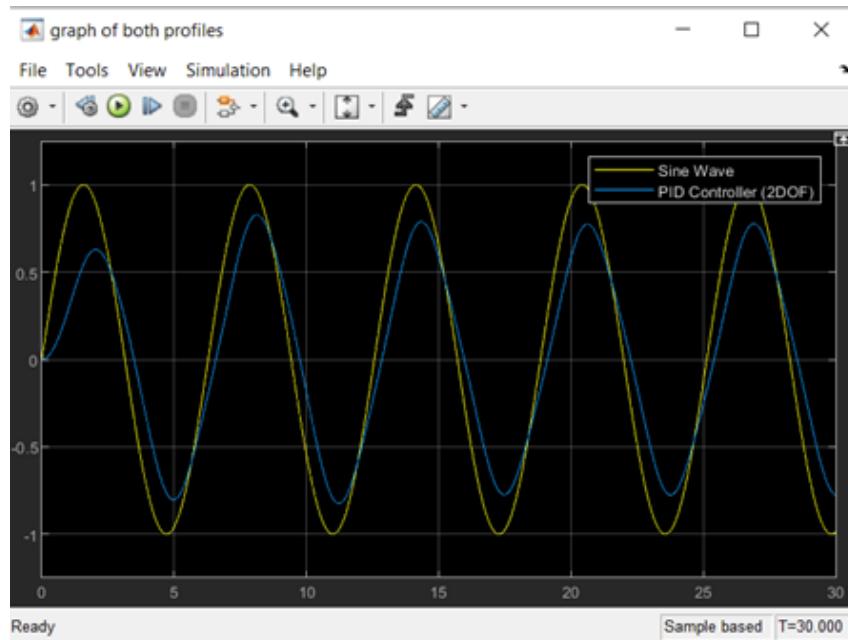


Fig 23 profile based on lateral velocity

Using lateral velocity gives a waveform similar to the sine wave (yellow) however the peaks don't match and the profile is still not exactly the same. Also, there isn't a way to generate a predefined lateral velocity profile based on the already available longitudinal velocity profile and curvature profile that limits the choice to the yaw rate and yaw angle.

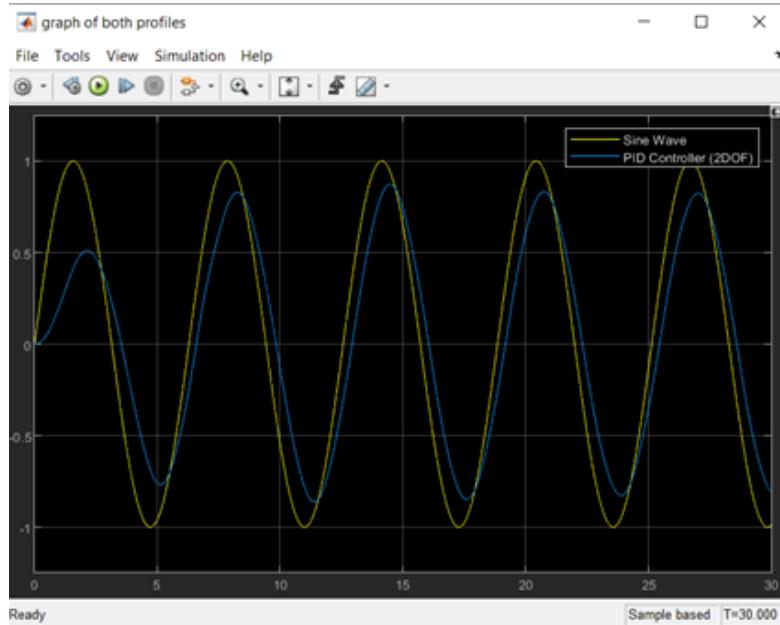


Fig 24 profile based on yaw rate

The profile based on yaw rate is pretty much similar to that of lateral velocity; it matches with the waveform but does reach its maximum point

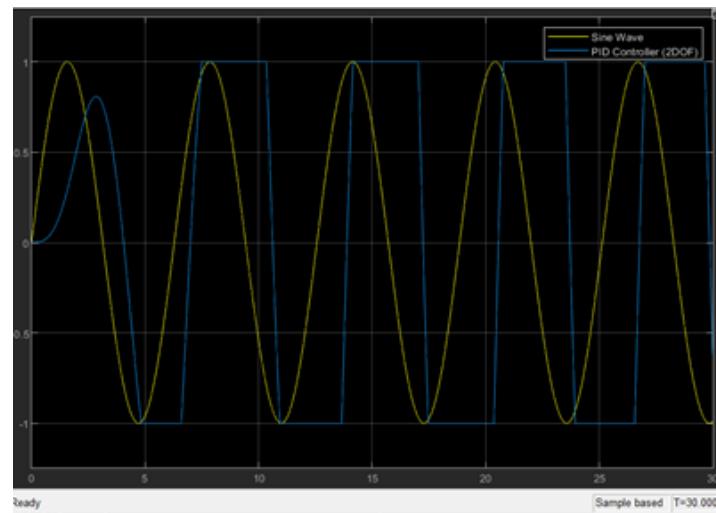


Fig25 profile based on yaw angle

This profile gives a wave form and reaches the maximum point of the sine wave but is not well defined or similar to that of sine wave.

Since the lateral and longitudinal velocity were out of choice and the yaw rate and yaw angle individually didn't give a proper profile for yaw rate the extreme points were not reached while for yaw angle the profile was deviated. So, the next iteration was to incorporate the effect of both. At first their product was taken but that generated a completely distorted profile so the sum was taken and the results generated were quite good as compared to the ones discussed above see fig 26

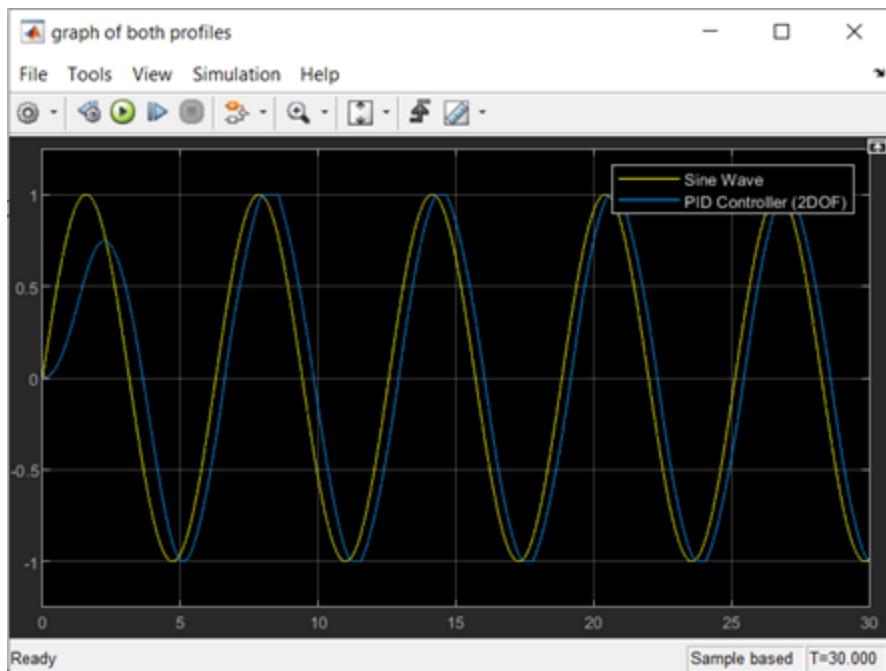


Fig 26 profile based on the sum of yaw angle and yaw profile

As can be seen in figure 26 the profile generated almost matches with the sine wave which implies that the sum could be used as the controlling parameter. The same process was done with other inputs apart from the sine wave and yet the sum of yaw angle and yaw rate generated the best profile.

The profile could further be improved by tuning the PID controller Fig 27 shows the profile generated when the proportion constant of the PID controller was significantly increased.

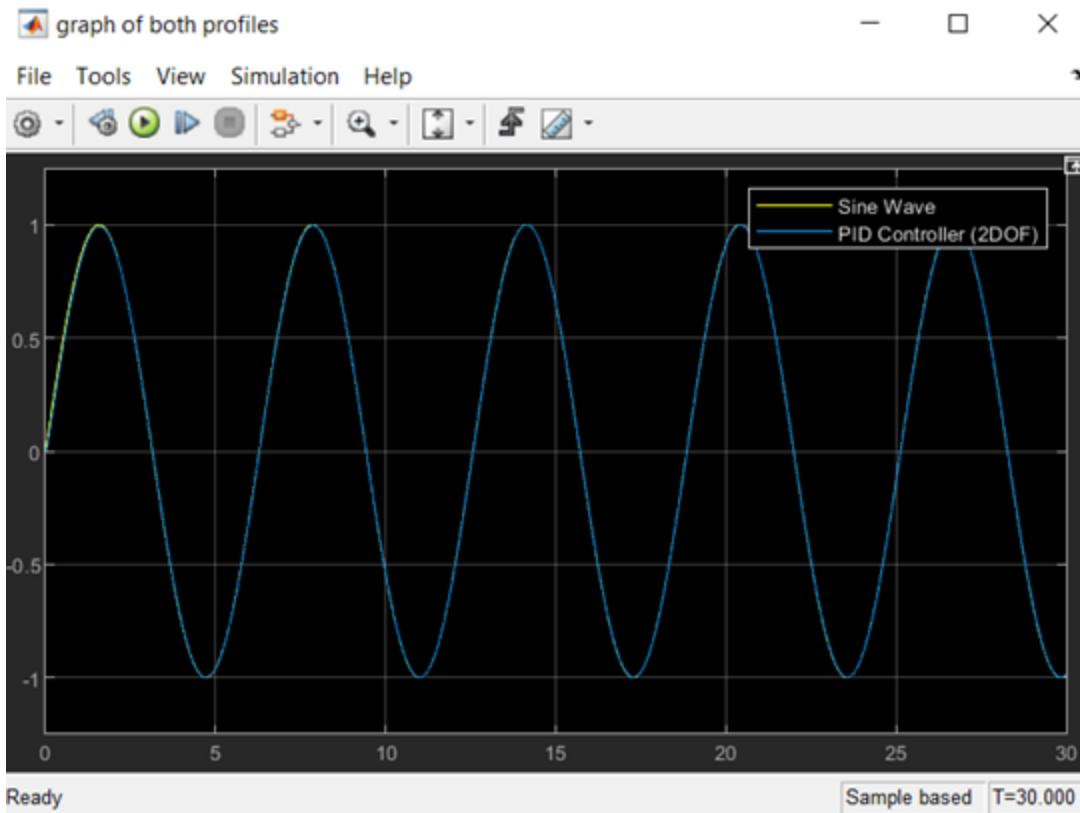


Fig 27 Tuned PID output

It can be seen that this change gave the exact overlapping profile for the steering. Tuning the PID improved the profile for other controlling parameters also but the sum of yaw rate and yaw angle was chosen since it gave the best output earlier and also the lateral velocity profile has limitations of its own that it could not be generated with the available variables.

Now the next task was to generate a reference profile for the yaw rate and its simple integration would give the yaw angle profile. As discussed earlier we have the velocity profile all over the track and its curvature values. Through simple geometry and calculations, it could be proved that angular velocity of the vehicle at any point of the track is equal to the yaw rate of the vehicle at that point. So, if  $v$  is velocity of vehicle at any point and  $\rho$  is the curvature then the angular velocity  $w$  could be given by

$$w = v * \rho$$

this  $w$  or the angular velocity is equal to the yaw rate ( $r$ ) of the vehicle this implies the yaw rate could be easily found with the simple multiplication of the already determined velocity profile and the curvature profile this yaw rate could be integrated to find the yaw angle ( $\psi$ ) profile.

A model was built on the basis of some hypothetical track data and the velocity profile was generated as discussed in (...) and this was used to generate the yaw rate and yaw angle profile. The data was such that there were constant velocity turns and one non constant velocity turn. Fig 28 shows the schematic of the model.

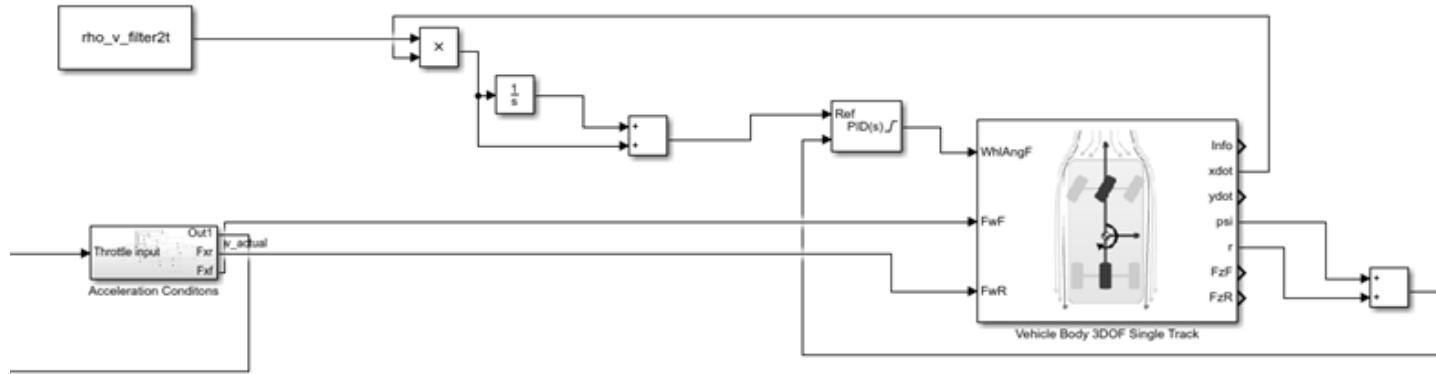


Fig 28 Test model

The block named rho\_v\_filter2t is the time dependent curvature obtained using transpose functions. The tractive force for the reference velocity profile has been generated and using the longitudinal velocity the yaw rate and yaw angle profile has been generated for using their sum as the controlling parameter the block x is the product block and the 1/s block is the integral block.

The results of this block are shown in fig 29

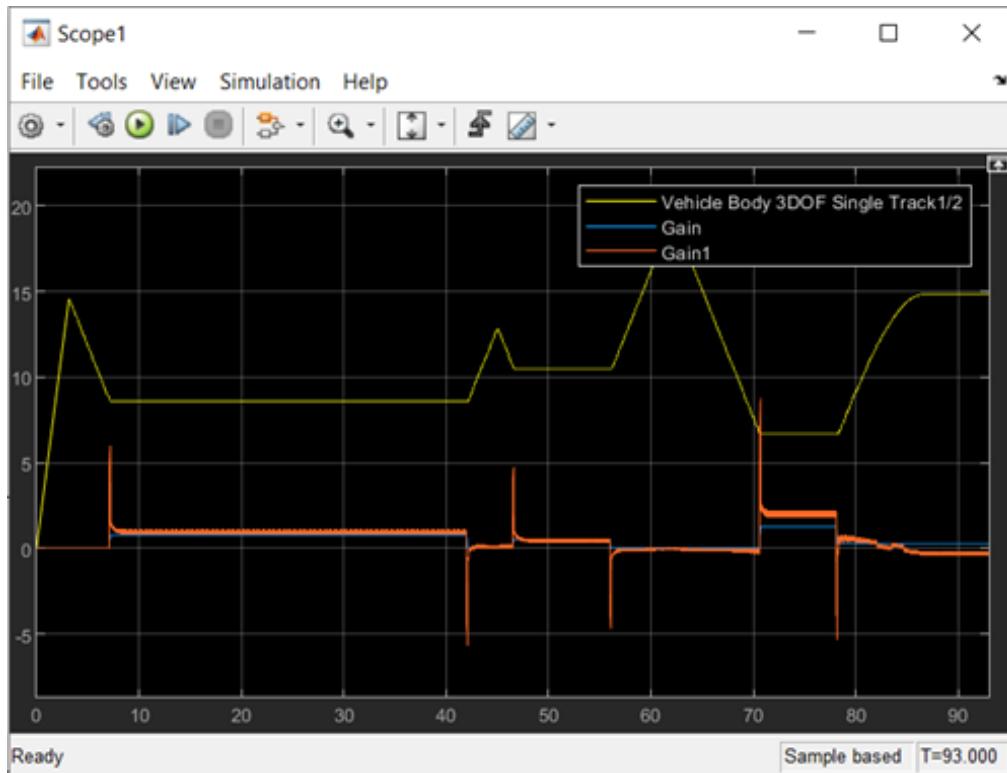


Fig 29 velocity, curvature and steering angle profile

The yellow line represents the velocity the blue line represents the curvature and the orange line represents the steering angle profile, the curvature and the steering angle have been multiplied by 50 in order to make it visible with respect to the velocity profile. It can be seen that at the non-zero curvature with constant velocity the steering angle is also constant while at the end where the velocity is not constant the steering angle changes accordingly. The spikes are due to the sudden change in acceleration at the end and starting of the turns (they are exaggerated due to a gain of 50) which suggest that while entering a turn the steering angle should increase rapidly (therefore positive spike) while at the end of turn it suggests a slight counter steer (therefore negative spike) and then zero steering angle.

### 2.8.2 Work to be done

The bicycle model discussed above has still many scope of improvement since it is not clear that whether the steering profile at the end should be the way it is and the model has not been tested with real track data and velocity profile where there are numerous sharp turns either at constant or accelerating/decelerating velocity, also significant amount of changes have been done on the powertrain side, since for the entire model to function it is necessary for the controlling parameter and the tractive force to be in complete agreement with each other which means that the reference profiles should be achievable with the given tractive force input otherwise the PID controller

would give distorted profiles (such as the spikes in Fig 11). So, the model is still under progress and needs to be tested with the tractive force inputs of the final lap time model to ensure correct inputs for the block and the PID controller. Also, apart from analysing the steering profile graph many other results such as the yaw profile, lateral velocity profile etc of the output could be analysed individually and also a comparison between them could be done for giving the optimized model. These types of optimization need to be done and may result in some better controlling parameter or some better way of obtaining the steering profile.

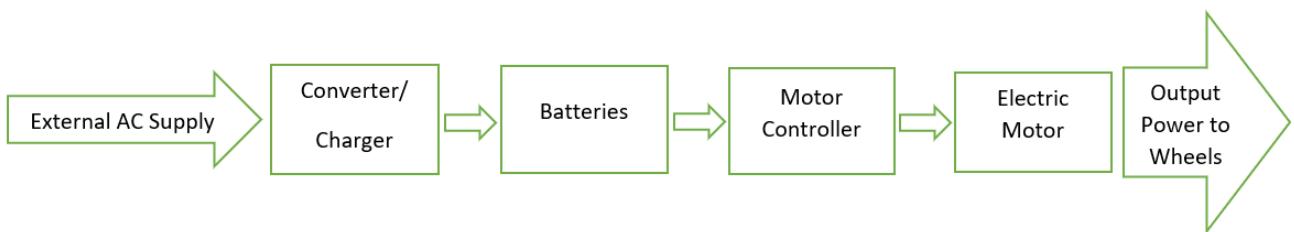
## 2.9 References

1. [https://en.wikipedia.org/wiki/PID\\_controller#/media/File:PID\\_en.svg](https://en.wikipedia.org/wiki/PID_controller#/media/File:PID_en.svg)
2. Lap Time Simulation (Výpočet teoretického času vozidla na závodním okruhu) - Thomas Novotny (<https://www.researchgate.net/publication/321485150>)
3. Tire and Vehicle Dynamics,Hans B Pacejka,2012,ISBN:978-0-08-097016-5,<https://doi.org/10.1016/C2010-0-68548-8>
4. <https://www.formulabharat.com/wp-content/uploads/2018/02/Formula-Student-Car-Design-Process.pdf>
5. Race car vehicle dynamics - William F. Milliken and Doughlas L. Milliken

## 3. Electrical Powertrain

### 3.1 Motor Selection

The Electric vehicle has many components that includes a charging module, converters, batteries, controllers and Electric Motor(s). The power flow in an Electric Vehicle can be shown as follows:-



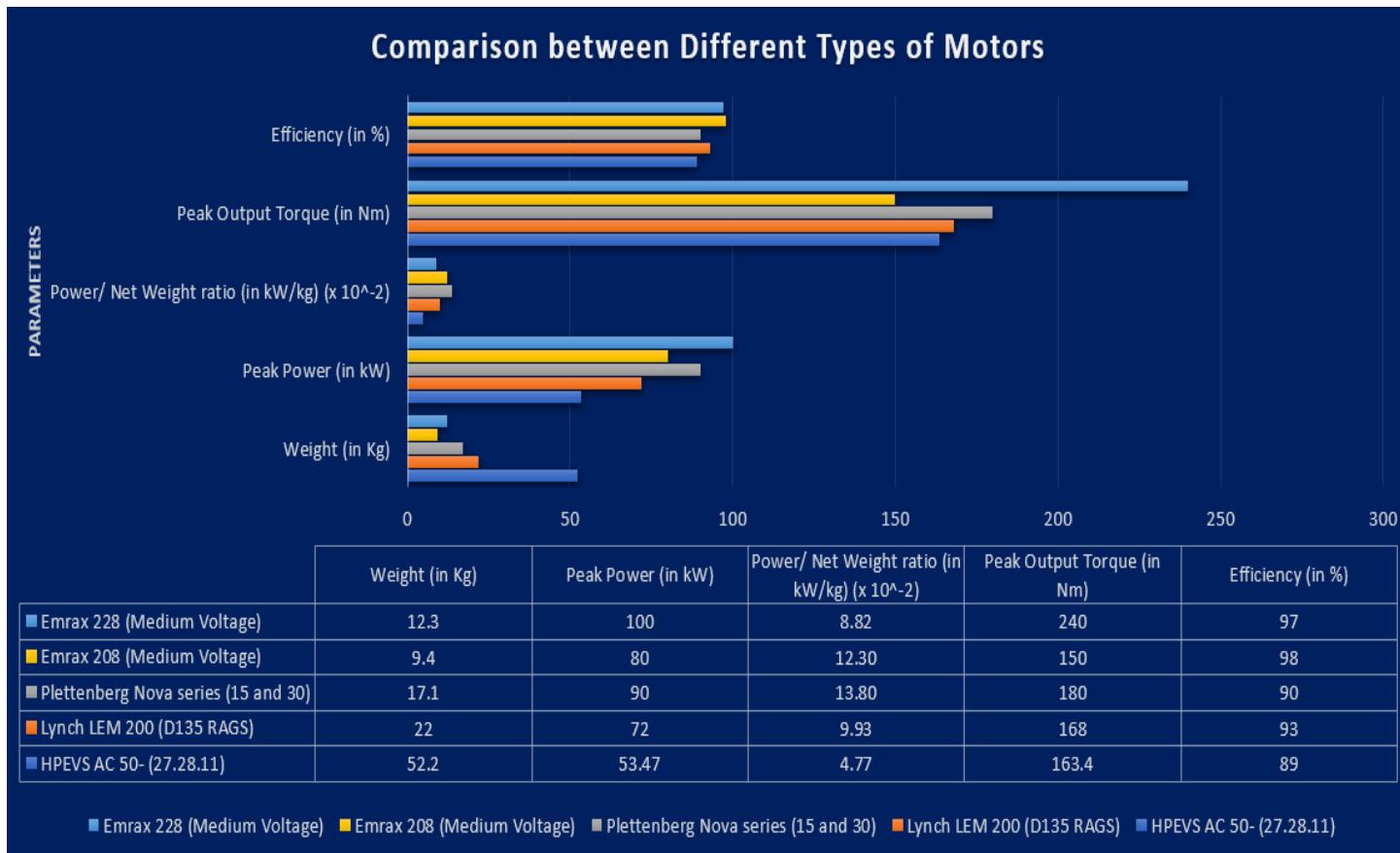
Therefore, it is clear that an electric motor determines the output characteristics of an Electric Vehicle as a whole, in terms of the output power, torque, speed, etc. While designing an Electric Vehicle, the Electric Motor is the first and foremost electrical component to be selected, thereby, making its selection crucial.

Our team wanted to achieve a benchmark while selecting the motor. This included achieving the maximum power, maximum speed and maximum efficiency with minimum weight and also making sure that it is quite compact and in accordance with the EV rules of the Formula Bharat rulebook. Moreover, the motor should be able to produce sufficient torque to overcome the force due to load and other external opposing forces, including rolling resistance force, aerodynamic drag, etc. to ensure the optimum performance of the car. Since the rulebook restricts the maximum power to 80kW power draw, the selection of the motor had to also take into consideration the peak torque output.

We did a survey on the different types of motors, used by other FSEV teams. This enhanced our thought process while selecting the Electric Motor. We compared the following motors, keeping in mind the above mentioned factors:-

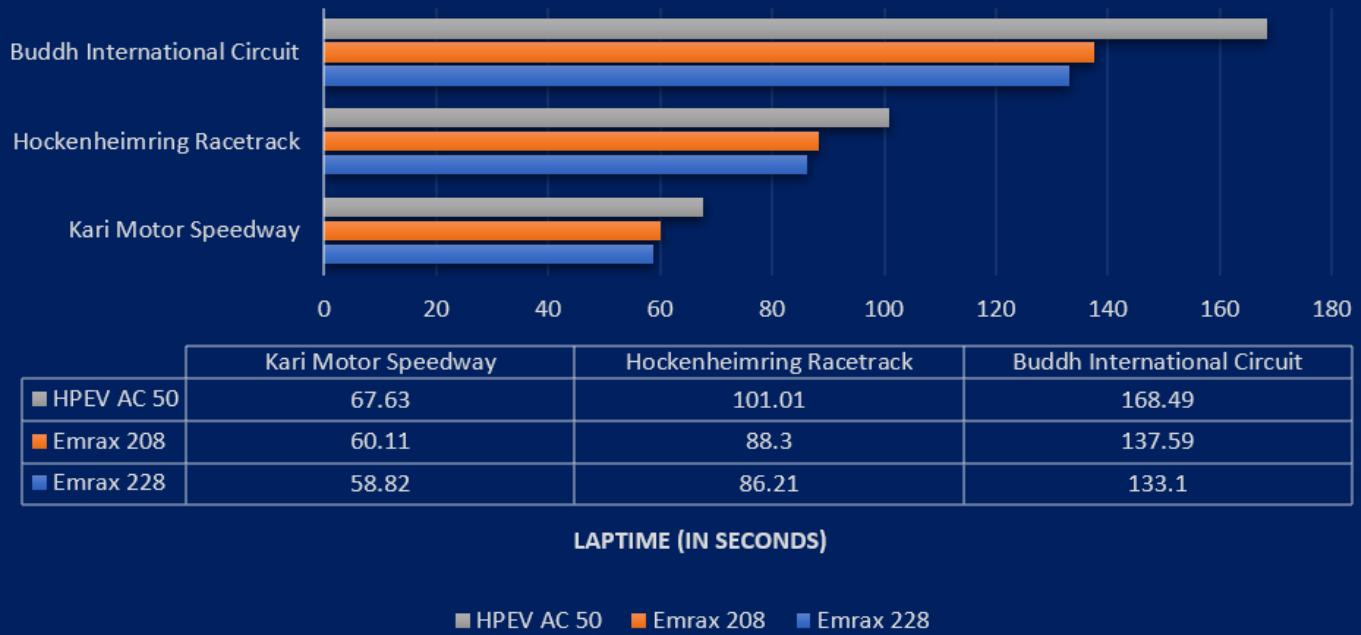
- 1) HPEVS AC 50- (27.28.11)
- 2) Lynch LEM 200 (D135 RAGS)
- 3) Plettenberg Nova series (15 and 30)
- 4) Emrax 208 (Medium Voltage)
- 5) Emrax 228 (Medium Voltage)

It is to be noted that the Emrax motors are PMSMs (Permanent Magnet Synchronous motor); Lynch motors are Brushed DC motors; HPEVS are AC Induction motors and the Plettenberg Nova series are BLDC (Brushless DC motors).



**(Note:- Power to weight ratio is calculated by taking the ratio of continuous power delivered by the motor and the total weight of the car.)**

## Laptime Simulations for different motors on different racetracks



**(Note:-** Lap time simulations were done using Optimum Lap software. Due to unavailability of proper data, lap time simulations for other motors (Plettenberg and Lynch 200) couldn't be done.)

### Detailed Analysis:-

- The HPEVS motor excessively weighs 52.2kg, leading to a poor power/weight ratio (0.0477 kW/kg). Although it is priced comparatively less and is robust, the HPEV being an induction motor, has high rotor losses (due to induced magnetic field). This reduces the overall efficiency as compared to the other motors. In addition to this, the Lap time simulations show that it has a poor lap time and lower acceleration.
- The Lynch LEM motors have to be used in pairs (weighing a high of 22 kg) in order to achieve a peak power of 72kW and a peak torque of 168Nm. They have a high efficiency of 93% and a low battery voltage requirement of 110V which proves to be advantageous. However, its power to weight ratio is 0.0993kW/kg which is comparatively low. Moreover, two motor controllers need to be used which further increases the weight. In addition to this, Lynch LEM are brushed motors which leads to problems like poor heat dissipation (due to limitations of the rotor), high rotor inertia, EMI generated by brush arcing, lower maximum speed as compared to a BLDC or PMSM motor, etc. We ruled out this motor as better choices were available.
- The choice between the Emrax motors and the Plettenberg Nova series motors was a difficult one; the former being a PMSM while the latter, a BLDC; whether to go for RWD or 4WD. In general, PMSMs

have a higher efficiency and lower mass as compared to BLDCs. Moreover, BLDCs have a problem of torque ripples (at commutation), which is not the case with PMSMs. With a pair of Plettenberg Nova 15 and a pair of Nova 30, we could achieve traction control making this extremely advantageous for the performance of the car. However, the weight of the 4 motors of Plettenberg (17.1kg) used along with 4 motor controllers exceeds the weight of Emrax motor (9.4-12.3 kg) (which requires only 1 motor controller) nearly by 90%. The peak power achieved by Plettenberg motors (90kW) is 12.5% more than that of Emrax motors(80kW) while the peak torque is 20% more. However, Emrax has an efficiency of 98% which is higher than that of the Plettenberg motors (efficiency of 90%). After an exhaustive discussion amongst the team members and deliberation, we decided to go for a RWD layout for the sake of simplicity (this year the team would also be manufacturing its first Electric Vehicle) and thus, made us opt for the Emrax motors as the befitting choice in this FSEV Concept challenge and the upcoming FB 2021. The future plans of the team does include a transition from RWD to 4WD once an in depth research on 4WD and traction control has been completed and the appropriate motors are found.

- The final choice was to be made between Emrax 208 and Emrax 228. The peak output power of Emrax 228 medium voltage is 100kW which exceeds the limit of 80 kW as mentioned in the rulebook (because of which we can't entirely depend on the Lap time simulation results). Although, the peak output torque offered by Emrax 228 MV (240Nm) is more than that of Emrax 208 MV(150Nm), the power to weight ratio of Emrax 208 MV (0.123 kW/kg) exceeds that of Emrax 228 MV (0.082 kW/kg). In addition to this, the voltage requirement of Emrax 228 MV (470V DC) is 46.875% more than that of the Emrax 208 MV (320V DC), which would increase not only the size of the battery pack but also the overall weight of the car (by roughly 57Kgs). Handling such a high voltage would be a great risk factor for the safety of the team members, thus causing fluctuations in the reliability parameter. As compared to Emrax 208, Emrax 228 can achieve a higher acceleration, but a lower maximum speed, by using a lower gear ratio.
- Thus, Emrax 208 (Medium Voltage) is the best choice for us as it is a perfect blend of being light weighted, having a very high efficiency and high power, achieving an adequate lap time and achieving a relatively high peak output torque.

## 3.2 Motor Controller

After the motor's selection process is complete and the motor finally selected is the Emrax 208, the next step was to select a motor controller that is the most compatible with Emrax 208 while not putting pressure on any of the other HV and LV electrical equipment.

The motor controller is the electronics package that operates between the batteries and the motor to control the electric vehicle's speed, and acceleration, much like a carburetor does in a gasoline-powered vehicle. The controller transforms the battery's direct current into alternating current (for AC motors only) and regulates the battery's energy flow. Unlike the carburetor, the controller will also reverse the motor rotation (so the vehicle can go in reverse), and convert the motor to a generator (so that the kinetic energy of motion recharges the battery when the brake is applied).

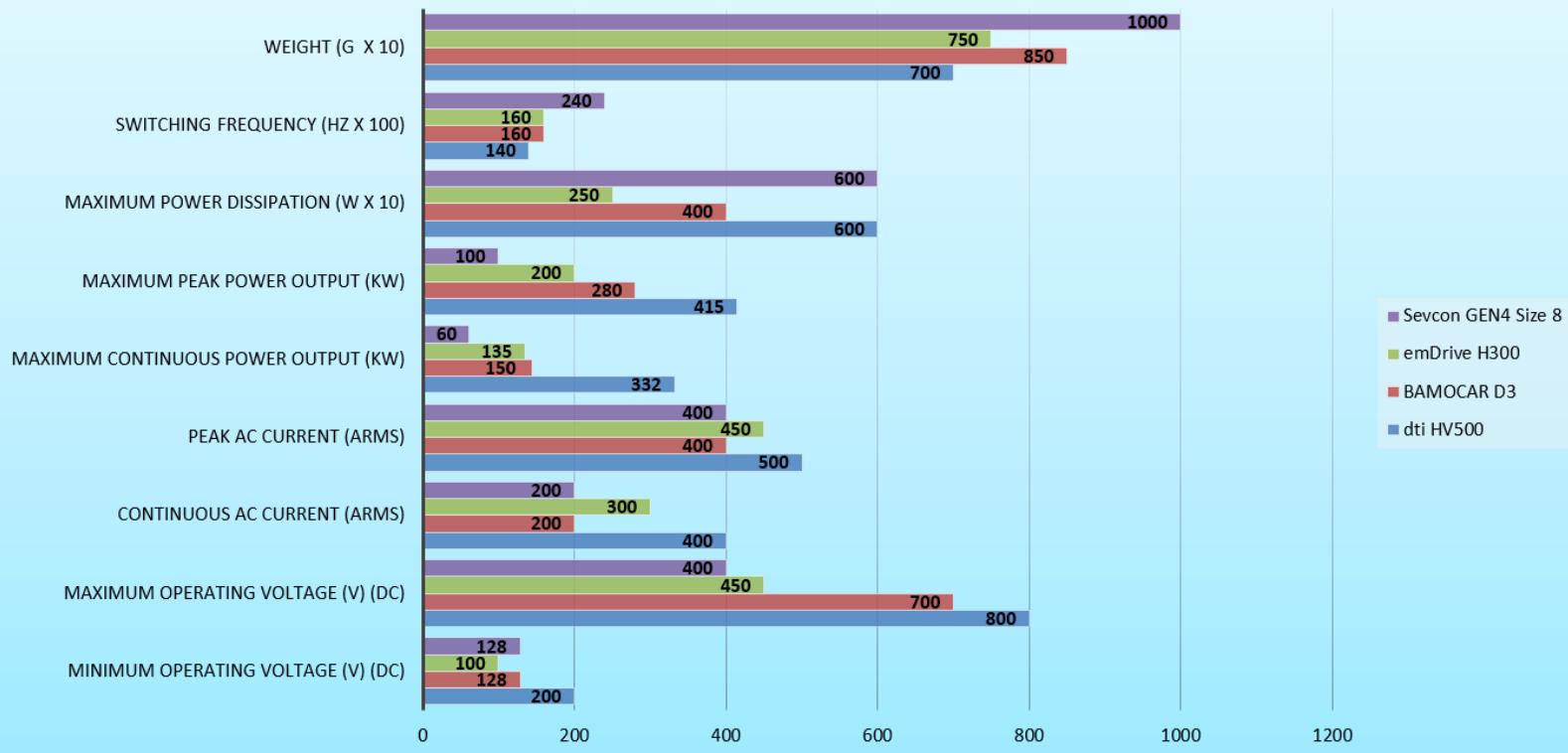
Many factors influenced the selection of the motor controller, taking into consideration the events that wanted the most out of the vehicle like the Acceleration and the Endurance and efficiency events and keeping in mind the limitations of the rulebook. To make our motor controller selection the frontrunner, we surveyed the other available motor controllers that were used in any FS event before and compared the statistics concerning every one of them. The four narrowed down motor controllers in comparison were:

- 1. dti HV500 (Drivetrain Innovation High Voltage Motor Controllers)**
- 2. BAMOCAR D3 (Unitek Elektronik Industries)**
- 3. emDrive H300 (Emsiso Advanced Motor Controllers)**
- 4. Sevcon GEN4 S8 (Sevcon)**

The comparison majorly took place over the parameters that were of the utmost importance to the vehicle, keeping in mind the driver's rules and safety. We had to make sure that the controller's power output after having received the proper voltage from the accumulator was in the range, which was suitable for the efficient activity of the motor and did not fall behind the requirements whenever needed. The power output, the power dissipation without proper use as heat and sound, the temperature range for the motor controller's operation and efficiency derating with temperature changes were studied to keep the motor controller's efficiency at the peak. The motor operation in an EV is as good as the motor controller can provide. The motor demands were maximum power, maximum speed, maximum efficiency, and sufficient torque to overcome any kind of resistance in the drive like rolling frictional resistance and drag due to the vehicle aerodynamics. So, the motor controller had to be one that could provide maximum continuous power output while operating at maximum efficiency without overstraining and overheating its internal components.

Motor Controller Model				
Specifications	dti HV500	BAMOCAR D3	emDrive H300	Sevcon GEN4 Size 8
Minimum Operating Voltage (V) (DC)	200	128	100	128
Maximum Operating Voltage (V) (DC)	800	700	450	400
Continuous AC Current (Arms)	400	200	300	200
Peak AC Current (Arms)	500	400	450	400
Maximum Continuous Power Output (kW)	332	150	135	60
Maximum Peak Power Output (kW)	415	280	200	100
Maximum Power Dissipation (W x 10)	600	400	250	600
Switching frequency (Hz x 100)	140	160	160	240
Weight (g x 10)	700	850	750	1000
Specifications not elaborated in the chart				
Dimension (l/w/h): (in mm)	420 x 213 x 77	355 x 230 x 135	263 x 244 x 127	322 x 322 x 107
Cooling Mechanism	Integrated Water Cooling	Water / Glycol / Air / Oil Cooling	Water / Glycol Cooling	Water / Glycol / Air / Oil Cooling
Design Grading	IP65	IP65	IP65	IP66
Operational Temperature:	0°C to 100°C (with significant efficiency derating)	0 to +45 °C (without derating); +45 °C to +65°C (derating in performance, reduction by 2% / °C);	-20°C to +60°C (derating is absent); +60°C to +65°C (with derating);	-30°C to +25°C (no current or time derating); +25°C to +80°C (no current derating, but reduced time at rated operating point); +80°C to +90°C and -40°C to -30°C (with derating);

## Comparison of Various Specifications for Different Motor Controllers.



**(NOTE: Power dissipation here refers to the amount of power that goes to waste and not supplied to the motor even after being taken from the accumulator, noise or heat, or any dissipation. Power output refers to the amount of power supplied from the accumulator to the motor considering the losses.)**

## One to One comparison.

- Let us take a look at the Sevcon Gen4 S8 motor controller first. The team wanted primarily to perform the best in two of the highest-scoring events, Acceleration, and Endurance. These events demand a high supply of peak and continuous power and thus would generate much heat due to the load on the motor and controller. The Gen4 S8 had the most efficient temperature relation for efficiency, and the derating was absent for either very high or shallow temperatures (as seen from the table), neither of which were achievable under ideal power supply conditions. Comparing this with emDrive H300's temperature derating, the latter's efficiency drop would occur at temperatures lower than the former and thus would not be better than the former. However, with this huge pro favoring the Gen4 S8, when we look at

the power outputs, both the peak and the continuous (100kW and 60kW respectively), we see that this motor controller stands nowhere near the others while having the same current supply rate (like the BAMOCAR D3). Even the dti HV500, which has a significant derating with temperature, can overshadow this problem by having a much higher power output (415kW and 332kW), thus bringing this above Gen4 S8. Looking at the switching frequency here (24kHz), a higher switching frequency will result in a smaller inductor but increase the switching losses in the circuit, which is evident from the power dissipation (6kW) across the other motor controllers on the table. The culmination if the low power output, the more weight of the motor controller (10kg), a high switching frequency,

- The dti HV500 was one of the first motor controllers taken into consideration and was turning out to be a promising one. Out of the ones taken for comparison, this one had the highest peak and continuous power output to the motor (415kW and 332kW respectively) and thus, if chosen, could have been able to sustain the power requirements for the Emrax 208 (80kW). The power dissipation value (600kW) is also the highest amongst the crowd and was way too high than what the team wanted to have. The peak and continuous values of RMS current (500Arms and 400Arms respectively) cause it to be functioning on huge amounts of electric current, which causes much heating of the motor controller and can also damage the electrical component surrounding it. Although these current values would be beneficial to the motor, not causing any hindrance to the motor. Still, as evident from the highest operating temperature being around 100°C, even after proper integrated cooling systems, there was much risk in installing other electronics around it without extra insulation from the other components, which would, in turn, add to the weight of the vehicle and thus increase the load, and expenses. Looking at the Sevcon Gen4 S8 and its operational temperatures, the derating in the controller's efficiency with an increase in temperature is way too eased for Gen4 S8 than for dti HV500. Even though the latter's (dti HV500) operating temperature is lower than the former (Gen4 S8), the clever use of liquid cooling in Gen4 S8 will not cause it to reach such high temperatures, thus causing constant efficiency. The minimum operational voltage for this dti HV500 (200V DC) was also too high as compared to the other motor controllers in question, like the BAMOCAR D3 (128V DC) or the emDrive H300 (100V DC). Thus if any failure were to occur in any of the individual cells or a group of cells in the accumulator, causing a supply of less than normal voltage to, the motor controller would not be able to function and thus keep the motor; hence the car running. Due to these factors, the dti HV500 was renounced as the team's motor controller choice.
- The next motor controller checked alongside the dti HV500 was the emDrive H300. Most attractive about this motor controller was its lower power dissipation than the other motor controllers in question (250kW, whereas the BAMOCAR D3 stands at 400kW, the dti HV500 is way up at 600kW). This fact allowed the team to think that they could ease the load onto the accumulator for the voltage demand and thus supply more power efficiently to the motor. However, the peak power output supplied by this motor controller (250kW) was not good enough, with respect to the other motor controllers as in the table, for the motor to be sustained and to keep it running in cases of sudden acceleration, like while at the start of the race line or during events such as Acceleration. The motor controller also falls short at the continuous power supply (135kW), although by a smaller amount compared to BAMOCAR

D3, still, that small difference would make the vehicle underperform in the Endurance event. The previously mentioned two events are the most scoring events in the entire FSAE competition, and the team did not want to fall behind in those two major ones. Due to the weightage of the other factors against this motor controller, smaller factors like the nominally high switching frequency of the MOSFET or the IGBT installed inside dominate towards its infamy. Even though the operational range of voltage for this motor controller is way lower than the others, the price is also lower than most others (\$2600), which favor this model, but the dominating stats were the power outputs. Thus this motor controller also did not meet the team's expectations completely

- The last decision to be taken fell between the BAMOCAR D3 and emDrive H300. After discussing the emDrive H300, factors favoring BAMOCAR D3 were the range of operational voltages (128V DC to 700V DC) which helps the motor controller to function at around both the extremes of the voltage supply from the accumulator, thus reducing the problems caused due to low voltage supply or short circuits. This case was not supporting the emDrive H300 as mentioned earlier. The maximum peak and continuous power outputs of BAMOCAR D3 (280kW and 450kW respectively) also triumph over emDrive H300. This motor controller(8.5kg) weight is more than that of the emDrive H300 (7.5kg), which is something that we thought about how to resolve. As a redemption, the BAMOCAR D3 offers air and liquid cooling(effective weight around 8.7kg), whereas the emDrive H300 only uses a liquid coolant (effective weight around 8.1kg). This factor caused a huge reduction in the motor controllers' effective weight difference and thus reduced the weight dependent factor for the BAMOCAR D3 and favored it more. The switching frequencies of both were the same (16kHz) and hence not a point for debate. Thus considering all the factors, the team sought Unitek BAMOCAR D3 as the most suitable motor controller for the vehicle and thus decided to install it in the vehicle

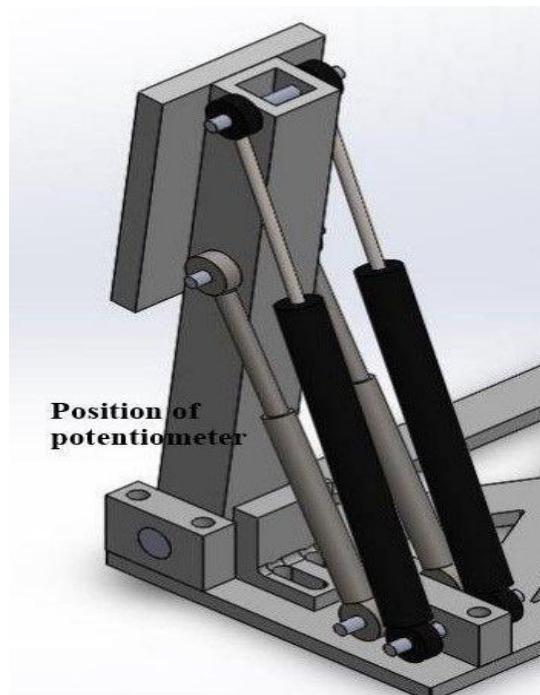
### 3.3 Electronic Torque Control

The motor is controlled using the torque control mode, and that requires a differential input signal to the motor controller. This requires the measuring of the pedal travel. The pedal travel is measured using the Accelerator Pedal Position Sensor(APPS).

The accelerator pedal position sensor is used for measuring the travel of the accelerator pedal. According to rulebook two sensors must be utilized as APPSs and in case implausibility of values of the two sensors persists for more than 100 ms, the power to motor is completely shut down.

**The implausibility is defined as a difference of more than 10% of pedal travel.**

There were multiple ways to measure the plausibility. We tried both the approach of both the hardware as well as a software.



*Fig : Potentiometer Mounting*

## -HARDWARE

This would have required us to compare the values of the APPS signals directly using the hardware control, It involved the use of comparator and subtractor.

This approach was difficult in terms of completely satisfying the rulebook, as reducing the torque down to 0 when we get an error, was not possible. Simulations showed that we were getting output voltage from the circuit which did not reduce torque to 0.

## -SOFTWARE

For using software comparison, we had to create a difference in the voltage signal

The APPS is designed using two linear potentiometers placed at the base of the car,integrated into the pedal box attached to the shaft such that the travel/stroke length is minimum for 100% pedal travel i.e fully applied.

It was decided to use 2 same potentiometers to prevent their transfer functions intersecting at any pedal angle. Our approach was to shift the transfer function of one of the signal lines of the potentiometer, allowing them to be parallel, and therefore non intersecting.

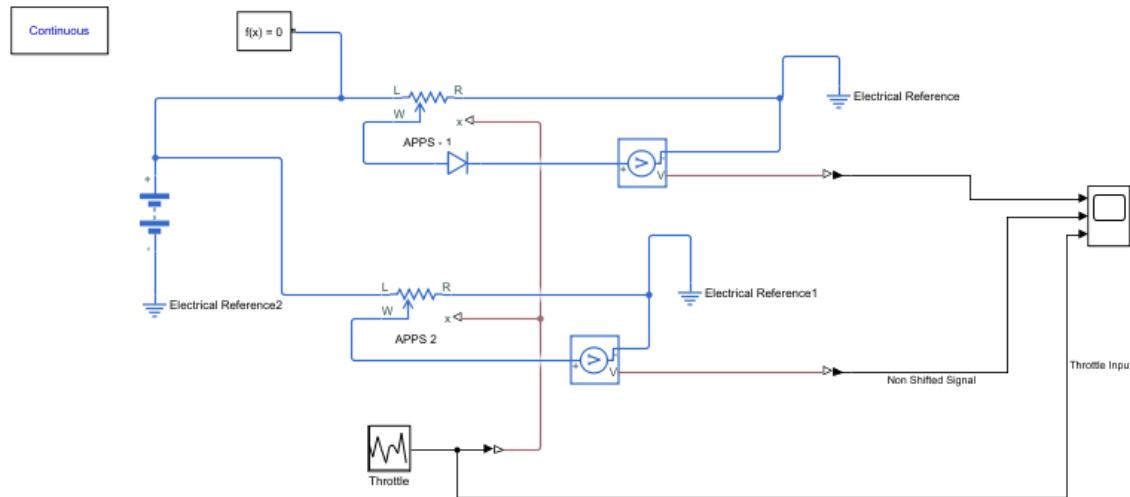


Fig: Simulation of APPS Circuitry

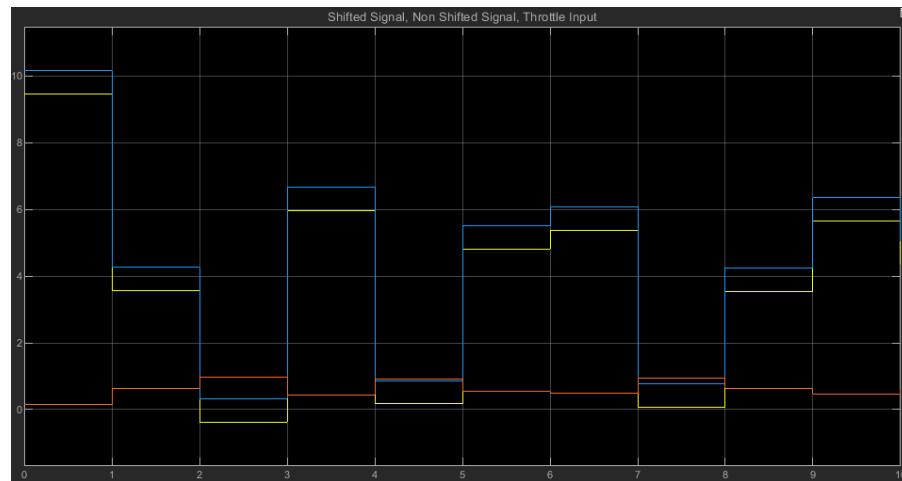


Fig: Simulation showing difference of 0.7V(Diode Offset Voltage) for a given pedal cycle

## KPM SERIES SENSOR

### Mini size ends-joints, stroke up to 700m



*Fig: Potentiometer*

### 3.4 Motor Mounts

The motor will be mounted on both ends, thus 2 different mounts were prepared, 1 for the live end and the other for the dead end. Difference in their design comes as the dead end has wires and connections to the motor while the live end is free of obstructions.

The dead end mount was directly bolted on the motor and attached to the chassis. Slots were made for cooling connections and smaller holes to allow the mount to be bolted on the motor.

The live end mount was mounted on top of bearings attached to the motor shaft.

The first step in designing the mounts was locating fixtures for the mount on the chassis. Using these points and the motor location, geometry of the mounts was made by also considering accessibility, compactibility, and weight of mounts. Material used was Aluminium 6 series because it was lighter and more readily available than materials like Mild Steel.

Upon running simulations, the design was finalised after obtaining a satisfactory factor and mass.

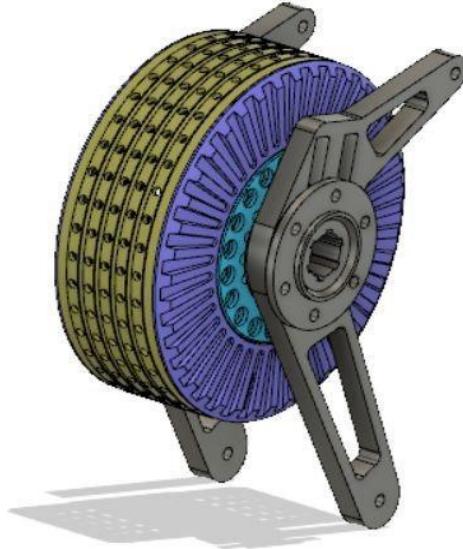


Fig: Motor mount

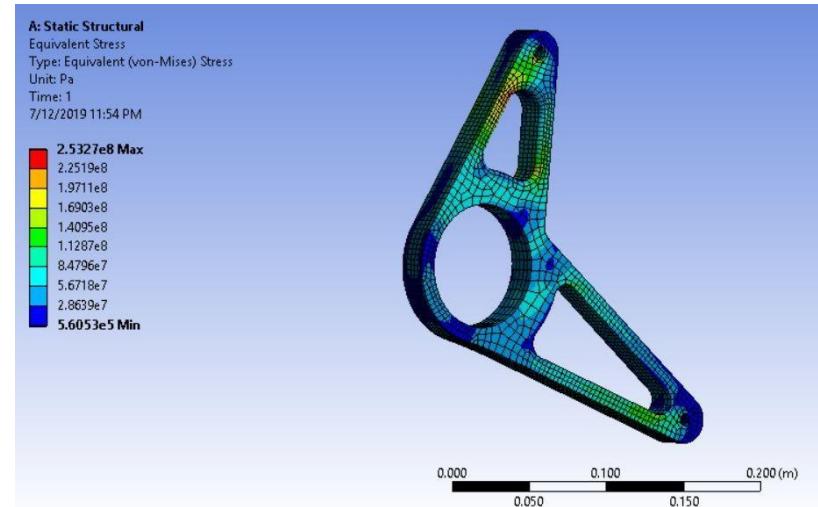


Fig: Mount Simulation

### 3.5 Drivetrain

#### Sprocket Design, Chain Tension Calculation, and Chain Selection:

It is imperative that the number of teeth on both the sprockets be a whole number. Hence, in this iterative design process, the number of teeth on the driving and driven sprockets were finalised to be 10 and 56—giving rise to the optimum gear ratio of 5.6. To proceed further with the design, it is necessary to finalise the pitch of the sprocket which is an important input in calculating the dimensions of the sprocket and ultimately the tension in the chain that will be generated by the operating torque. However, the pitch of the sprocket should match that of the chain for smooth operation, and the chain tension has to be known to decide the chain number. To resolve this catch-22 situation, it was assumed that chain 41 with a pitch of 12.7 mm (0.5 in) would be a suitable fit.

With the knowledge of the number of teeth and the pitch of the sprocket, the pitch circle diameter of the driving and driven sprockets were calculated as 40 mm and 226 mm respectively. Using these values, the chain tension was calculated to be 6.36 kN. The average tensile strength of the chosen chain is 10.7 kN which gives rise to a safety factor of 1.68. Hence, the chain-drive system is designed to withstand the working load, but at the same time, the chain is also the weakest part of the drivetrain assembly which means that it will fail before any other critical or expensive component. Chain failure is much less dangerous, and the replacement is also cheap and simple.

$$\text{Pitch Diameter} = \frac{\text{Pitch}}{\sin\left(\frac{180}{\text{Number of teeth}}\right)}$$

*For the driving sprocket, number of teeth = 10;*

$$\text{Hence, Pitch Diameter} = \frac{12.7}{\sin\left(\frac{180}{10}\right)} = 41 \text{ mm}$$

*For the driven sprocket, number of teeth = 56;*

$$\text{Hence, Pitch Diameter} = \frac{12.7}{\sin\left(\frac{180}{56}\right)} = 226.5 \text{ mm}$$

As the tyre's traction limit was reached before the motor generated the maximum torque, the limiting condition to find the tension in the chain was taken to be the tyre-slip condition. That is, the maximum tension in the chain occurs just when the tyres are about to slip. The tyre's traction limit was found out to be 1530 N from experimental data. As the car uses two wheel-drive system, the chain tension must generate a torque to slip both the tyres.

$$\begin{aligned} \text{Torque in sprocket} &= \text{Tyre Traction Limit} \times \text{Radius of Tyre} \\ &= 1530 \text{ N} \times 0.23495 \text{ m} \times 2 \text{ (Both tyres)} = 718.947 \text{ N-m} \end{aligned}$$

$$\text{Torque in sprocket} = \text{Chain Tension} \times \text{Radius of Sprocket}$$

$$\text{Chain Tension} = \frac{\text{Torque in Sprocket}}{\text{Radius of Sprocket}} = \frac{718.947 \text{ N-m}}{0.113 \text{ m}} = 6362.36 \text{ N}$$

A minimum angle of wrap of 120° meant that the sprockets should be at least 186 mm apart. However, to prevent disengaging of the chain due to excessive slackening, the upper limit was also set to be sixty times the pitch of the chain – 762 mm. Considering other design and space constraints, the centre-to-centre distance was finalised as 254 mm, giving an angle of wrap around the smaller sprocket of 140 degrees.

## 4 . Accumulator

### 4.1 Cell chemistry and model selection

Parameters considered for selection of cell chemistry:

1. High Energy Density (Wh/L)
2. High Specific Energy (Wh/kg)
3. High Specific Power (W/kg)
4. Charging and Discharging abilities
5. Ecologically sound
6. Sustainable

Various chemistries like Nickel Metal Hydride , Nickel Cadmium , Lithium ion cells , Lithium polymer etc. were considered. Keeping in mind the various parameters, lithium ion cells were selected due to the following advantages:

1. They have no memory effect which allows users to charge and discharge them much more flexibly than other batteries.
2. Lithium-ion batteries generally offer a very high coulombic efficiency throughout the state of charge range.
3. Lithium-ion batteries have very low self-discharge rates compared to other types of batteries.

There were also some disadvantages to this high performance. Short circuit currents can be much higher and an uncontrolled release of energy can be larger. They are fragile and need to be handled with caution. But its high energy content and power capabilities made us select Lithium ion cells over other chemistries.

Lithium ion phosphate (LiFePO<sub>4</sub>) was selected due to its stable charge and discharge characteristics.

We calculated the energy requirement of the battery pack to be around 6 kWh. Keeping the various selection parameters in mind and considering the number of cells required based on the energy requirement and size constraints of our module, an excel spreadsheet was prepared shown below.

Considering all the possibilities shown above, model no- **AMP20M1HD-A** of **A123** was selected due to less number of cells required, leading to compact accumulator design and providing high power density.

It also had added advantages such as low internal resistance, safer operation due to inherent resistance to thermal runaway. General specifications of the cell are shown below.

Name	A123	Melasta	Samsung	Bestgo	TEAMGIANT	Melasta
Model Number	AMP20M1HD-A	SLPBB042126	INR18650-25R5	BCPNE20T	HR3780156240	HP49C260
Column2	Lithium-ion	Lithium-ion	Lithium-ion	Lithium-ion	Lithium-ion	Ni-Zn
Type	Pouch	Pouch	Cylindrical	Pouch	Pouch	Cylindrical
Specific Energy (Wh/kg)	131	190	206	180	210	213
Specific Power(W/kg)	1333	2400	1627	361	628	286
Energy Density(Wh/L)	247	445	545	372	395	525
Weight(g)	495	126	45	410	565	45
Energy(KWh)	0.064845	0.02394	0.00927	0.0738	0.11865	0.009585
No. of cells required	90	245	633	79	49	613

Specification	Value	Notes/Comments
Peak 10s Discharge power	820W	SOC = 100%, Tcell = 23 °C, Assumed DCR = 2 mOhm (nominal)
DCR Impedance	1.5 – 3 mOhm	10s, 240A, @ 50% SOC
ACR Impedance	0.78 mOhm	1kHz, @ 50% SOC
Operating Temp Range	-30 °C to +60 °C	Ambient around cell
Storage temperature range	-40 °C to +65 °C	
Weight	495 grams	+/- 10g
Cycle Life To 80% Beginning of Life (BOL) capacity	3000 cycles	100% Full DOD cycles, 1C/-2C @ 23 °C, 8 – 14 psi face clamp pressure

Specification	Value	Notes/Comments
Nominal Capacity	20 Ah	
Minimum Capacity	19.5 Ah	25 °C, 6A Discharge, 3.6V to 2.0V, at BOL
Nominal Voltage	3.3V	@ 50% SOC
Voltage Range	2.0 to 3.6V	Fully Discharged to Fully Charged
Absolute Maximum terminal voltage	4.0	Above which will cause immediate damage to the cell
Recommended maximum charge voltage	3.6V	
Recommended float charge voltage	3.5V	
Recommended end of discharge cutoff	2.0V	
Recommended standard charge current	20A	to 3.6V
Recommended maximum charge current	100A	to 3.6V , Cell temperature < +85 °C
Pulse 10s charge current	200A	23 °C ≤ Tcell < +85 °C, Vcell < 3.8V
Maximum discharge continuous current	200A	23 °C ≤ Tcell < +85 °C, SOC = 50%
Pulse 10s discharge current	600A	23 °C ≤ Tcell < +85 °C, SOC = 50%

## 4.2 Electrical Parameter Estimation

An electric vehicle's source of energy is its battery pack. It is one of the largest components of the vehicle. Making a battery pack not only requires a thorough understanding of the electrical aspects , but the purpose of the vehicle usage has to be kept in mind. The size of the battery pack was determined carefully considering multiple parameters which must follow the rules of the Formula Bharat Competition.

### CONSTRAINTS

#### 1. Weight

The battery is the heaviest component of the electric vehicle. It is important to strike balance between the weight and the energy requirement.

#### 2. Chassis Size

The battery has to be fitted in the chassis in a way that it is easily removable for charging and fitted properly when in use.

## Determination of Battery Parameters

Determining the battery parameters such as energy requirements, power requirements, voltage, current etc. is essential to analyse the performance of the car.

We developed our own lap time simulation software. The results obtained were quite similar to the values obtained by hand calculations.

The following hand calculations were done in order to determine the parameters required:

- **Determining Battery Voltage**

The motor **EMRAX 208 Liquid Cooled** was selected and the nominal battery voltage was determined using its power and current requirements.

Continuous Power Output: 30-32 kW

Continuous Current: 160 Amps (RMS)

Nominal voltage \* Continuous Current input = Continuous power output

Therefore, **Estimated nominal voltage = 282.2 V**

- **Determining Battery Capacity**

The longest event of the competition is ENDURANCE. The rules don't allow us to charge the battery in between. It is a 22km event and thus the minimum range of the battery pack should be 22km.

The average power and average speed were considered from the simulation results.

Minimum Range = 22km

Average power = 16.212 kW

Average Speed = 60 km/hr

Calculate time = Range/Average speed = 22/60 = 0.367 hours

Energy required = Average power \* time = 5.94 kWh

Therefore, **Energy Capacity Required = 5.94 kWh**

This value is very close to that calculated using the lap time simulation software which was **5.87 kWh**.

- **Determining Battery Power**

The event also has an acceleration sub-event. This event requires the vehicle to have the maximum acceleration i.e. reach the maximum velocity (and with it the maximum Kinetic Energy) within the least amount of time.

Since Power = Change in Kinetic Energy / Time , the higher the power, the better the acceleration.

The rulebook has limited the Power output of the battery to 80W for safety reasons. The Battery pack was designed keeping it in mind.

Now as per the datasheet of the cells used, the peak 10s discharge power was 820 W for a single cell. So for 90 cells (The battery pack is 90s1p), we would get a peak power output of  $90 \times 0.820 \text{ kW}$ .

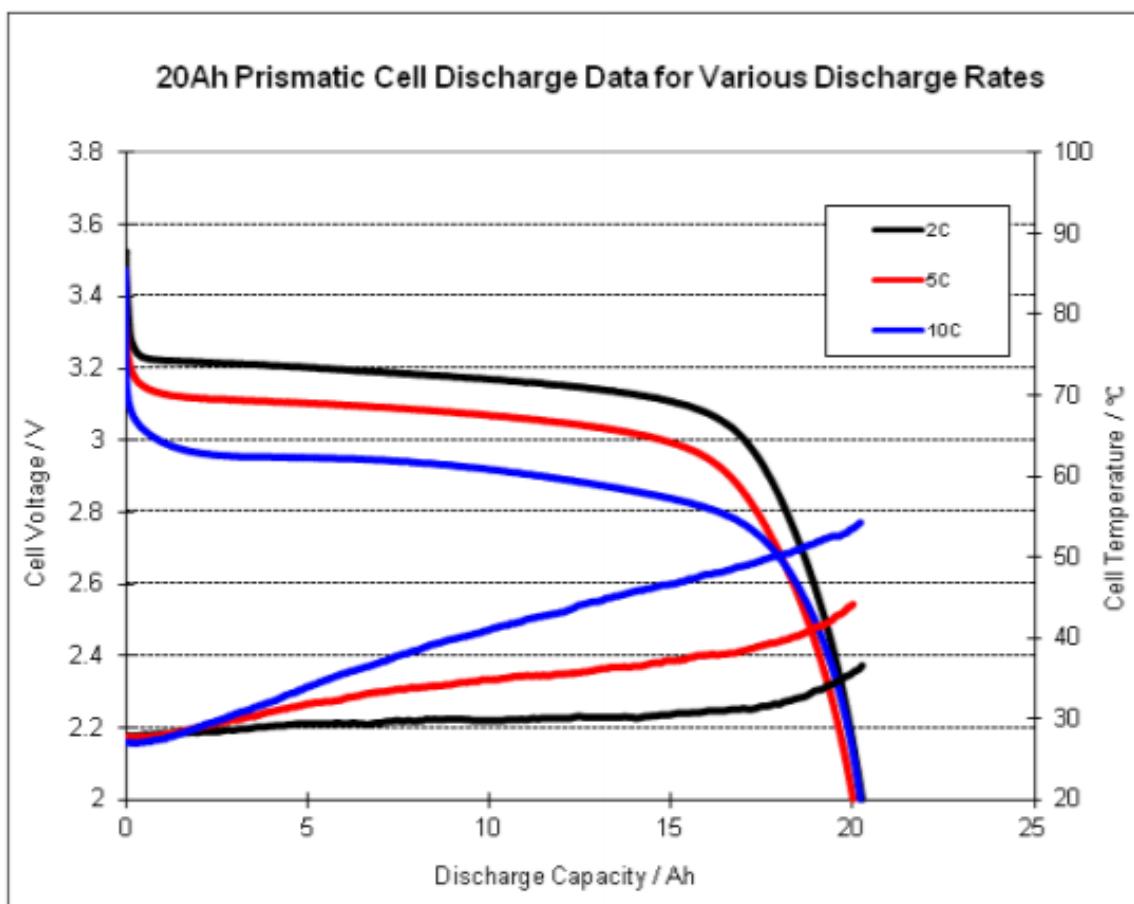
**Peak Power Output = 73.8kW for 10s**

## 4.3 Cell Configuration

Pack sizing methodology was followed for cell configuration:

1. Increase series count of cells until the desired voltage is reached.
2. Increase parallel count of cells until the desired energy capacity is reached.

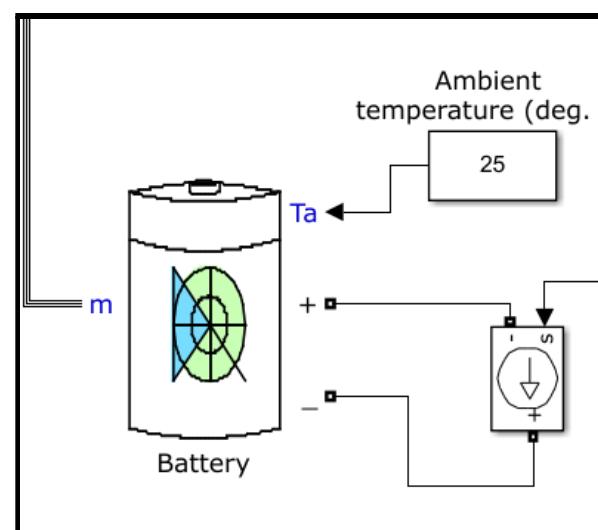
- The maximum voltage of 19.6Ah A123 cells is 3.3V. Initially, the team came up with a configuration of 96s1p. It fulfilled both the voltage and energy requirements.
- Then after doing some analysis of the discharge profile of our cells, we figured out that very little amount of the total energy is stored at maximum voltage. The voltage of the cells falls to 3.2 almost immediately. We also found out that by dividing the area under the curve(shown below) between 3.2 and 3.3 by the total area under the curve, only 0.5% of the total energy is stored in this range.
- Thus we can charge the cells only up to 3.2V, this increases the capacity gain. Thus our new configuration now would be 90s1p. With this configuration we can divide our battery into 6 modules each with 15s1p. This also does not break the rule that each cell stack should not exceed 120V.



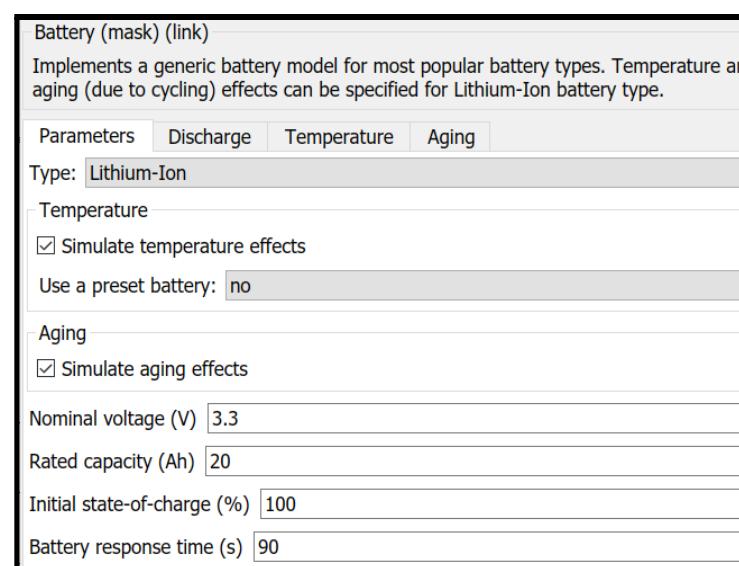
## 4.4 BATTERY MODELLING ON MATLAB-SIMULINK

For the purpose of battery modelling we have used pre existing blocks in MATLAB software to simulate the Temperature, Age, Capacity Fade with age, Voltage and SOC of cell and running it on preset current profiles which are similar to the ones we are going to face in the Endurance race.

### 4.4.1 BATTERY BLOCK IN MATLAB



The battery block in MATLAB is a self contained set of equations which model several battery chemistries and can be used to extract valuable information about our cell's performance. The input parameters to this block as interpreted from the datasheet is as follows:



Parameters	Discharge	Temperature	Aging
<input checked="" type="checkbox"/> Determined from the nominal parameters of the battery			
Maximum capacity (Ah)	20		
Cut-off Voltage (V)	2.475		
Fully charged voltage (V)	3.8412		
Nominal discharge current (A)	8.6957		
Internal resistance (Ohms)	0.00165		
Capacity (Ah) at nominal voltage	18.087		
Exponential zone [Voltage (V), Capacity (Ah)]	[3.5653 0.98261]		
Display characteristics			
Discharge current [i <sub>1</sub> , i <sub>2</sub> , i <sub>3</sub> , ...] (A)	[100 120 140 160 180 200]		
Units	Ampere-hour	Plot	

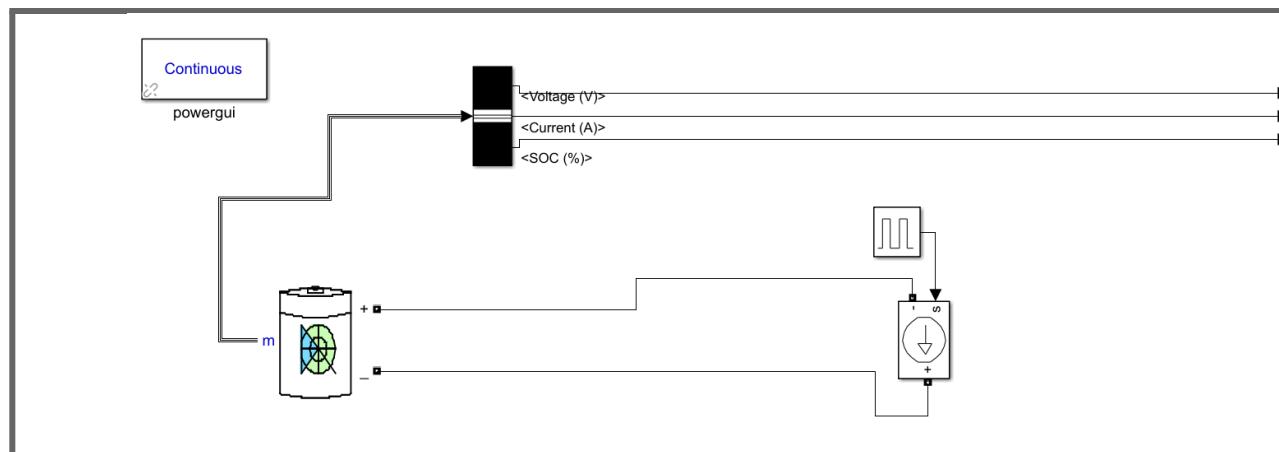
Initial battery age (Equivalent full cycles)	0
Aging model sampling time (s)	30
Aging characteristics at ambient temperature Ta1	
Ambient temperature Ta1 (deg. C)	25
Capacity at EOL (End Of Life) (Ah)	14.625
Internal resistance at EOL (Ohms)	0.003*1.2
Charge current (nominal, maximum) [I <sub>c</sub> (A), I <sub>cmax</sub> (A)]	[20, 60]
Discharge current (nominal, maximum) [I <sub>d</sub> (A), I <sub>dmax</sub> (A)]	[20, 200]
Cycle life at 100 % DOD, I <sub>c</sub> and I <sub>d</sub> (Cycles)	1500
Cycle life at 25 % DOD, I <sub>c</sub> and I <sub>d</sub> (Cycles)	10445
Cycle life at 100 % DOD, I <sub>cmax</sub> and I <sub>d</sub> (Cycles)	5000
Aging characteristics at ambient temperature Ta2	
Ambient temperature Ta2 (deg. C)	45
Cycle life at 100 % DOD, I <sub>c</sub> and I <sub>d</sub> (Cycles)	982

These parameters have been used for all the models which are presented below have been used throughout the models presented.

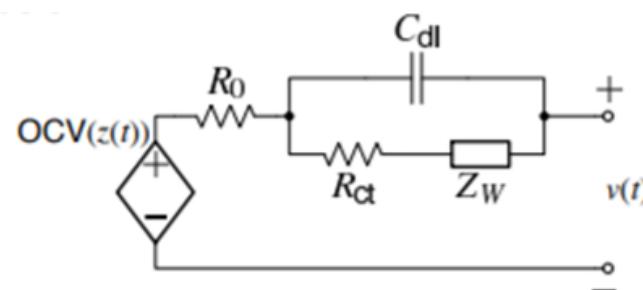
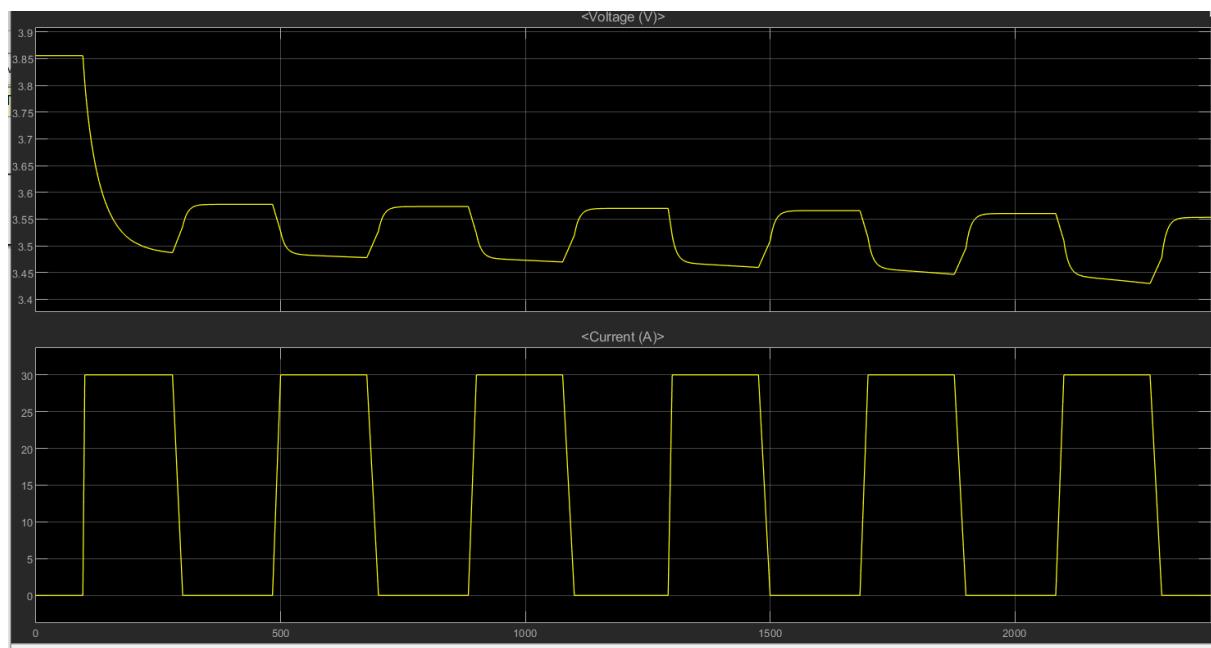
The battery block can conveniently be used to study discharge characteristics of the cell since it provides three output ports Voltage, Current and SOC of the battery to study them.

#### 4.4.2 PULSE DISCHARGE TEST

It is standard procedure to run the any cell through a pulse discharge test to get a quick approximation of the parameters for an ESC(Enhanced Self Correcting) model of Li-ion cell. Due to the unavailability of a Cell-Cycler at our institute we have used a simple cell model to simulate the results of the same.



The results of the test were really close to the actual expected results



The estimated parameters from this experiment, using a single RC pair circuit

1.  $R_0 = 1.667 \text{e-}3$
2.  $R_{ct} = 1.633 \text{e-}3$
3.  $C_{dl} = 438 \text{mF}$

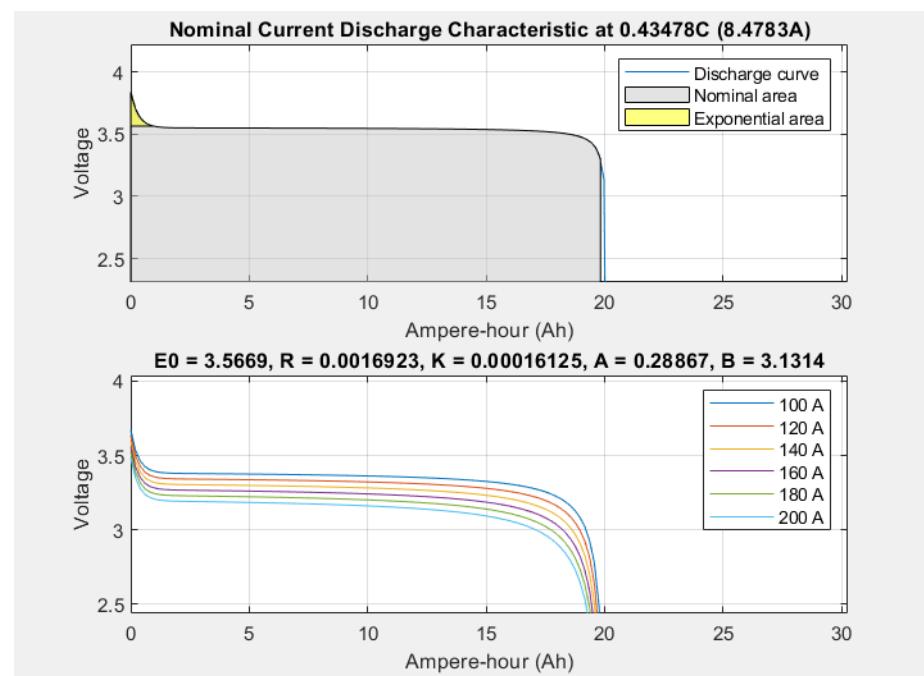
More results can be obtained from the model by tweaking parameters of the experiment like the pulse discharge current and pulse widths.

1. Heavier discharge pulse results in a sharper fall from the exponential zone to the baseline nominal voltage
2. An effect which is noticeable in short high current pulses is that, as the number of pulse progresses periodically the relaxation time of the battery after the pulse got higher after each pulse (i.e. the smoothness of the kink increased after each pulse) as demonstrated in figure below

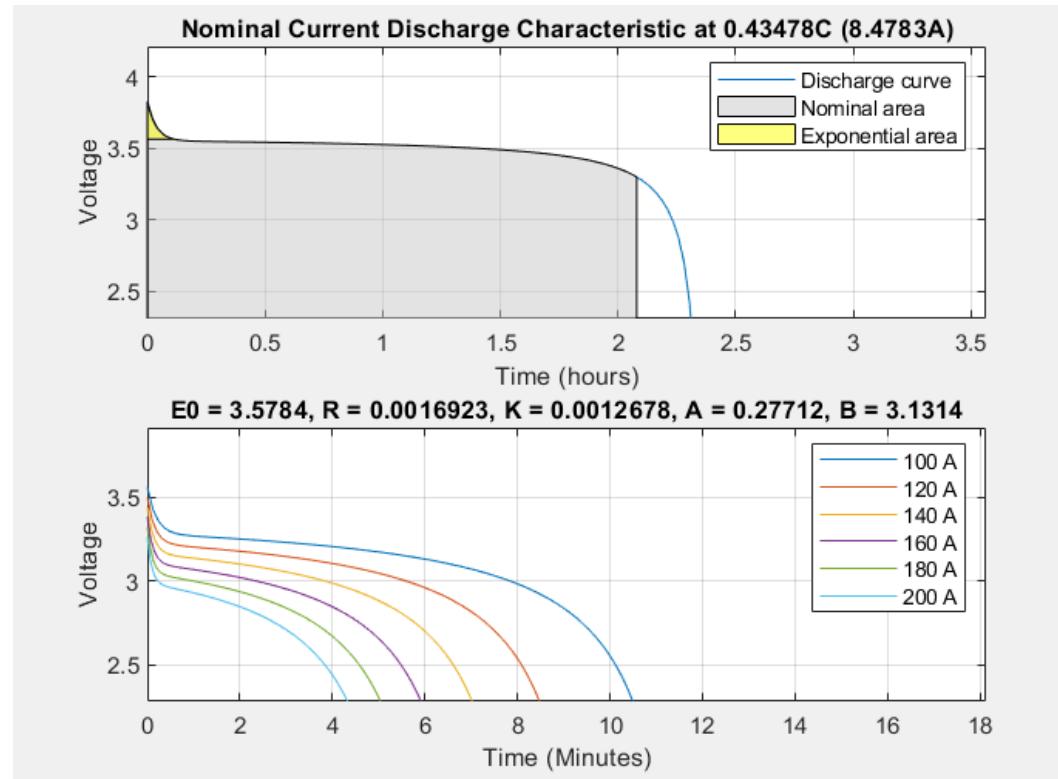


#### 4.4.3 VOLTAGE-DISCHARGE CAPACITY & VOLTAGE-TIME EVOLUTIONS FOR DIFFERENT CURRENTS

It is essential to understand our cell's capacity to hold charge and also its terminal voltage evolution with time. For the following data we have used different continuous discharge currents to see how the interested variables evolve.

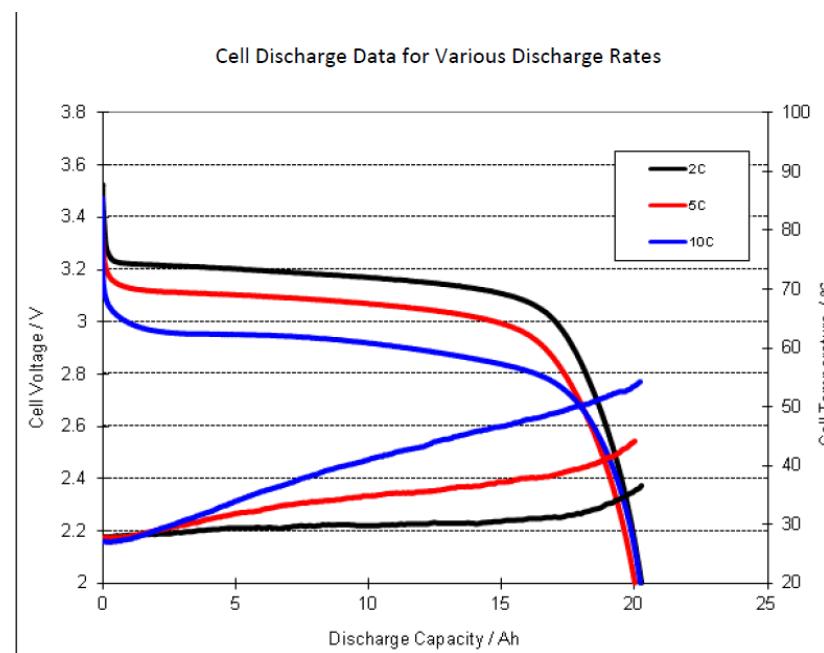


For a single cell we can see that the capacity tends to decrease significantly by about 3Ah on doubling the current requirement. (Cell parameters obtained from datasheet for all the simulations are present at the end of this section)



Now for time-evolution plots of the voltage,

The time evolution is straightforward in the sense that a higher current requirement takes up the cells capacity in a shorter amount of time. For an actual graph from the manufacturer we have



We can see that the first plot almost closely resembles the actual datasheet values.

## 4.5 Battery Management System

### Main Features

- Monitors every cell voltage in series
- Field programmable and upgradeable
- Intelligent cell balancing (efficient passive balancing)
- Enforces min. and max. cell voltages
- Enforces maximum current limits
- Enforces temperature limits
- Professional and robust design
- Monitors state-of-charge
- Retains lifetime data about battery history
- Integration with 3rd party smartphone apps (Torque, EngineLink) and external displays

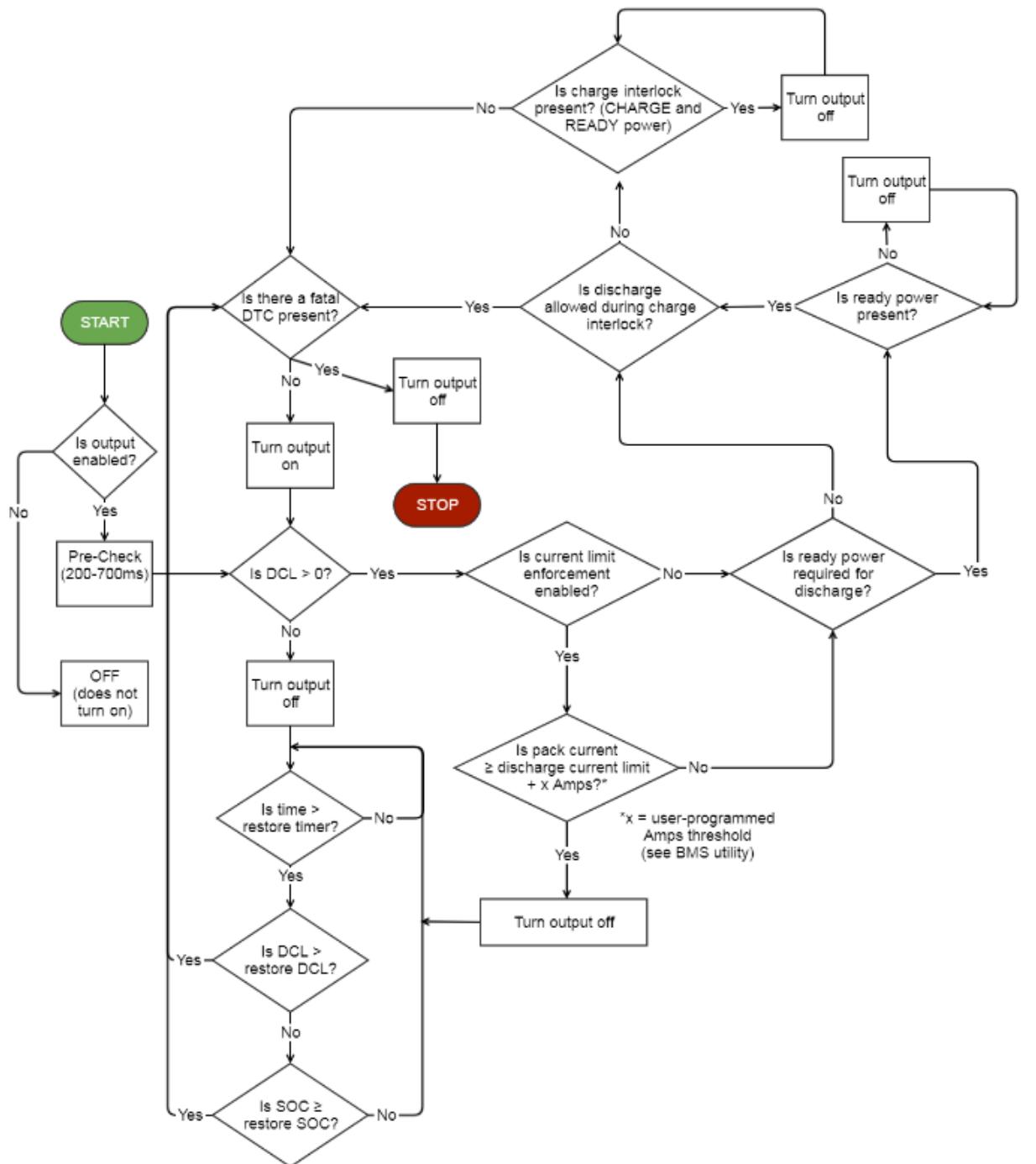
### Battery Compatibility

- Compatible with almost all lithium-ion cells
- One-click setup for common battery types
- Supports 4-180 cells in series per BMS unit (2x additional remote units can be used in series)

### Battery Calculations

- State of Charge (SOC) & Pack Health
- Open-Circuit (sitting) cell voltages
- Charge & Discharge current limits
- Internal resistance (for all cells and total pack)

### **Discharge Enable Output Flow Chart**



*Fig: Discharge enable output flow chart*

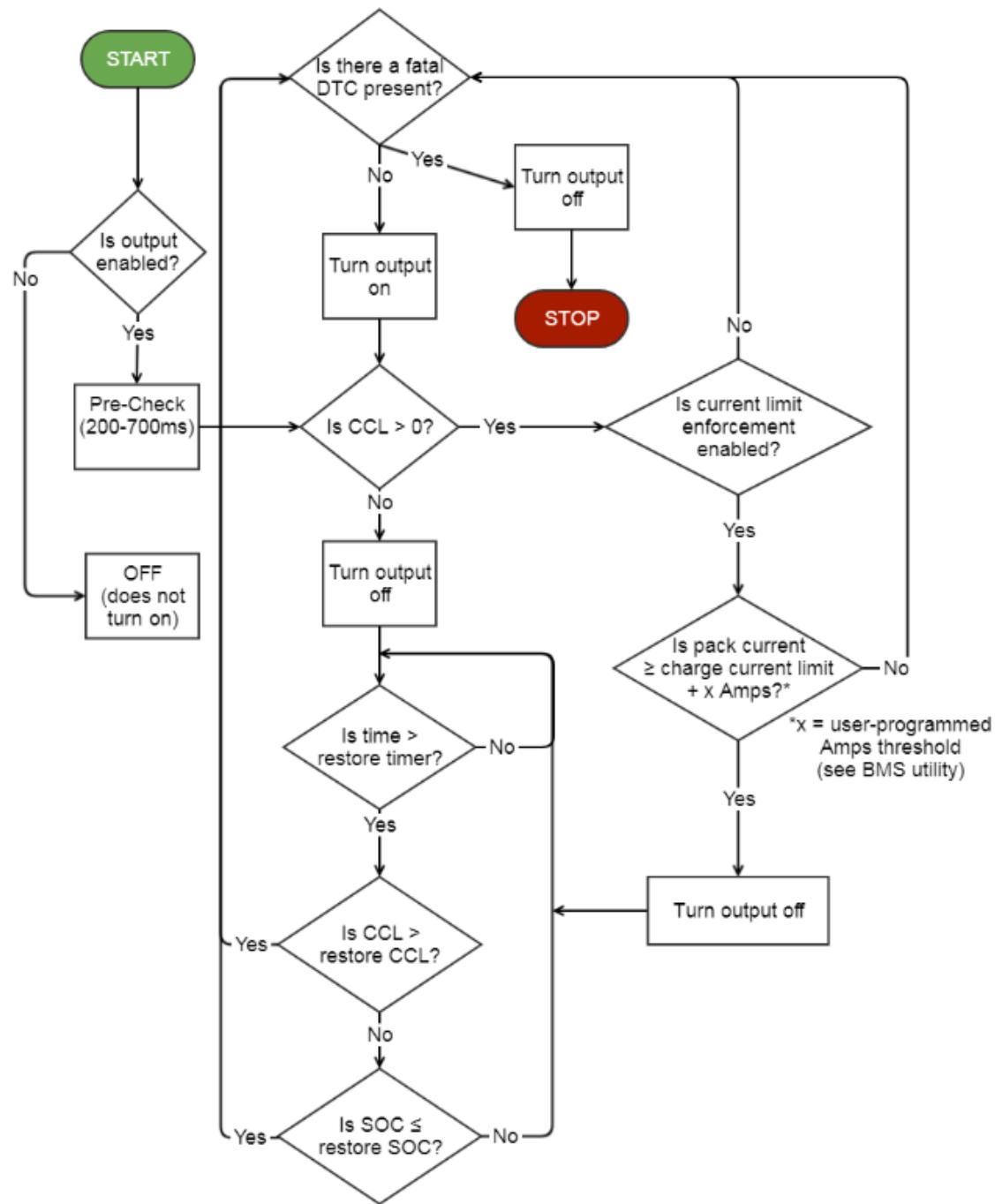
**Charge Enable Output Flow Chart**

Fig: Charge enable output flow chart

## Centralized Design

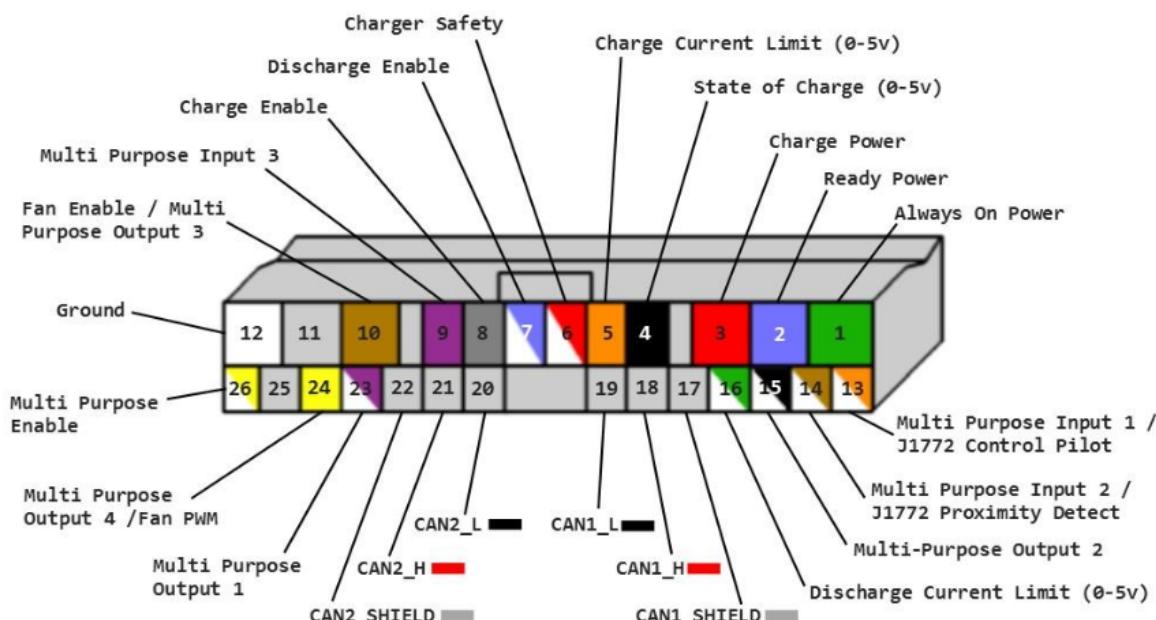
- No cell tap boards or external circuitry
- Fast cell voltage polling (every 25 mS typical)
- High immunity to EMI and other noise
- High accuracy cell voltage measurement

## 2x Programmable CAN BUS Interfaces

- CAN2.0B (11-bit and 29-bit IDs supported)
- Independently operate at different baud rates
- Fully customizable message formatting
- Field upgradeable firmware and settings using either CAN interface
- One-click setup for many common chargers and inverters
- ISO-15765 OBD2 protocol compatible
- Compatible with CAN-Open and J1939 Charger Support
- Integrated support for J1772 charging stations
- Works with J1772 proximity & pilot signals
- Supports CHAdeMO DC fast charging protocol

## Input / Output

- Easy interfacing with chargers and loads
- On/off outputs for controlling charge and discharge sources
- 0 – 5V analog outputs for gradual current reduction (improves usable range of battery)
- Thermal management controls for battery cooling / heating



*Fig: BMS*

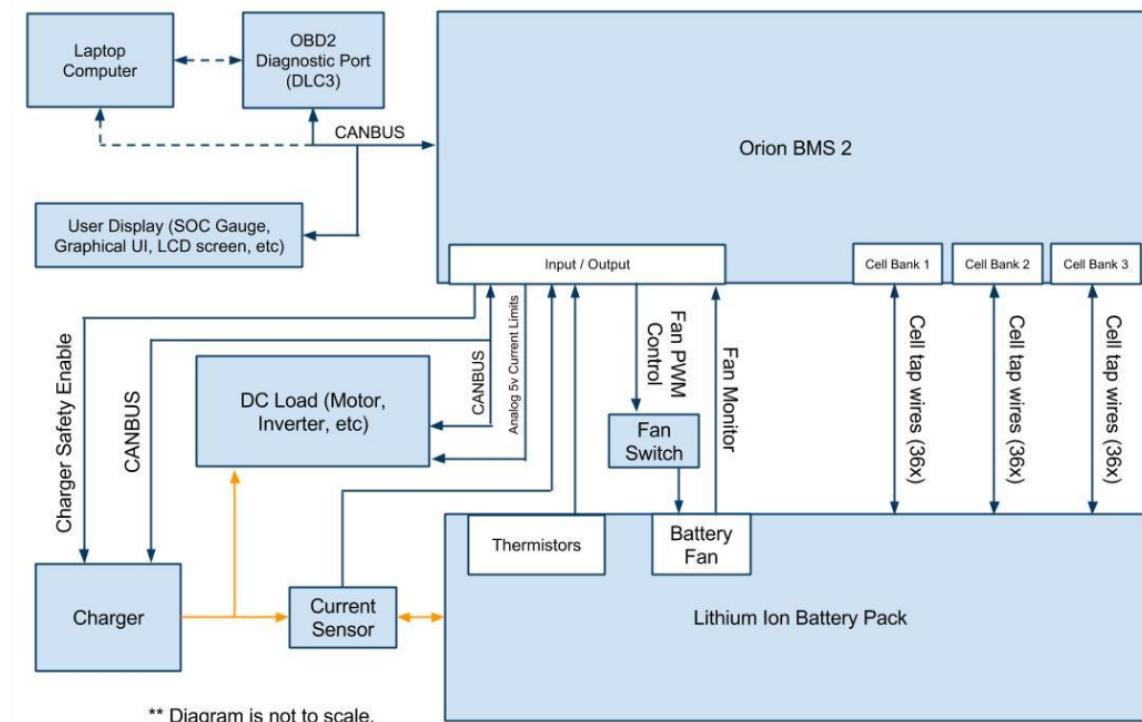


Fig: Schematic of BMS connections

## Diagnostic Features

- Diagnostic trouble codes quickly identify and diagnose battery problems
- Freeze frame data records exact conditions and battery data when a fault occurred
- Supports OBD2 automotive protocol for storing diagnostic trouble codes and polling live data

## Data Logging

- Unit tracks total number of battery cycles
- Records detailed lifetime battery usage and environment conditions internally.
- All BMS parameters can be logged using PC utility software
- Optional WiFi Connect module can record any parameters to a memory card or Internet
- Internal event logging for easy troubleshooting

## Additional Data Logging Methods

While the internal data logging capabilities of the Orion BMS are extensive, there are times when more sophisticated logging is required. There are several ways to do more exhaustive logging of data being produced by the BMS. These additional data logging methods include:

- Data logging through the Orion BMS Utility: The software utility provides multiple ways to take more detailed logs of the battery related data produced by the BMS. First there is the “Live Graph & Data Logging” tab (available at the top of the utility) which allows the operator to specify several live parameter fields to regularly log which is very helpful for logging specific fields or groups of fields. The utility will indefinitely log this information to the computer in a CSV (comma-separated-value) format for easy processing. Second, there is the “Live Cell Data” tab (also available at the top of the utility) which has a dedicated logging function (a record button is present in the bottom right corner of the screen). This performs high speed logging on the individual cell voltages as published by the BMS, in addition to other important parameters

like State of Charge, Pack Current, Charge / Discharge Current Limit and relay output statuses. Logs generated from this screen are also output in a CSV (comma-separated-value) format and are very helpful for general purpose diagnostics.

- For long term data logging where a computer would be less practical, there are 2 expansion module devices that can directly interface with the Orion BMS: The Data Logging Display Module and the Orion Connect WiFi Module. The Data Logging Display Module communicates with the Orion BMS via CANBUS and allows for logging of data to an onboard SD card. It also allows for displaying the State of Charge by an internal LED bar graph. The Orion Connect WiFi Module also allows for logging of data to an onboard SD card in addition to being able to transmit this data to the Orion Connect Cloud servers via WiFi. This allows for data to be viewed remotely wherever an internet connection is available.  
Data logging can also be performed by third party devices. The CANBUS interfaces on the Orion BMS can be fully customized to allow it to conform to a number of standards or protocols.

#### Other Features

- Integrated isolation fault detection circuit
- Multiple remote modules may be used in series
- Automotive grade locking connectors
- Temperature compensation for improved monitoring in different temperatures
- Integrated status LED for indicating faults

## Fault Code Failsafe Chart

DTC Code	Fault Description	Failsafe Behavior
P0A0A	Internal Heatsink Thermistor Fault	Disables Cell Balancing
P0A0B	Internal Software Fault	No Failsafe Behavior
P0A0C	Highest Cell Voltage Too High Fault	No Failsafe Behavior
P0A0D	Cell Voltage Over 5 Volts Fault	Voltage Failsafe
P0A0E	Lowest Cell Voltage Too Low Fault	No Failsafe Behavior
P0A0F	Cell ASIC Fault	No Failsafe Behavior
P0A1F	Internal Cell Communication Fault	Voltage Failsafe
P0A02	Weak Pack Fault	No Failsafe Behavior
P0A04	Open Wiring Fault	Voltage Failsafe
P0A05	Input Power Supply Fault	Input Power Supply Failsafe
P0A06	Charge Limit Enforcement Fault	Relay Failsafe
P0A07	Discharge Limit Enforcement Fault	Relay Failsafe
P0A08	Charger Safety Relay Fault	Relay Failsafe

#### Cell Voltage Monitoring Specs

- Cell voltage measurement resolution of 0.1mV.
- Maximum individual cell voltage rating: 0.5v to 5v per cell tap.
- Cell voltage measurement total error <0.25% across full product temperature range.
- Total pack voltages from 12vDC up to 800vDC (maximum).

- Supports from 4 to 340 cells per battery pack (requires remote modules for more than 180 cells, 800vDC maximum).



## Reliability & EMI Immunity

- Operates through the highest class passenger vehicle load dump ISO 7637 Class IV (178V, 400mS, 0.5 ohm source.)
- Operates through ISO 7637 “cold crank” brownouts down to 5v on input supply rail and can operate > 100mS with no power (with initial voltage of at least 12v)
- Meets EN 50498: 2010 EMC Aftermarket Vehicle Directive
- Meets European UNECE Reg 10.05 (Replaced Road Vehicle Directive) Failure Modes

## Failsafe Modes

The Orion BMS has several failsafe software modes to ensure that the batteries are protected against internal and some external failures of the BMS. These modes are designed to place the priority on protecting the battery.

- Voltage failsafe (non-operating)-This is the most serious failure mode and is triggered when the BMS has determined that it no longer has accurate cell or total pack voltages. This can be caused by an open (disconnected) tap wire (an open wire fault code), any populated cell which is reading a voltage below 0.09 volts or if a cell voltage reads over 5 volts. In a configuration with remote modules, a communication failure between the main BMS unit and any remote unit will also result in a voltage failsafe condition. Because the BMS cannot protect the cells if the accuracy of the cell voltages is compromised, the BMS is forced to enter into a non-operating failsafe mode. When the BMS enters into this voltage failsafe condition, the BMS will begin to gradually de-rate the charge and discharge current limits from their last known value down to 0 to prevent charging and discharging. The amount of time to de-rate the limits is specified in the profile and is designed to provide some usable time of the battery after the failure has occurred. The gradual current limit reductions are intended to alert the operator to the fact there is a problem while providing enough power to allow the application to come to a safe stop. This is particularly useful if the application is an electric vehicle or application where having some available power for a short period of time may be useful. This error condition should always be investigated, and the cause corrected prior to clearing the code or continued use of the battery.
- Current sensor failsafe mode (degraded operation)-This failsafe mode is triggered when the BMS determines that the current sensor is either unplugged or has otherwise become inaccurate and cannot be trusted or if the BMS is configured for no current sensor. In this mode the current sensor is disabled and will measure 0 amps. The BMS will continue to operate and protect the batteries purely using voltage based conditions. However, all functions relying on the current sensor are disabled. Care should be taken to correct this issue as quickly as possible, but it is possible to continue using the battery pack in this failsafe condition. The changes made in this failsafe mode:

- o Internal resistance calculations disabled (both cell and total pack)
- o Open cell voltage calculations disabled for both pack and individual cell calculations. The open cell voltages will read the same as the instantaneous voltage readings. This results in highly inaccurate state of charge drifts.
- o State of charge. This cannot be accurately calculated and will be guessed purely on voltage and based on drift points. Drift points are based on open cell voltages, so SOC will vary considerably and should not be trusted to be totally accurate.
- o Charge and discharge current limits switch to a voltage failsafe calculation mode and may be higher or lower than they should be. However, they will rapidly adjust if voltages approach minimum or maximum levels.
- o Overcurrent protection is effectively turned off (cell voltages may provide some level of current limiting, but this is only based on cell voltages and not measured current). The BMS cannot enforce over current limit protections since current is unknown in the event of a current sensor fault.
- Digital on/off or Relay failsafe- The Relay failsafe mode is triggered when the BMS turns off a digital on/off output, and the BMS continues to measure current flowing into or out of the battery respectively. This failsafe mode is triggered if current does not stop within 500mS after the output has been turned off. In this failsafe mode, all digital on/off relays are turned off and latched off until the fault code is reset or the BMS is power cycled. A diagnostic trouble code is stored when this happens. This failsafe will only activate if the offending relay is enabled in the settings profile (disabled relays are ignored).
- 12 / 24v supply power failsafe (degraded operation)-The BMS requires a nominal 12v-24v input main power to operate properly. The BMS is equipped with an internal voltage sensor. If internal voltages drop too low for continuous operation, the BMS will enter into Input Power Supply Failsafe mode. Orion BMS 2 units can operate through voltage sags (cold cranking) at voltages down to approximately 4.5v for brief periods less than 5 seconds without causing this failsafe mode. This fault will set if the measured input voltage is less than 8v for more than 8 seconds. If this failsafe mode occurs, charge enable, discharge enable and charger safety outputs will turn off. This is done because a contactor powered by the same power source as the BMS may fail to operate, or may intermittently operate at too low of a voltage. In this failsafe mode all digital on/off outputs are set to off and charge and discharge limits are set to zero immediately. The 5V analog outputs may remain active but cannot be guaranteed to be accurate. This failsafe mode will set a diagnostic trouble code which will remain for later diagnostics but will automatically restore normal operations once normal operating voltage has been met.
- Internal memory failsafe (non-operating mode)- In the event of an internal BMS memory failure (i.e. if the memory that stores the profile is damaged), the BMS will load the factory default battery profile with all outputs and inputs disabled to protect the battery. A diagnostic trouble code will be set to indicate this problem has occurred

### Charger Safety Output Flow Chart

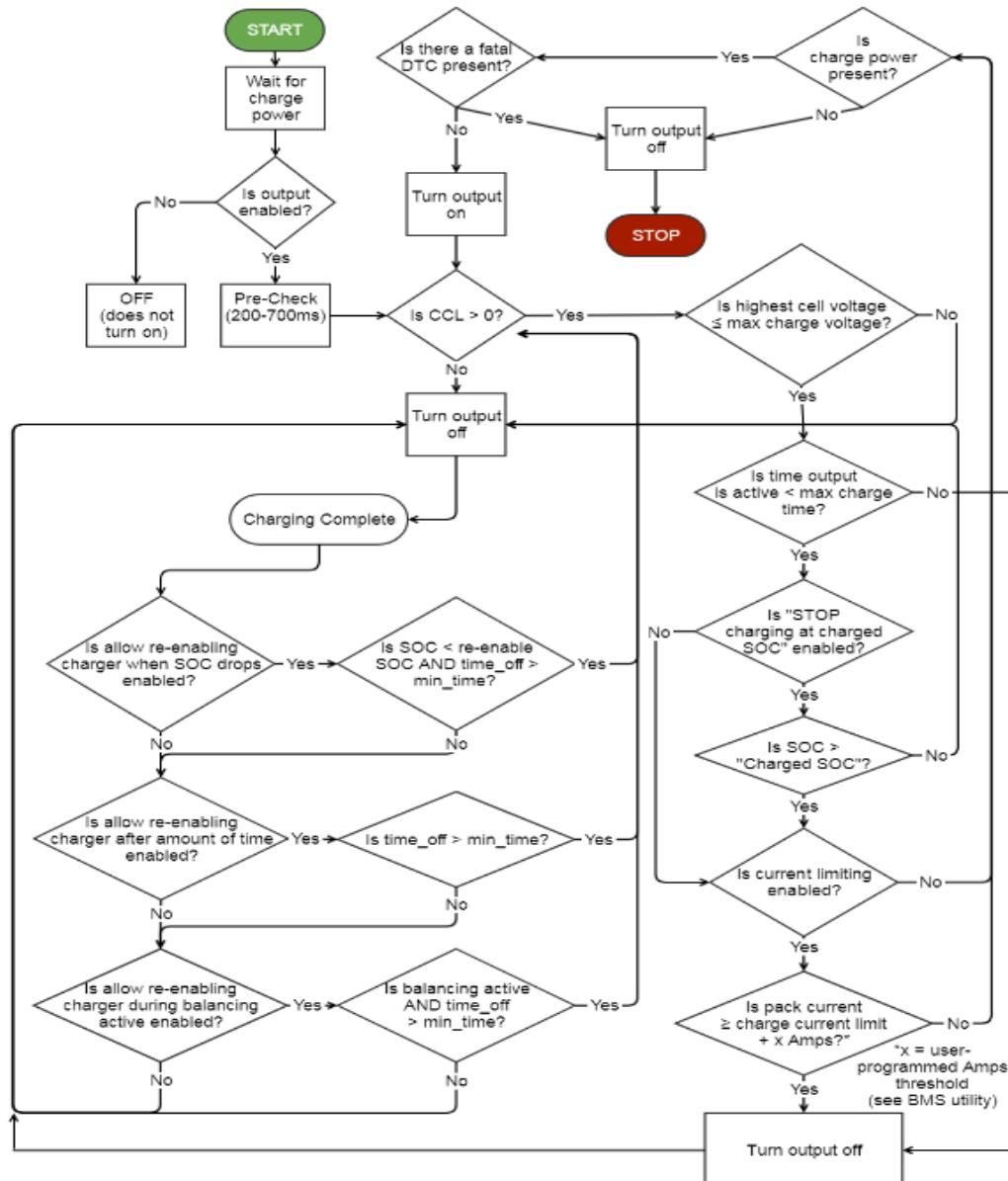


Fig: Charger safety output flow chart

### Product Dimensions & Weight (Typical, With Heatsink)

- 24-72 Cells: 7.15" (W) x 6.72" (L) x 2.37" (H) — 2.50 lbs
- 84-108 Cells: 9.50" (W) x 6.72" (L) x 2.37" (H) — 3.25 lbs
- 120-180 Cells: 15.52" (W) x 6.72" (L) x 2.37" (H) — 4.80 lbs

### Isolation

- Cell taps isolated from input power supply, chassis and I/O
- 2.5kV isolation between each connector of cell taps
- Isolation allows for use of in-pack safety disconnects and fuses
- High voltage isolation fault detection circuit to monitor the breakdown of wire insulation

### Battery Histogram Data

The Orion BMS will also track a number of vital usage statistics and figures across the entire lifetime of the battery pack in order to help installers warranty the battery pack. These metrics include the following:

- Total amount of time the pack has spent at different temperatures (expressed in seconds per 5 degrees Celsius)
- Total amount of time the pack has spent at different states of charge (expressed in seconds per 5% SOC)
- Total amount of time the pack has spent at different charge / discharge C-rates (expressed in seconds per 0.5C increments with higher resolution between -1C and 1C)

## I/O Interfaces

- 2 Digital signal outputs for enabling charge and discharge.
- 1 Digital signal output to control a battery charger
- 5 Digital programmable multi-purpose outputs
- 2 Digital programmable CANBUS (CAN2.0B) interfaces.
- 3 Analog 0-5v outputs that represent the following signals: Charge / Discharge Current Limits and State of Charge (SOC)
- 1 PWM fan output and fan speed feedback monitor (external switch and relay required, uses MPO4)
- 8 Thermistor inputs (Can support up to 800 thermistors through external thermistor expansion modules (sold separately))
- 1 Dual range current sensor input (measures pack current)

## Power Supply

- 3 redundant 12V—24V DC power supplies for reliability
- BMS retains data and settings without power
- Low power sleep mode

## Thermistor Relay Module

### Schematic

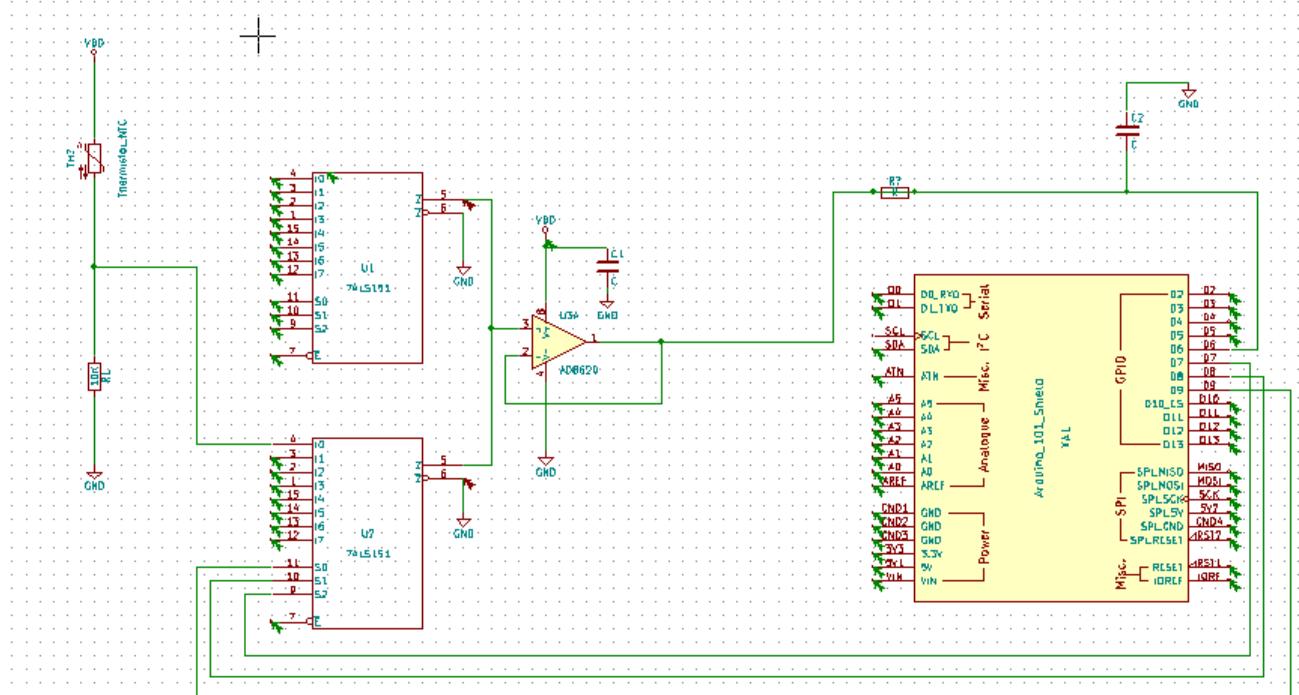


Fig: Thermistor Relay Module Schematic

We have developed a Thermistor Relay Module which involves a setup of 16 thermistors placed near the negative terminals of battery. The thermistor is placed across a 10K resistor and the voltage drop provided by these thermistors is measured and is expressed as a temperature value after suitable calculations. 16 of the sensor arrangements are connected to both the multiplexers (8 each. We chose 74LS151 as 8:1 Muxes are easier to procure). These connections aren't shown in the schematic as they are redundant. The output signal is run through a op-amp running on a negative feedback loop and fed to the microcontroller. The micro-controller receives the voltage measurement and does the following calculation inorder to determine the temperature at the specific thermistor.

The following parameters are pre-defined for the NTC Thermistors and are used for temperature calculation

$$\beta = (\log(RT_1/RT_2)) / ((1/T_1) - (1/T_2))$$

$$R_{inf} = R_0 \cdot \exp(-\beta/T_0)$$

These parameters are used along with the latest measurement of voltage drop to give a temperature value.

$$V_{out} = V_{in} * (\text{MuxOut}) / 1024.0$$

$$R_{out} = (R_t * V_{out}) / (V_{in} - V_{out})$$

$$TempK = (\beta / \log(R_{out}/R_{inf}))$$

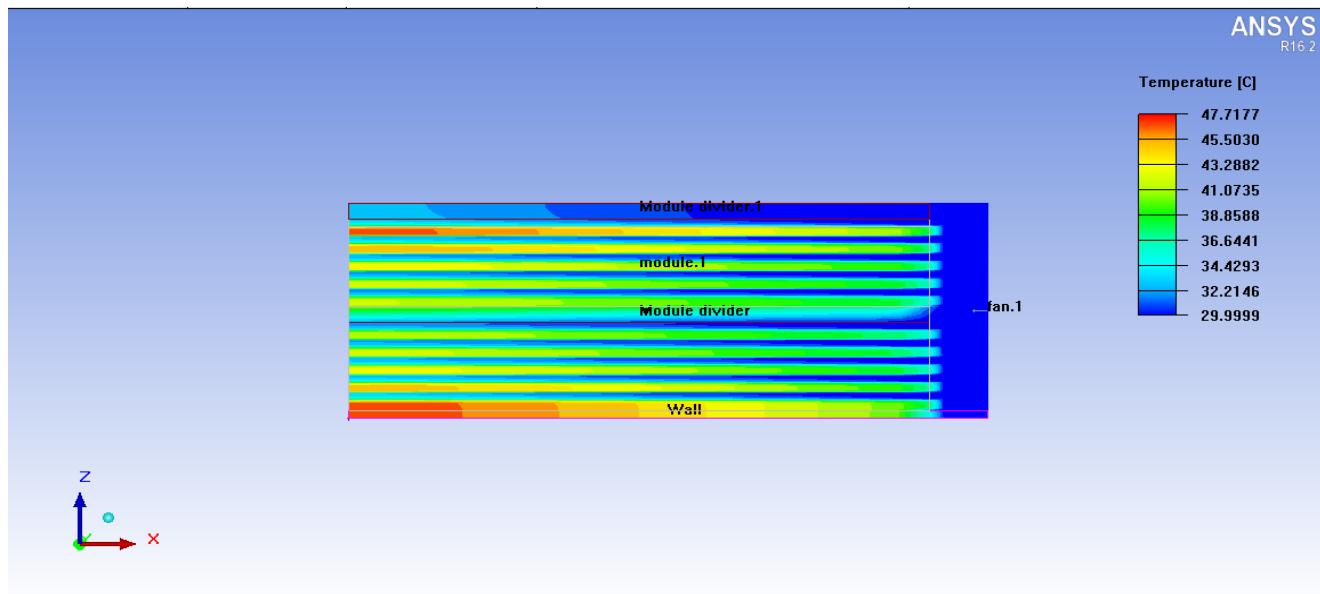
$$TempC = TempK - 273.15$$

The microcontroller sends out a selection signal to the multiplexers (Connections to the second mux not shown in the figure).

Essentially, the mux selects one sensor signal at a time and checks the temperature reading against a predetermined threshold value. In this case, we decided upon a temperature of 55 degrees Celsius.

In case of a signal reading exceeding the threshold, the microcontroller enters a redundancy loop. Here, the system creates a delay of 1000ms and checks for the reading of the anomalous signal. Incase the temperature detected is still above the threshold value, the module sends out a signal to the shutdown circuit. This model can be extended to include greater number of sensors in further editions by daisy-chaining such modules.

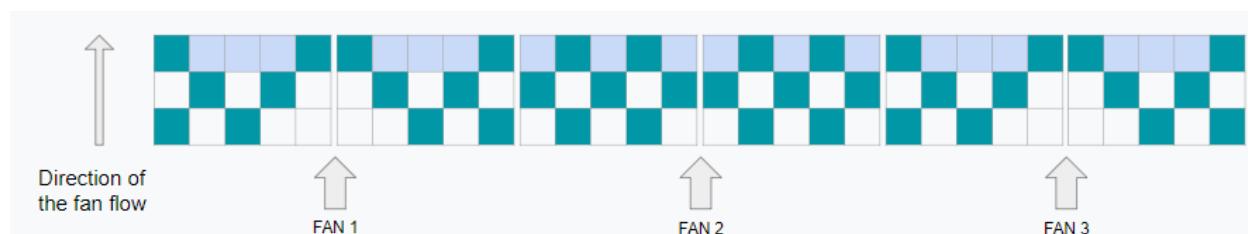
Based on our cooling model, the temperature critical cells were identified in our pack by running ANSYS Icepak Simulations to generate surface temperature plots and as expected, the cells that were farthest away from the fans and corner cells towards the end of the module, that received the least amount of fan's airflow and hot air from the rest of the cells, were found to be lying in the maximum temperature range.



*Fig: Ansys simulation for surface temperature plots*

While covering all the critical cells, the temperature sensors' layout is designed keeping the minimum requirement of measuring at-least 30% of Li- ion cells in mind, while also keeping the distribution uniform. Each temperature sensor monitors one cell.

The highlighted spots in the top view of our battery pack, represent sensor- monitored cells, 56 in number and make 62.22% percent of the total cells, out of which the dark marked cells (42%) are evenly distributed.



We plan to design a multiplexer that acts as an expansion module in order to accomodate thermistors beyond the 16 already allowed for in the Orion BMS 2.

## 4.6 Container and Module Design

- Initially we were planning to use aluminium to make the accumulator container as aluminium is less dense than steel and we thought that the aluminium container would be considerably lighter than a steel container. However, due to height constraints we had to switch over to steel as the walls of the container can be made thinner and the overall height of the container would fit below the chassis side members.

- When the weight of the steel and aluminium containers were compared it was seen that both the containers had a similar weight of approximately 23kg. The accumulator has a total of 90 cells in 90s1p configuration. The 90 cells are divided into 6 modules of 15 cells each. All the cells in each module are connected in series and all the modules are connected in series as well.

- Each module has 3 columns and 5 rows of cells. According to the EV5.3.2, each module can weigh at most 12 kg, have a maximum voltage of 120 V and maximum energy of 6 MJ, all of which are fulfilled with the 15 cells per module configuration. Each module consists of a steel rack which holds the 15 cells.

- The rack is fastened to the internal vertical walls of the container using bolts so that the module is held in place and withstands the accelerations that the car may experience in case of a crash, as mentioned in EV5.5.9. The rack also helps in maintaining an air gap between the cells, which is needed for cooling. Cool air will be pushed through these gaps by the fans.

- Each module is sandwiched between 2 FR4 sheets which is a fire retardant material and 2 nomex sheets which provide electrical insulation. This is in accordance with EV5.4.7. Each module is separated by the internal vertical walls of the container, which are made of steel. The walls help in mounting the modules to the container.

- Each module is connected to the next one with the help of RADLOKs which are required as per EV5.4.5. The top of all the modules is covered by a polycarbonate/FR4 sheet which acts as an insulator and prevents any small metallic debris or tools from falling onto the cells and shorting them (EV5.4.7)

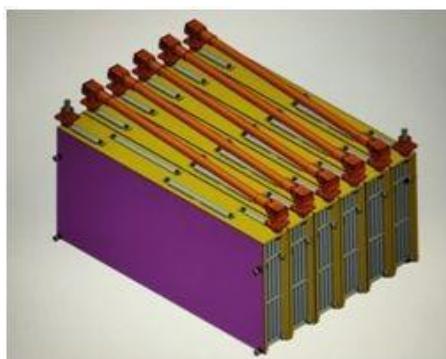


Fig: Isometric View 1



Fig: Isometric View 2

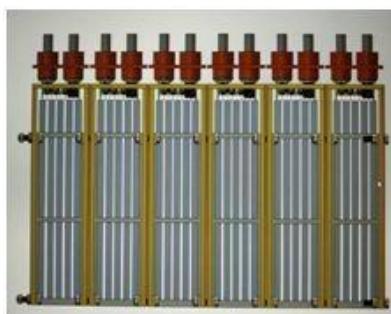


Fig: Orthographic View 1

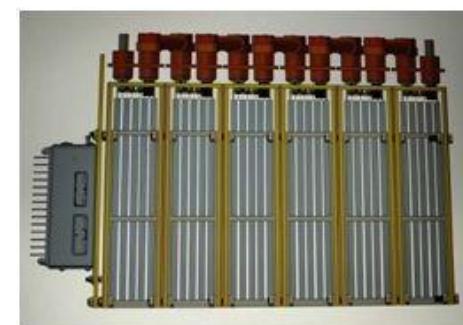


Fig: Orthographic View 2

CAR #03

4th FSEV CONCEPT CHALLENGE

Birla Institute of Technology and Science, Pilani

[inspiredkartersfs@gmail.com](mailto:inspiredkartersfs@gmail.com)

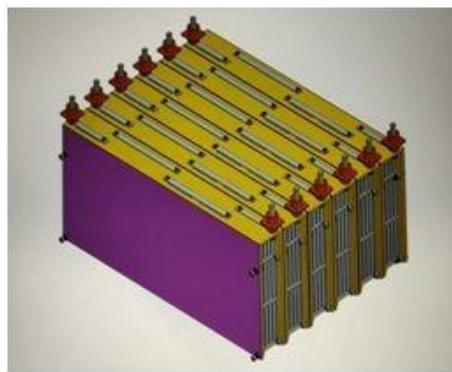


Fig: Isometric View 1 with Radloks

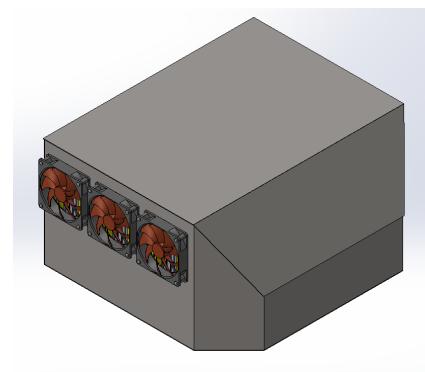


Fig: Final Battery with Container and Fans



Fig: A module with cell mounts



Fig: Fully Constructed Module without Radloks

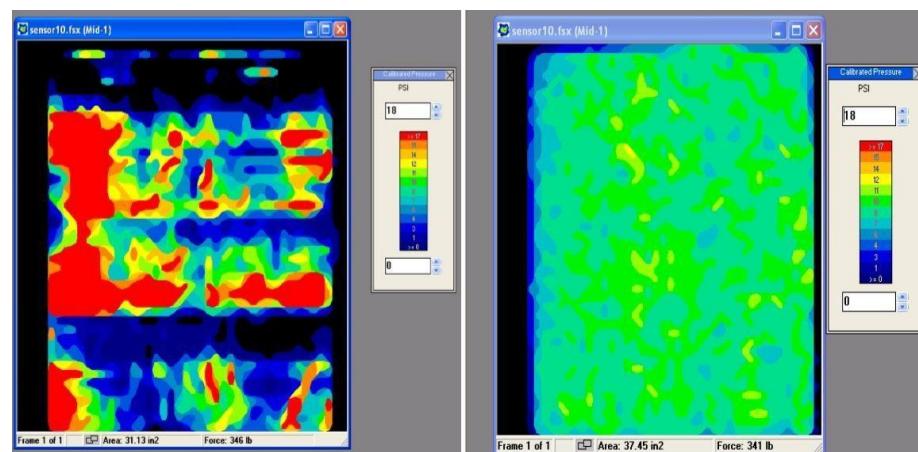


Fig: Pressure on cell face without neoprene and with neoprene

CAR #03

4th FSEV CONCEPT CHALLENGE

Birla Institute of Technology and Science, Pilani

[inspiredkartersfs@gmail.com](mailto:inspiredkartersfs@gmail.com)

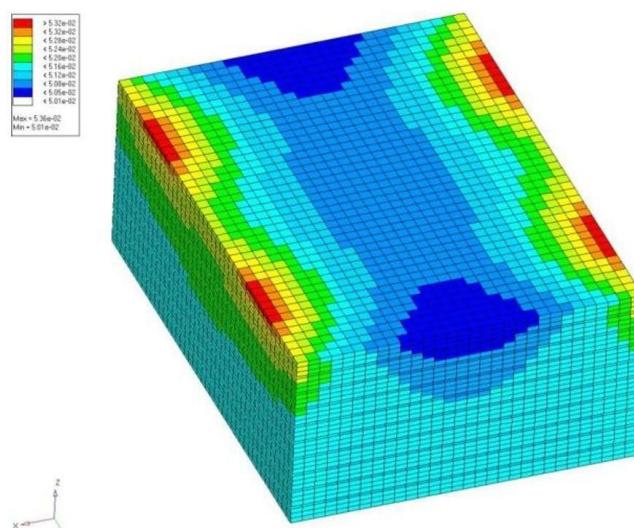


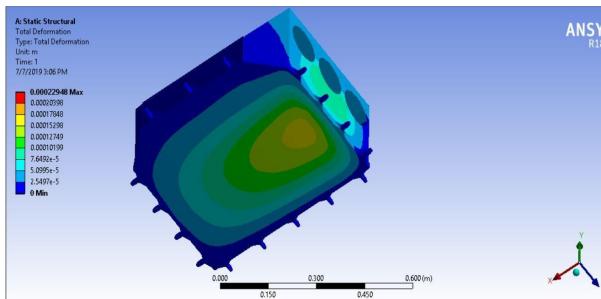
Fig: Pressure map across the surfaces of each cell in a stack

#### 4.6.1 Stress Simulations on container and it's mounts:

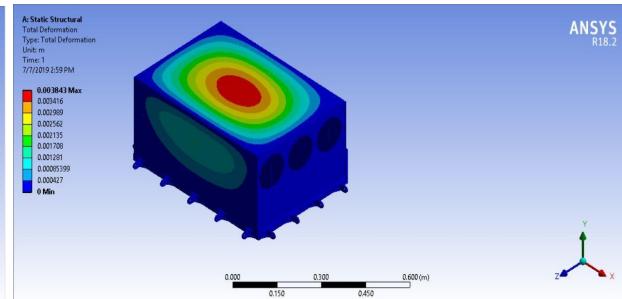
According to FSAE Rules the cell/segment mounting system must be designed to withstand the following acceleration:

- 40g in the longitudinal direction (forward)
- 40g in the lateral (left/right)
- 20g vertical (up/down) direction

##### Stress Simulation on container:



View1: 20g Vertical loading (safe deflection)



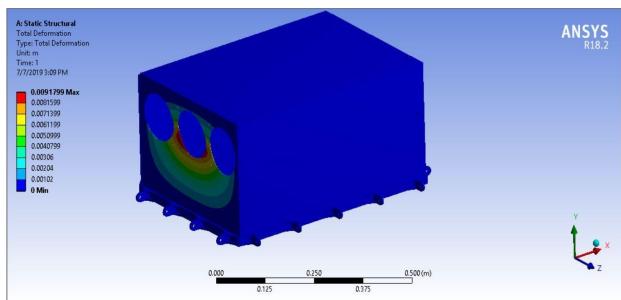
View2: 20g vertical loading (safe deflection)

CAR #03

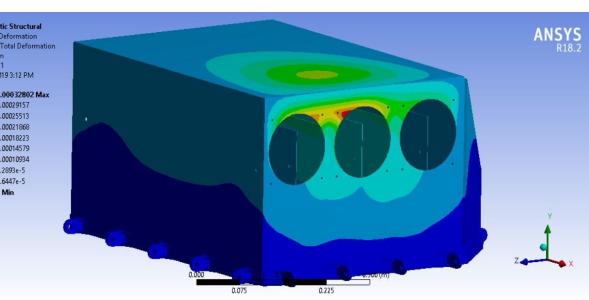
4th FSEV CONCEPT CHALLENGE

Birla Institute of Technology and Science, Pilani

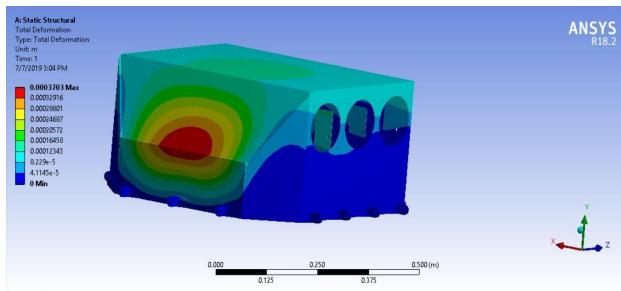
[inspiredkarters@gmail.com](mailto:inspiredkarters@gmail.com)



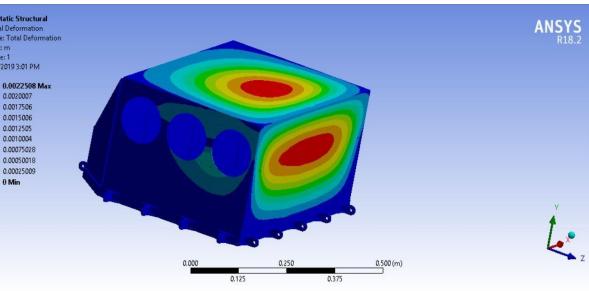
View1: 40g lateral (safe deformation)



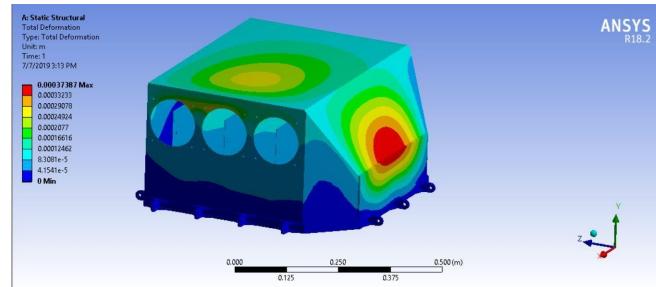
View2: 40g lateral (safe deformation)



View1: 40g longitudinal (safe deformation)



View2: 40g longitudinal (safe deformation)



Combined (40g lateral+40g longitudinal+20g vertical) safe deformation

The analysis is done by providing fixed support the bolt holes for container mounting. The particular analysis is done to make sure that the accumulator container remains intact and the cells are not damaged under impact or collision. The above images clearly portrays that the accumulator pack can withstand the force of 20g and 40g and thus is successful in the designing.

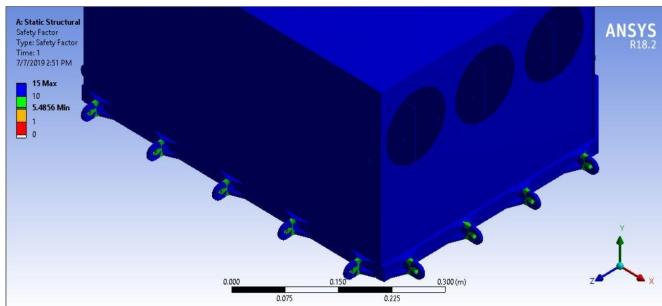
CAR #03

4th FSEV CONCEPT CHALLENGE

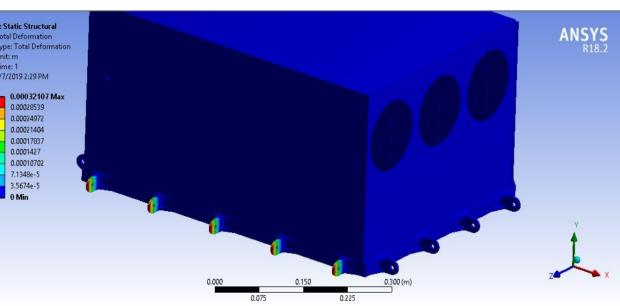
Birla Institute of Technology and Science, Pilani

[inspiredkartersfs@gmail.com](mailto:inspiredkartersfs@gmail.com)

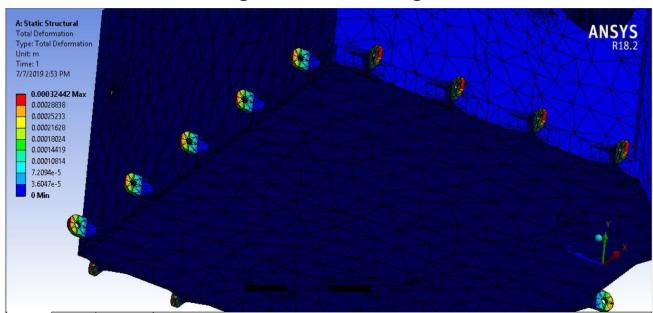
### Stress Simulation on mounts:



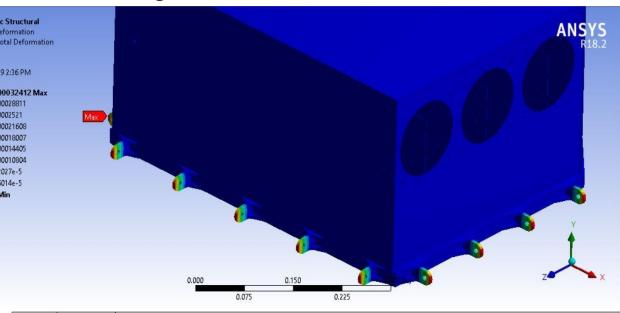
20g vertical loading



40g lateral deformation



Combined (40g lateral+40g longitudinal) deformation



Combined (20g vertical+40g lateral+40g longitudinal)

The stress simulation on mounts were done by keeping the container fixed and then applying the necessary required force in the required directions. The deformation is shown pictorially above and it can be successfully concluded that the mounts are rigid enough to withstand the forces. Hence the mount design is also successful design.

## 4.7 Charger

After going through various chargers, the best suited for our battery was selected. Model number- **DA3K3M17-360**.

It has the following features:

- Constant power and constant voltage state automatic conversion, effective saving time
- Active power factor correction  $\geq 0.99$
- Input over/under-voltage protection
- Output over/under-voltage protection
- Output over-current protection
- Output reverse connection protection
- Over temperature protection
- CAN communication intelligent fault alarm and protection function
- all volume, large power density
- Control guidance function connected with charging pile(optional)
- Functions of remote detection, burning, fault inquiry, etc.

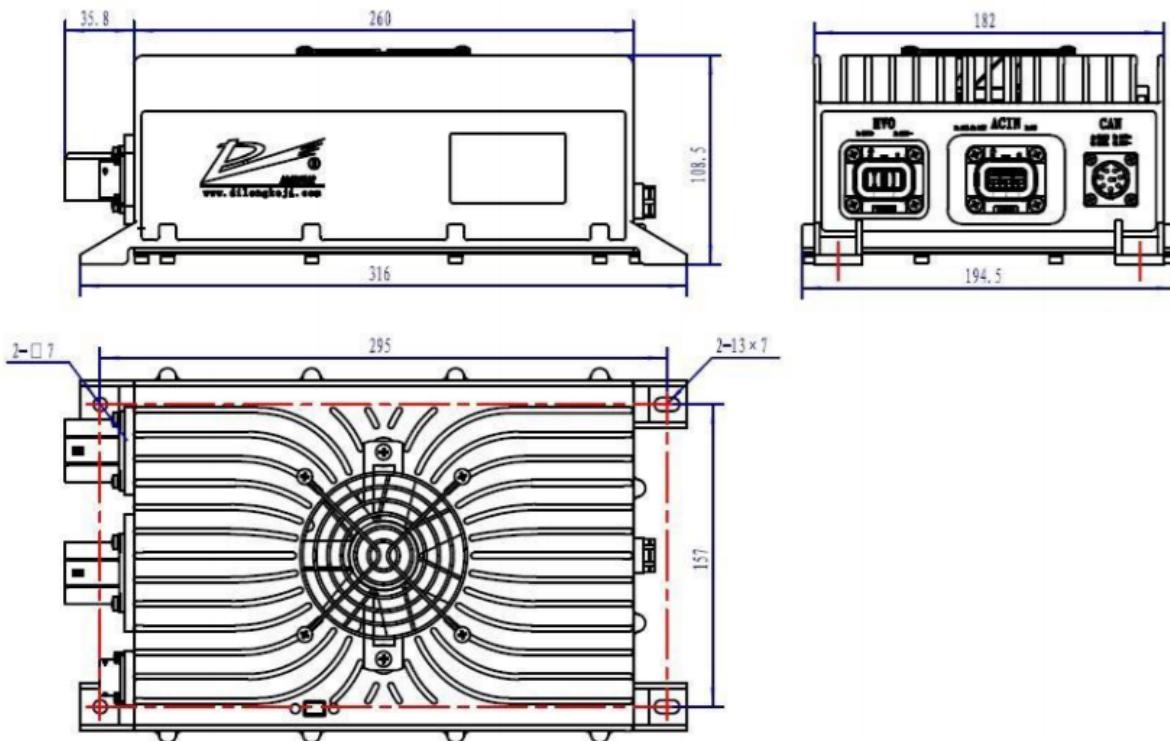
Specification/ Unit		Model	
		DA3K3M17-360	
Mains Input	Rated Voltage	Vac	220
	Voltage Range	Vac	85 ~ 264 (Input 85-176, automatically drop to 1200W)
	FREQ RANGE	Hz	47~63
	Efficiency(max)	%	$\geq 95$
	Power Factor		$\geq 0.99$ @220Vac, rated power
	Current(MAX)	A	$\leq 16$ @220Vac
Mains output	Rated Output Voltage	Vdc	360
	Constant voltage output range	Vdc	250 ~ 420 (According to the customer battery final decision)
	Max Current	A	12
	Constant power output	W	3300
	Output voltage accuracy	V	$\leq \pm 1\%$ Vo input 176Vac-264Vac, output 10%-100% load
	Line regulation	%	$\leq \pm 0.2\%$ Vo input 176Vac-264Vac, output 100% load
	Load regulation	%	$\leq \pm 0.5\%$ Vo input 220Vac, output 10%-100% load
	Ripple and noise	mV	$\leq \pm 3\%$ Vo 20M oscilloscope Twisted-pair cable test.

Charger Specifications

Auxiliary supply	Rated output voltage	Vdc	12
	Rated output power	W	2 (Activate BMS)
Property	Input over-voltage protection	Vac	$\geq 275$
	Input under-voltage protection	Vac	$\leq 80$
	Output Over-current protection	A	$I_o \geq 120\%$
	Output under-voltage protection	Vdc	$\leq 200$
	Output over-voltage protection	Vdc	$\geq 421$
	Over-temperature protection	°C	$85 \pm 5$
	Output short circuit protection		yes (CPU intelligent control)
	Reverse polarity protection		yes (CPU intelligent detection)
	Dynamic response	us	$\leq 200$
	CAN communication		No CAN communication
	Operating temperature		-40 °C ~ +85 °C ( Intelligent derating shell temperature above 70 °C )

Work environment	Storage temperature		-45 °C ~ +105 °C
	Operating humidity	%RH	20-90 (No condensation)
	Storage humidity	%RH	10-95 (No condensation)
	Vibration		QC/T413-2002(Other parts required)
	Cooling mode		Air cooling
Safeguard	Isolation Voltage		shell → input → output: 2.8KVdc; Output → shell: 2KVdc; Auxiliary Output → shell: 700Vdc
	Insulation resistance	MΩ	input → output → shell > 10 (25 °C, 70% RH)
	IP Grade		IP67
Structure	mode of connection		Waterproof aviation connector
	weight	Kg	5
	Size	mm	316*194.5*108.5

## Charger Schematics



## Charger Images



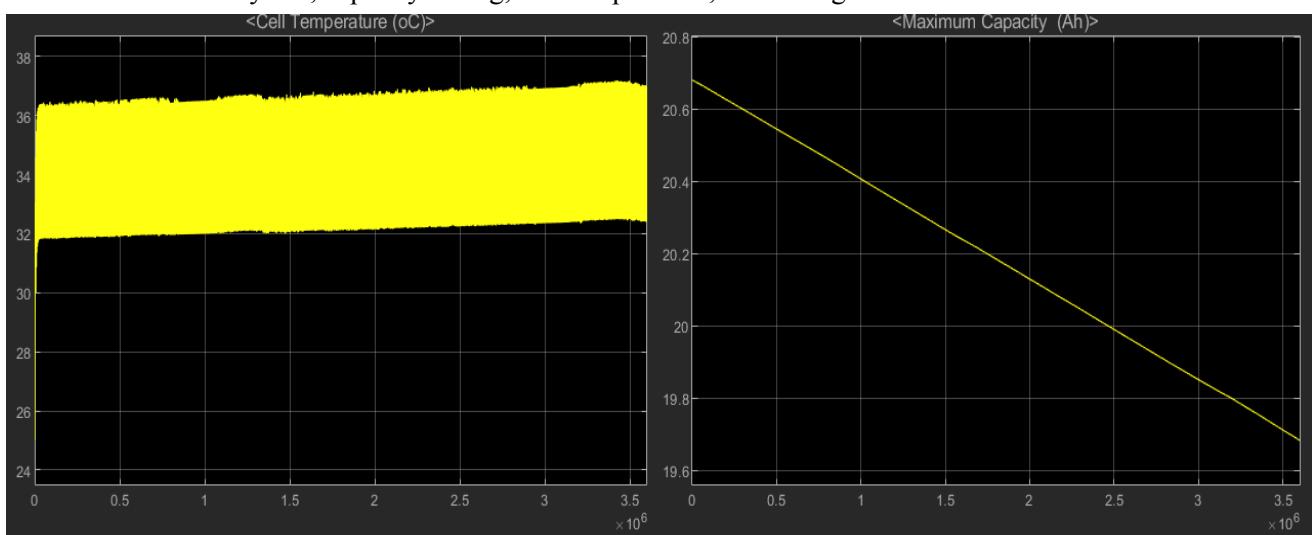
## 4.8 SINGLE CELL AGEING MODEL AND BATTERY CAPACITY EVOLUTION

We used the pre existing battery model along with a discharge profile and automatic charging logic to simulate continuous charging and discharging of the battery for almost 1000h (more than a month) which covers our testing and assemblage timelines so as to provide practical results towards ageing and capacity decrease of battery.

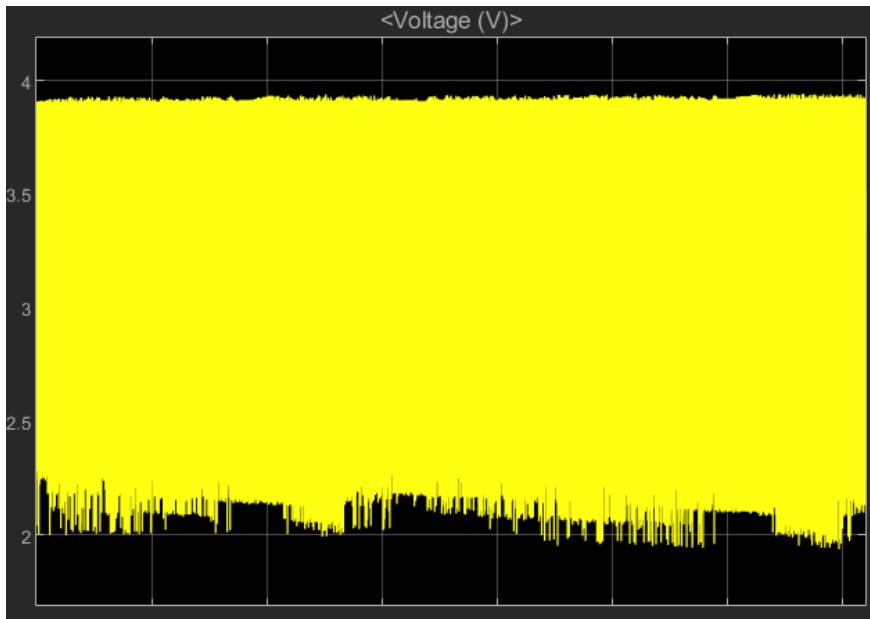
Capacity fading of Li-ion battery can occur due to variety of reasons (it is a research area), mostly due to the transition metal ions which interact with a region of the battery called the solid-electrolyte interphase, which forms because of reactions between the highly reactive anode and the liquid electrolyte that carries the lithium ions back and forth. For every electrolyte molecule that reacts and becomes decomposed in a process called reduction, a lithium ion becomes trapped in the interphase. As more and more lithium gets trapped, the capacity of the battery diminishes. Another theory for capacity fade attributes it to the electrode resistance varying with age (D. Zhang et.al.). In this paper, the authors attribute total cell impedance rising with age as a major factor which affects the total capacity of the cells. We know that cell impedance varies with SOC and is higher in the discharged state.

### 4.8.1 MODEL-1

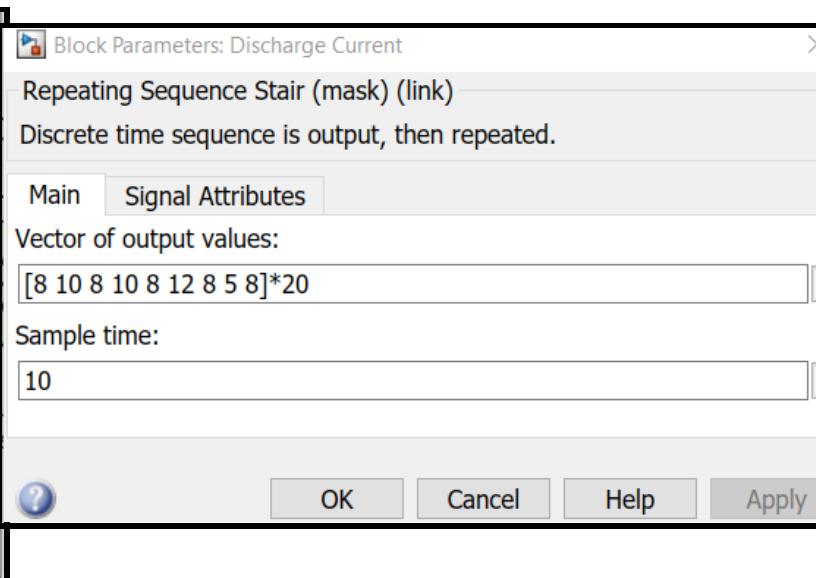
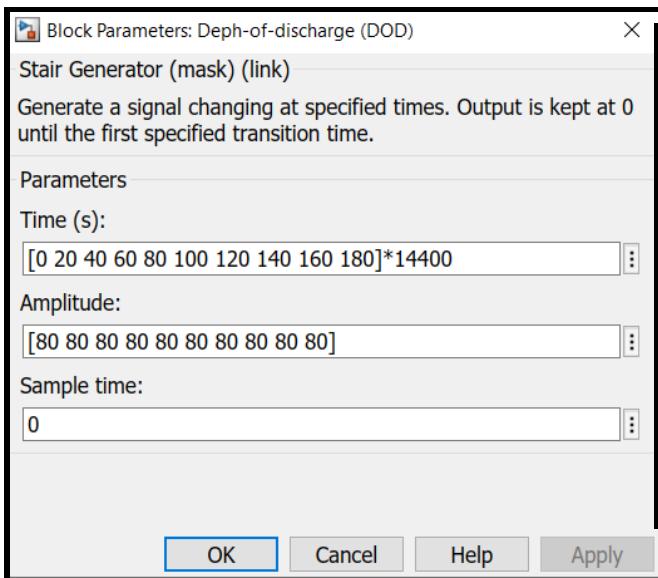
The first model uses a repeating discharge cycle for 1000h to measure the battery's response in terms of No. of full cycles, capacity fading, cell temperature, cell voltage etc.



As a repeating discharge profile is applied to the cell its maximum capacity is diminished as its age(equivalent full cycles) increases. A linear relationship is seen here as the age increases and the capacity decreases. Capacity fade is not yet showing evidence of having heavy discharges accelerating the same.



Above images show the results obtained on discharging the cell according to a fixed discharge profile given below:



In the above parameters we have used a constant Depth of Discharge limit of 80 i.e. SOC of 20% which is considered to be a safe limit for EV Discharging. Discharge profile is not aimed at being used to simulate real time racing demand but is a pragmatic EV discharge requirement for the racing conditions needed.

## **RESULTS**

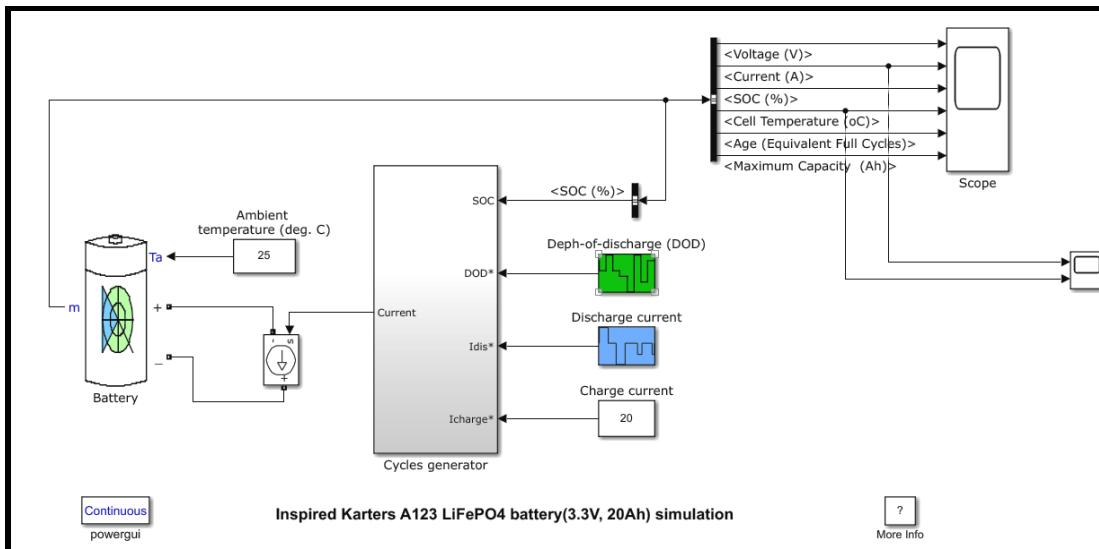
The cell temperature is a healthy 35-36° Celsius at best and cell voltage hovered around the healthy limits of 3.8-2V which is approved by the manufacturer

A noticeable capacity fade from about 20.65Ah(maximum at highest possible cell voltage) to about 19.8Ah. This has occurred while the battery was charged and discharged about 250 times. Cell manufacturer has specified that about 80% BOL capacity will be attained by the end of 5000 cycles. This is a conservative estimate upon using nominal charging current of 20A and staying within the voltage limits of the battery.

In the above model we have used 20A nominal charging current.

### 4.8.2 MODEL-2

Here we are also presenting a few more model results on a different discharge profile in which we use a constant discharge current until the battery reaches 20% SOC. This can be used to model the acceleration event, motor bench testing events, etc.



 Block Parameters: Discharge current X

### Stair Generator (mask) (link)

Generate a signal changing at specified times. Output is kept at 0 until the first specified transition time.

Parameters

Time (s):  
 [...]

Amplitude:  
 [...]

Sample time:  
 [...]

 Block Parameters: Depth-of-discharge (DOD) X

### Stair Generator (mask) (link)

Generate a signal changing at specified times. Output is kept at 0 until the first specified transition time.

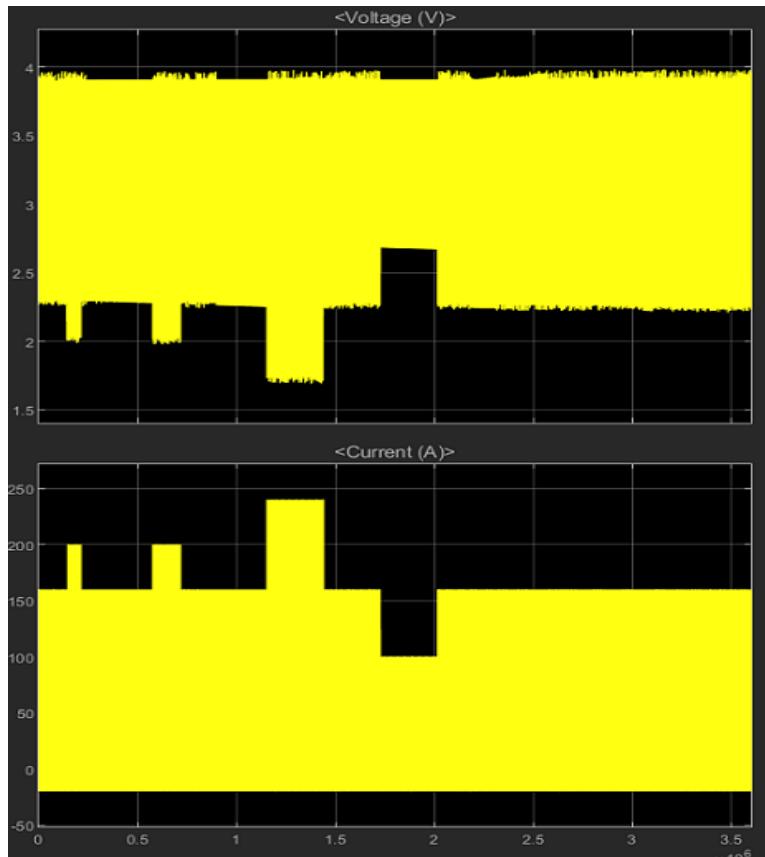
Parameters

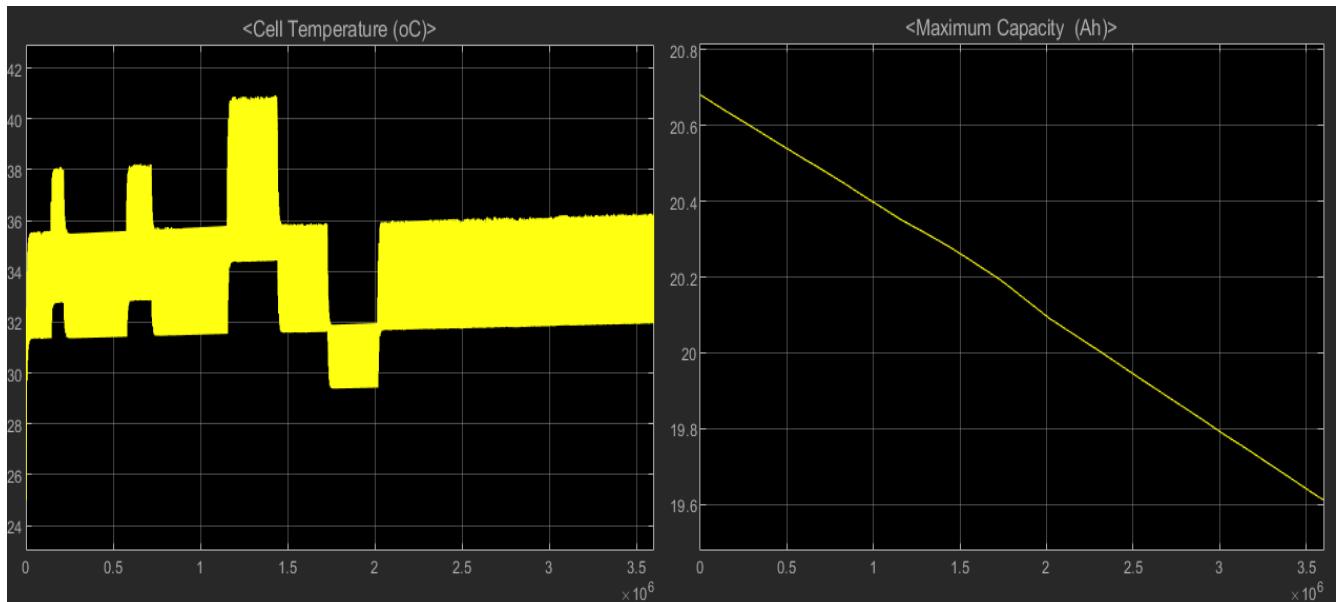
Time (s):  
 [...]

Amplitude:  
 [...]

Sample time:  
 [...]

Results are as follows:



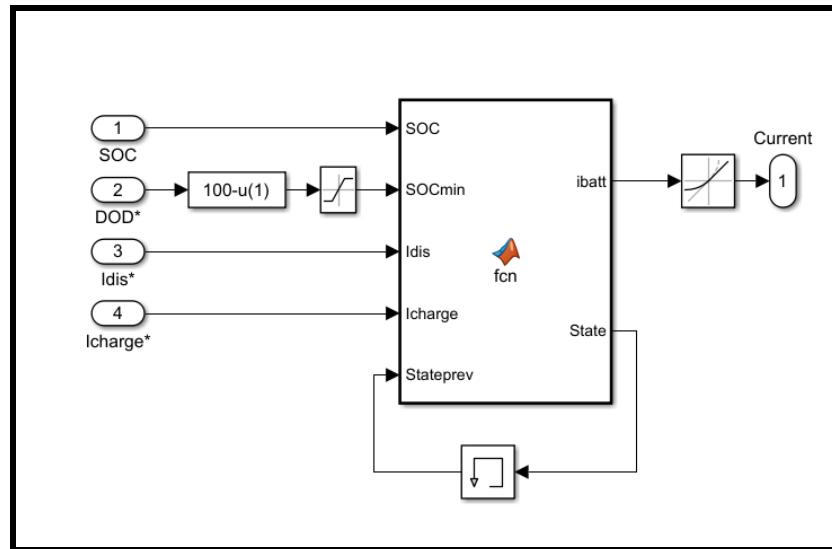


## RESULTS

Results are pretty straightforward :

1. Heavy discharging currents cause a raise in cell temperature. Maximum allowed temperature is 60°C, and even at a high discharge of 10C the cell is maintaining at about 40°C
2. While in a heavy discharge situation the temperature rise can affect open circuit voltage of the cell and hence since most Li-ion cells have a negative temperature coefficient, we can see that the voltage drops below the nominal safe limit of 2V when the cell temperature reaches 40-42°C
3. As opposed to the earlier discharge profile we can see here that the heavy constant charge discharge cycle can create sustained high temperature fluctuations, more than the repeating current discharge profile used in the first model
4. Wrt the first model the results of this record a heavier drop (about 0.2Ah) in capacity for the same number of cycles

## 4.8.5 MODEL EXPLANATION

Cycles generator

Cycles generator is a MATLAB function which discharges and charges the battery, factoring into account the SOC, Depth of Discharge specified and the discharge current.

```
function [ibatt,State] = fcn(SOC,SOCmin,Idis,Icharge,Stateprev)
%#codegen
State=Stateprev;
if(Stateprev==1 && SOC<=SOCmin)
    State=0;
end
if(Stateprev==0 && SOC>=99)
    State=1;
end
if (Stateprev==1)
    ibatt=Idis;
else
    ibatt=-Icharge;
end
```

This code takes care of the charging and discharging depending upon the DOD and Idis limits set by us. The State and Stateprev variables are used to identify the current and previous states in terms of 1 and 0(0-charging)

and 1-discharging). The memory block stores the current state, generates a time lag and makes it the previous state.

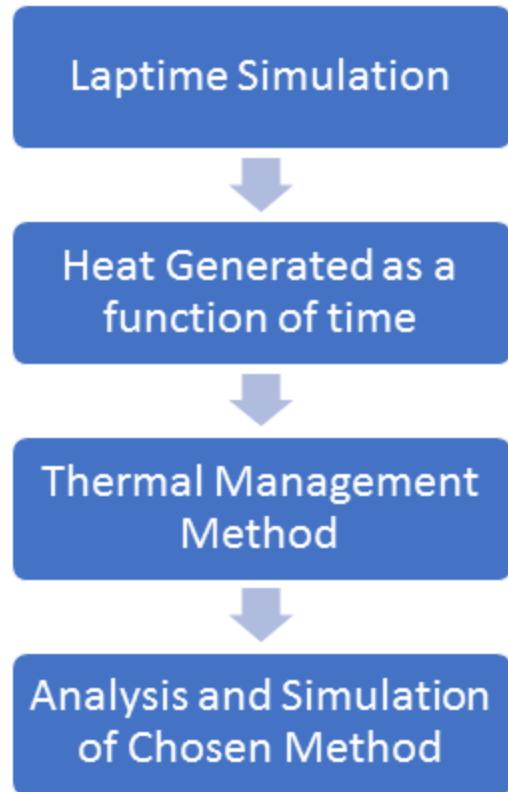
## 4.9 Thermal Management

### 4.9.1 Design Objective

The primary aim to keep the temperature of all cells under the safe limit of 50C (safe limit is 55C, buffer of 5C)

- Minimize the spatial requirements of the accumulator box. This is important as the accumulator box majorly influences the final size of the car(trackwidth and wheelbase)

### 4.9.2 Design Process Flowchart:



#### 4.9.3 Heat Generated by the Battery

The design starts from an understanding of the cooling requirements of the battery. This is understood by looking at the Lap Time simulation(relevant section). The simulation was carried out for different tracks and the corresponding Current v Time profiles were generated. The heat generated was then calculated from the current profile using Joule's heating effect formula. The average current is 47A. The average heat generated was estimated to be 11.37 W for each cell based on the internal resistance of each cell(The manufacturer gave us an estimated resistance of 2.5mOhms). We were not able to test the cells ourselves, this the design is based on the above value

### Current v Time for a Single Lap

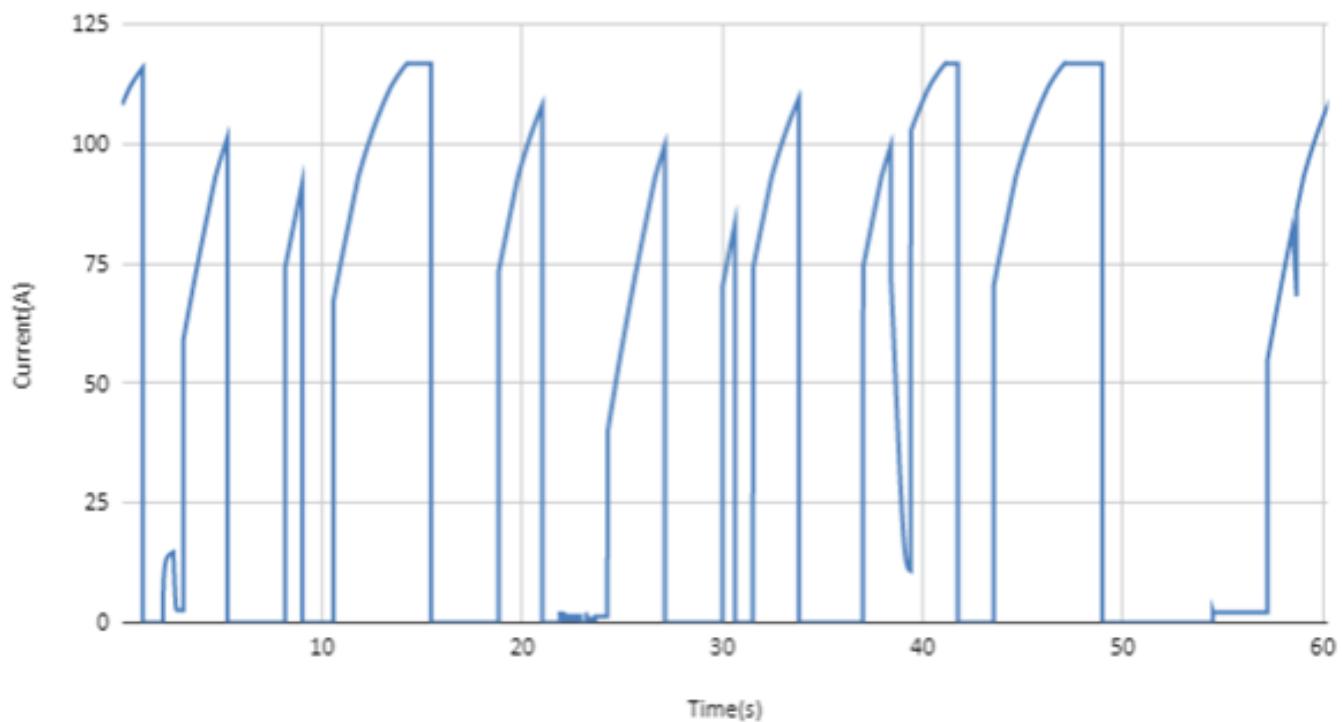


Fig. :Current vs Time simulated from the Lap Time Simulation for a single lap of endurance.

#### 4.9.4 Cooling Solution:

Multiple cooling solutions were considered to manage the battery temperature under the required limits. The following solutions were considered:

- Liquid Cooling: There are multiple methods of liquid cooling. But the basic idea of each method is to create hydraulic system where you pass a certain liquid into the accumulator system which absorbs heat generated through convection and the liquid is then cooled using a radiator type cooling system. An example of such a cooling is engraved fins on the cell where fluid is passed and a hydraulic circuit as described previously is formed. The advantages of the system is that liquid based cooling systems are incredibly powerful in their heat transfer capabilities. They have a significantly higher thermal conductivity and are hence able to absorb heat much better. The major problem with such a system is reliability ,safety and complexity. It is generally very

complex and unsafe to pump water and handle hydraulics inside an accumulator and hence this idea was discarded.

- **Convection Based Air Cooling:** The basic idea is to use either natural or forced convection to create airflow inside the battery pack to extract the heat that is generated. The major advantages of this method is the simplicity of implementation. Air as a coolant is also readily available. The major disadvantage of such a system is it compares poorly to a liquid cooling solution terms of just raw cooling power.

Other cooling solutions like implementation of a passive heat sink was also considered. Thus it was decided to implement a convection based cooling because of the simplicity and ease of implementation. If the cooling solution was found to be inadequate, other cooling solutions would've been considered.

### Implementation

Forced convection cooling model was implemented with the use of a fan to provide the required airflow. The cells of the accumulator was placed on racks with spacings so as to provide openings through which the air can flow. The airflow was along the lateral surface of the cells thereby absorbing the heat from the cells and being let out at the exhaust. (Fig). The inter-cell gap was used as a design parameter. The idea was to minimize this gap as much as possible while fulfilling the primary objective. As described previously, these gaps contributed significantly to the size of the accumulator.

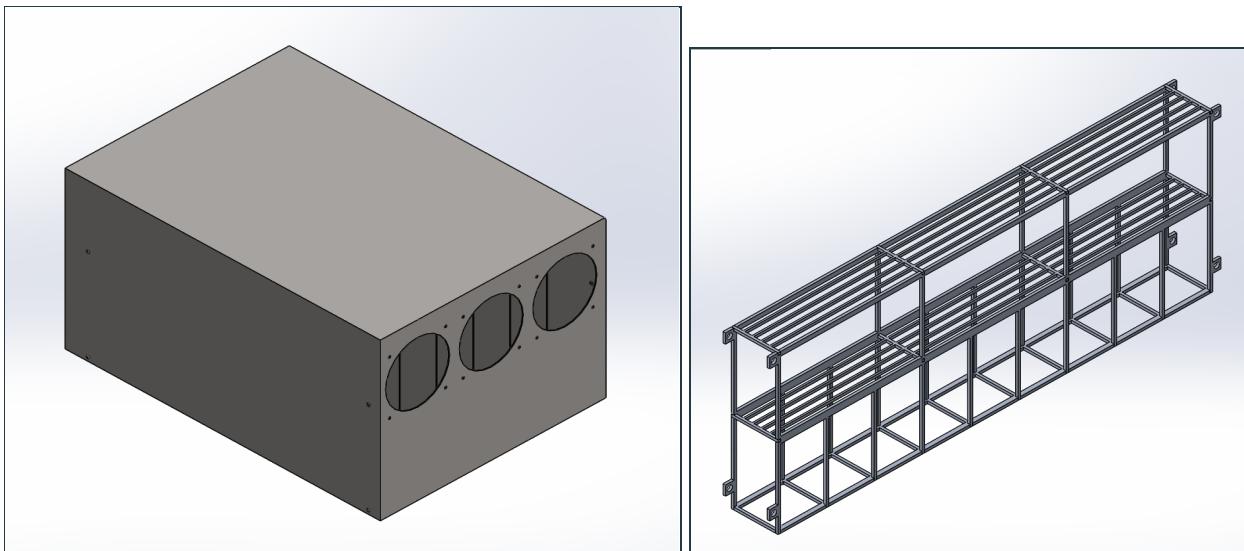
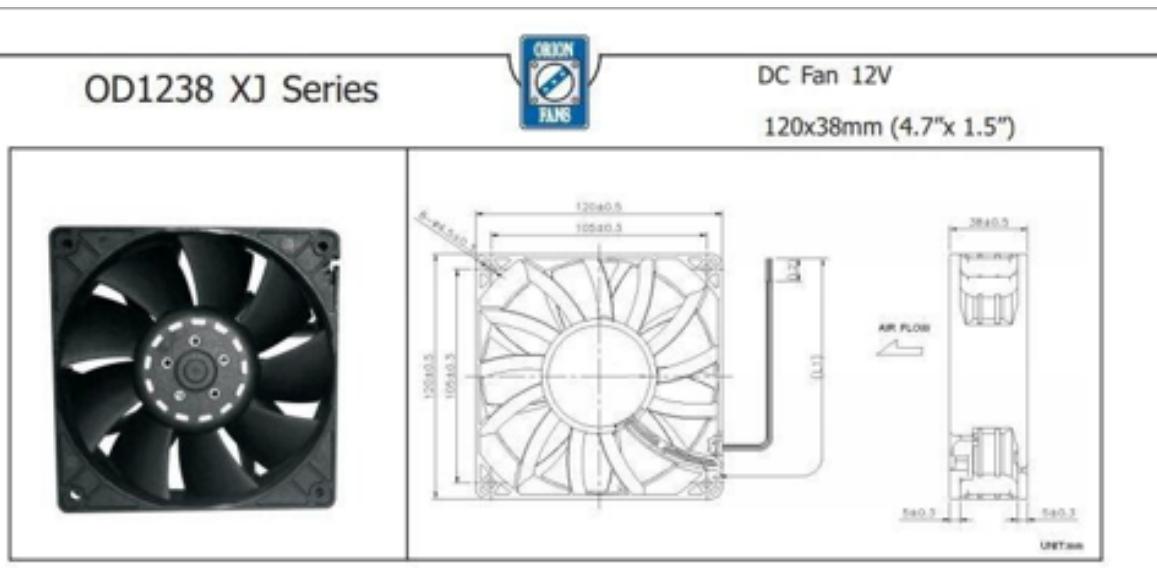


Fig:a)A model of the accumulator container showing the fan inlet.b)The cell racks

#### 4.9.5 Fan Selection:

A prior hand calculation based on how much airflow would be required to remove all of the heat under the assumption the complete airflow being heated up to cell temperature was done. This was an idealistic calculation which gave a lower baseline of at least 27CFM required. This was not possible with natural convection. Hence we required a fan to induce forced convection.

The fan selection was done by an iterative process. First we looked at the fans present in Indian market and filtered them. We then calculated an approximate operating point(based on the fan curve and system resistance curve). The cfm was then compared among the different fans. There was also a consideration of cost and also the space the fan would occupy. We could directly discard fans that cannot provide more than 54 cfm airflow. With experience and various filters we narrowed down on the fan. This was an iterative process based on the simulation results. If the results were not favorable, we went back and looked for a suitable fan. The figure below presents the details of the present fan. This fan was chosen for its very high pressure drop and the CFM that it provides. Three fans were procured and placed in parallel.



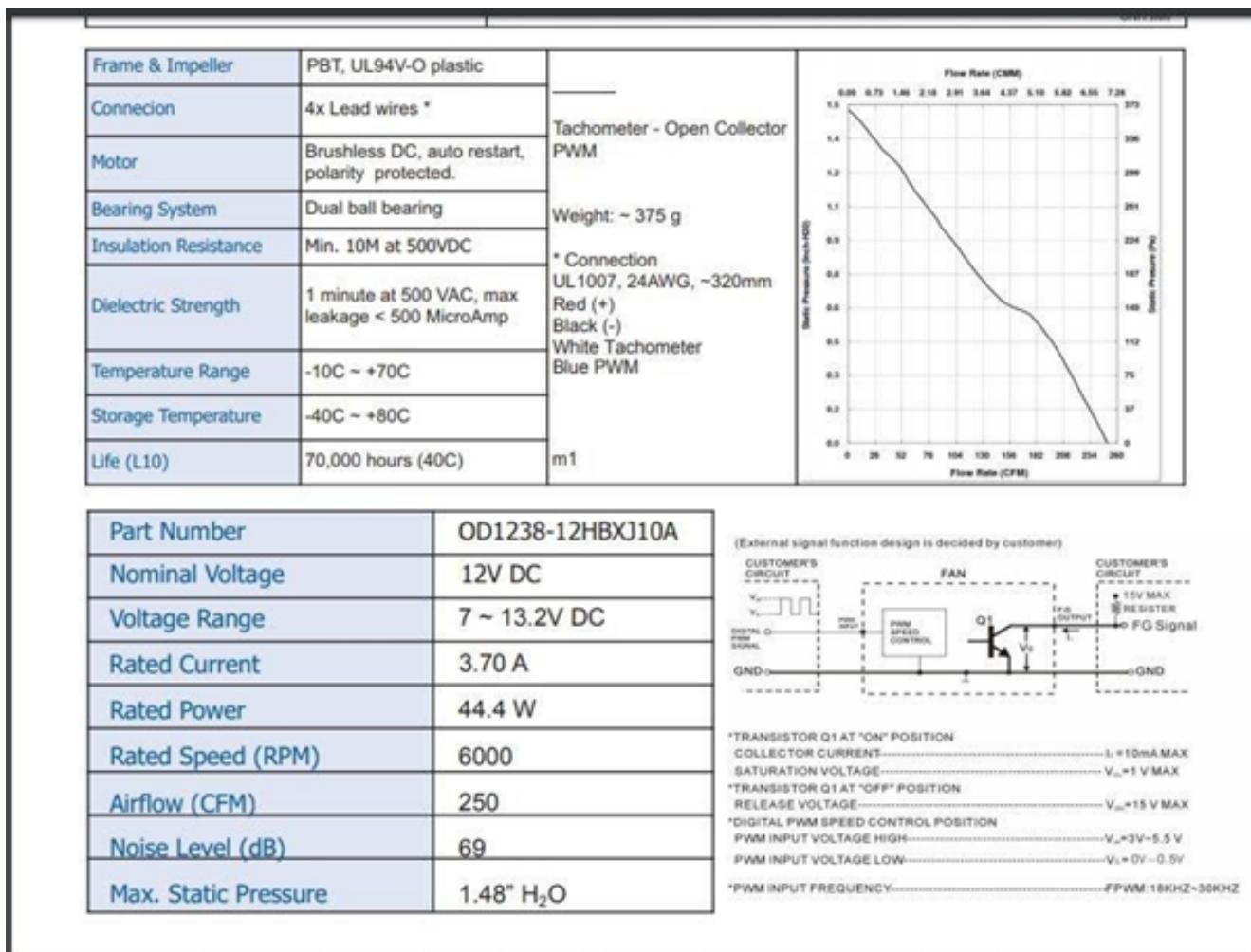


Figure:Details of the selected fan

#### 4.9.6 Analysis and Simulation

The methodology followed was to import CAD model of the battery pack and simplify the model and then simulate it using ANSYS Fluent. ANSYS Icepak maybe a better choice but the software was not available under student license.

The problem was simplified from a transient problem into a steady state problem. This was done to simplify the analysis greatly. So a steady state heat generation of 11.37W(time averaged heat generation over the lap) for each cell was applied. Now this does not model the instantaneously higher heat generation which can be seen at very high current values. We believe that a buffer on the maximum temperature of the cells would be enough to make sure that the temperature doesn't cross the threshold to result

in thermal runaway. A further improvement to the model would be a transient simulation either through Ansys FLUENT only or a lumped parameter analysis model.

The simulation was carried out for 2 modules of the accumulator with a single fan. The gap between cells was a variable design parameter. The simulation was simplified to 2 modules to decrease computing power requirements (ANSYS Fluent Student version has a limit on the number of elements) and reduce the complications of the problem.

#### 4.9.6.1 Pre-Analysis and Calculation

A fan is characterized by its fan curve. The fan curve tells us what volume flow rate the fan can provide at a specific pressure drop. These are inversely proportional and peak pressure drop and peak cfm are at opposite ends. This means that there is always a tradeoff between the pressure drop and the volume flow rate the fan produces. To find out the pressure drop required by the system for any particular rate of airflow, we need to calculate the system resistance curve.

System resistance is the resistance/blockage offered by the system to freeflow. As in our model we have created openings and forced the air to flow through a specific channel, we have created a resistance to flow.

The laminar resistance to flow can be calculated by the Hagen–Poiseuille equation. This resistance is characterized by the length of the pipe and the dimension of the opening. The flow in the opening was predicted to be laminar ( $Re < Re_{cr}(2300)$ ). The friction factor is just the laminar friction factor  $f=64/Re$ .

### Laminar Pressure Drop

$$\Delta P_L = \frac{32\mu L \dot{V}}{A D_h^2}$$

## Head loss

$$\Delta P_o = \frac{\rho \dot{V}^2}{2(C_d A)^2}$$

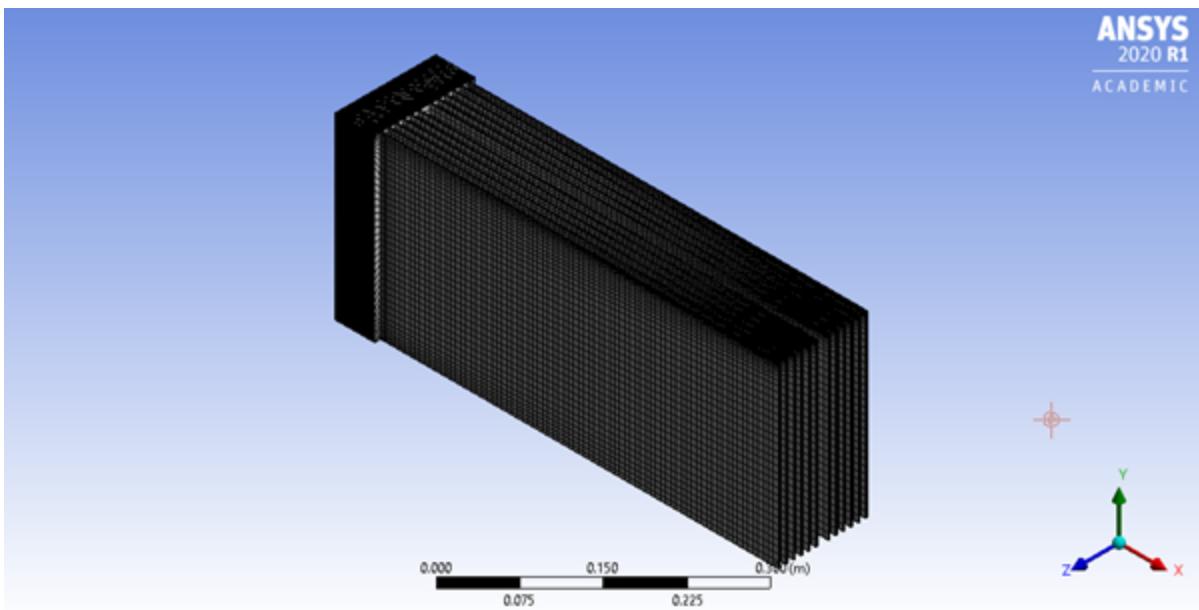
The head loss incurred upon entering the channel is characterized by the following equation.

All the channels of flow were considered parallel to each other. Thus a Resistance as function of flow velocity was formulated.

The operating point of the fan is calculated by super imposing the fan curve and the system resistance and checking the point of intersection. The calculated operating point was()

### 4.9.6.2 Geometry and Meshing in ANSYS Fluent

Only the fluid domain was simulated on FLUENT. This was done to optimize the computing power and STUDENT version restriction to generate the best possible mesh. The figure below shows the resolved mesh:



Various sizing function and inflation functions were used to get a finer resolution of the mesh on the flow walls and flow entry into channels. This was

required to capture the flow physics in near wall regions .This is the primary aim as we are trying to capture the friction factor and heat transfer into the flow,both of which are wall phenomena.Irrespective preliminary calculations suggest the flow inside is laminar,so the mesh need not be very highly resolved.The mesh contains 350320 elements and 299910 nodes.

#### 4.9.6.3 Problem Setup and Solution Procedure:

The following boundary conditions were applied.

- Fan Inlet Boundary condition:This was applied to X-face(ref fig) and was simulated as pressure jump.This is an inbuilt source device in Fluent.The fan curve was input into the solver which also determined the operating point.
- Wall:In terms of fluid flow the no-slip boundary condition was applied.The channel walls had a heat generation of 11.37W/cell(this was converted into the corresponding heat flux form)
- Outlet:The far faces of the channel were modelled as zero-pressure outlets.

The solver used the standard k-epsilon equation along with the energy equation to simulate the flow.This was selected because the k-epsilon model was particularly good at simulating wall phenomenon(although the flow inside the channel may be laminar only).A pseudo-transient coupled solver was used.The initial temperature was assumed to be 30C which is representative of the conditions found on Kari Motor Speedway.

#### 4.9.6.4 Postprocessing and Results:

The post processing was done in ANSYS CFD-post to understand the temperature distribution and the feasibility of the solution. The major results are the operating point of the fan and the subsequent temperature distribution. The operating point of the fan is (215Pa, 0.027m<sup>3</sup>/s). This deviates from both -the hand calculation and even the fan curve. The operating point of the simulation was actually estimated by observing the pressure drop across the container. This discrepancy I believe is caused by the excessive blockage to flow passage from the start. The complete area flow of the fan is not usable. Hence an increased pressure drop in channelling the flow laterally.

The maximum temperature of the complete system is 45.349C which is well under the limit that we discussed above. This is the maximum temperature of the cell walls. At the cell core it will be slightly higher. But this is a localized temperature field. The worst cell average temperature which can be understood by averaging the temperature across the whole cell which comes to be 43.2C

The velocity and CFM in the channel suggest that the flow is indeed laminar, but the flow entry is turbulent in nature. The Nusselt number and friction factor in the channel also conform with laminar correlations.

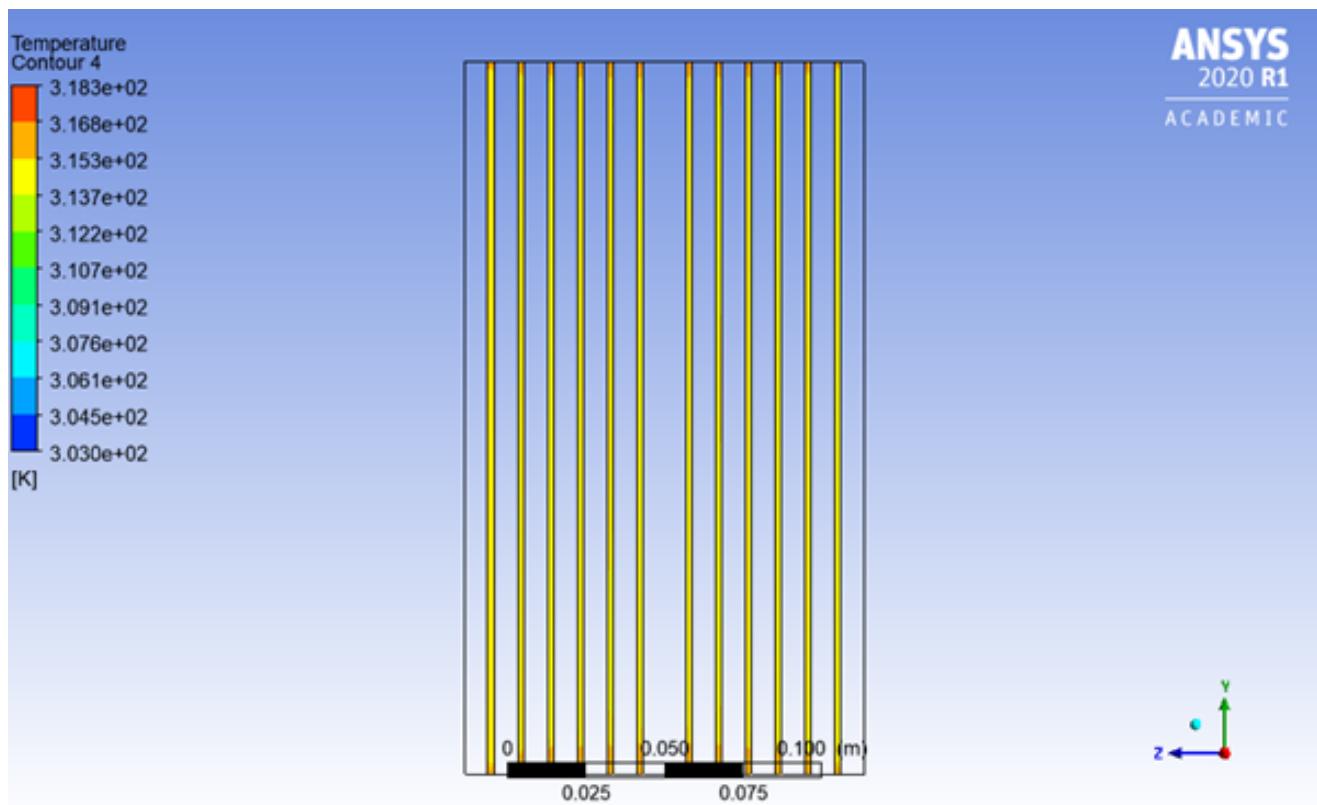


Fig :The temperature distribution at the outlet

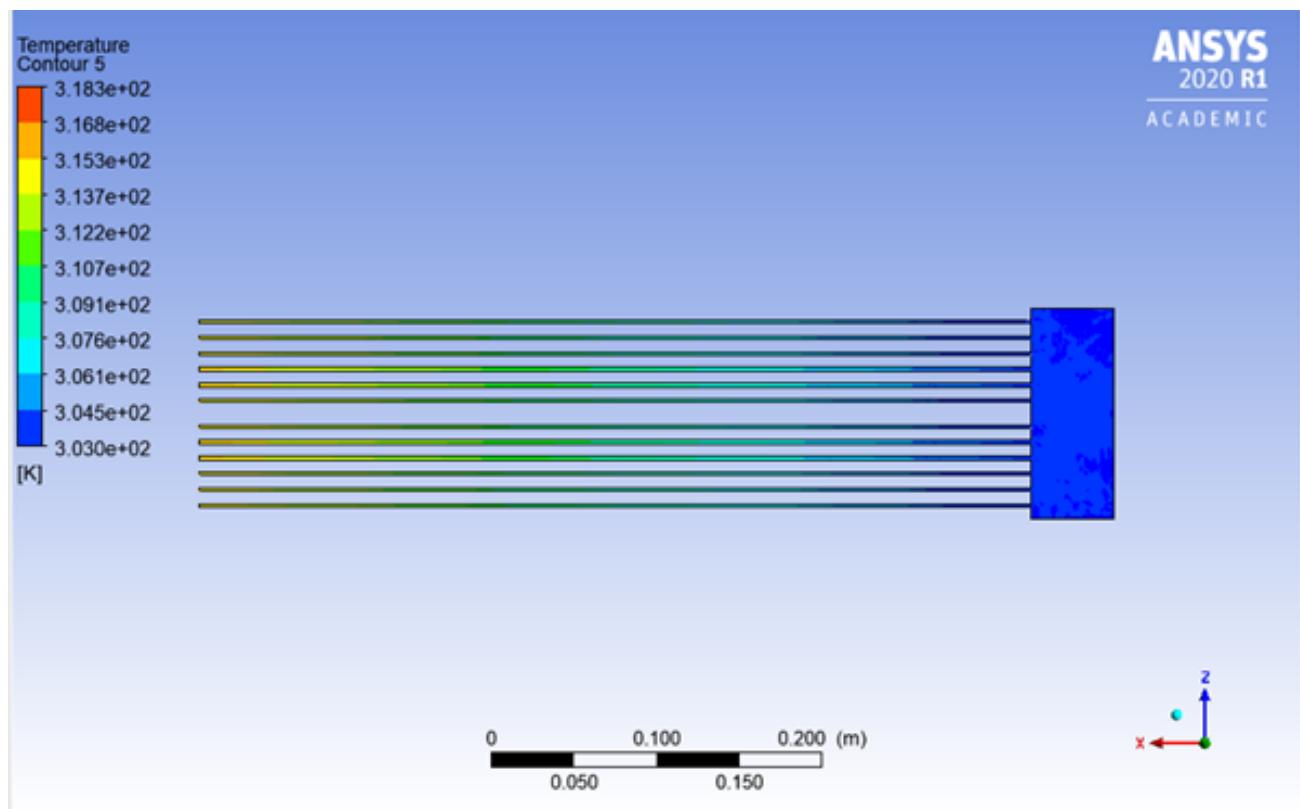


Fig :The temperature distribution of the flow field from the top view

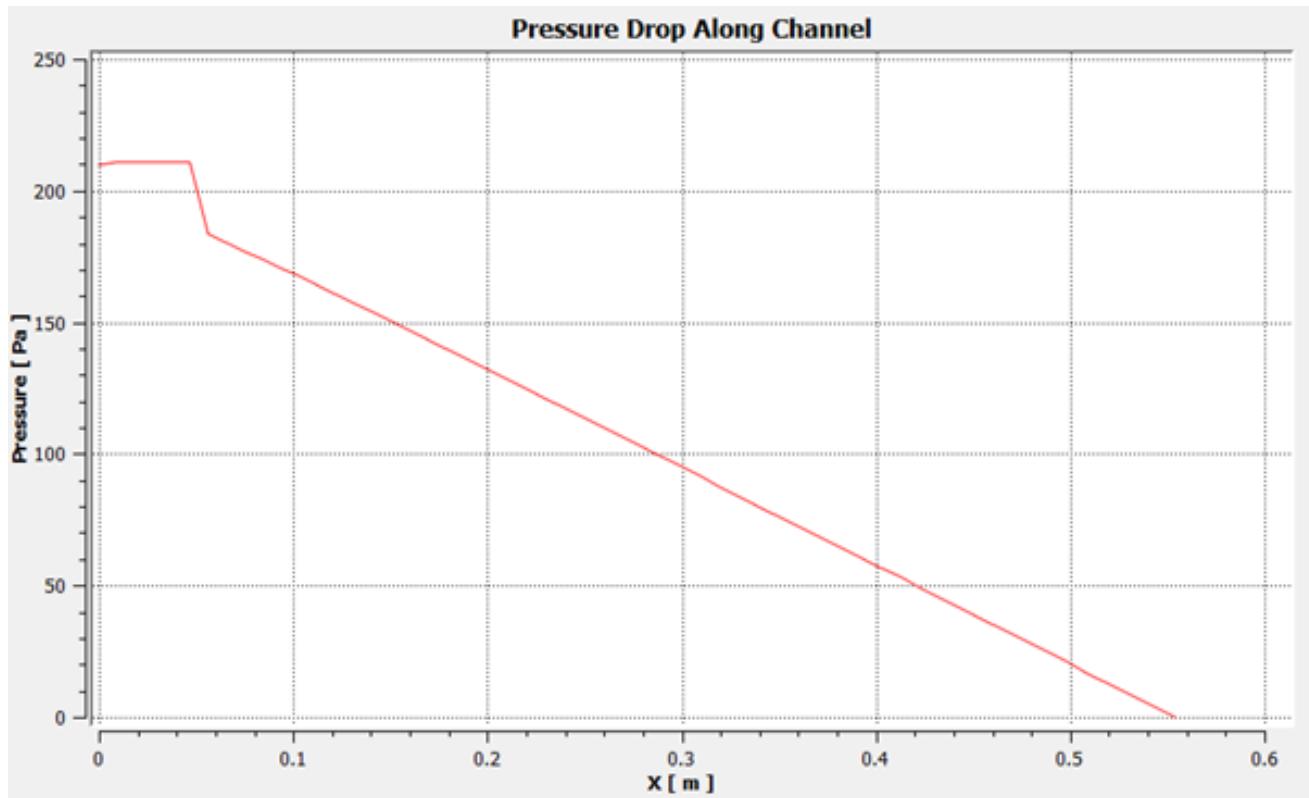


Fig :The pressure drop across a channel. As previously expected there is a head loss and laminar pressure drop loss.

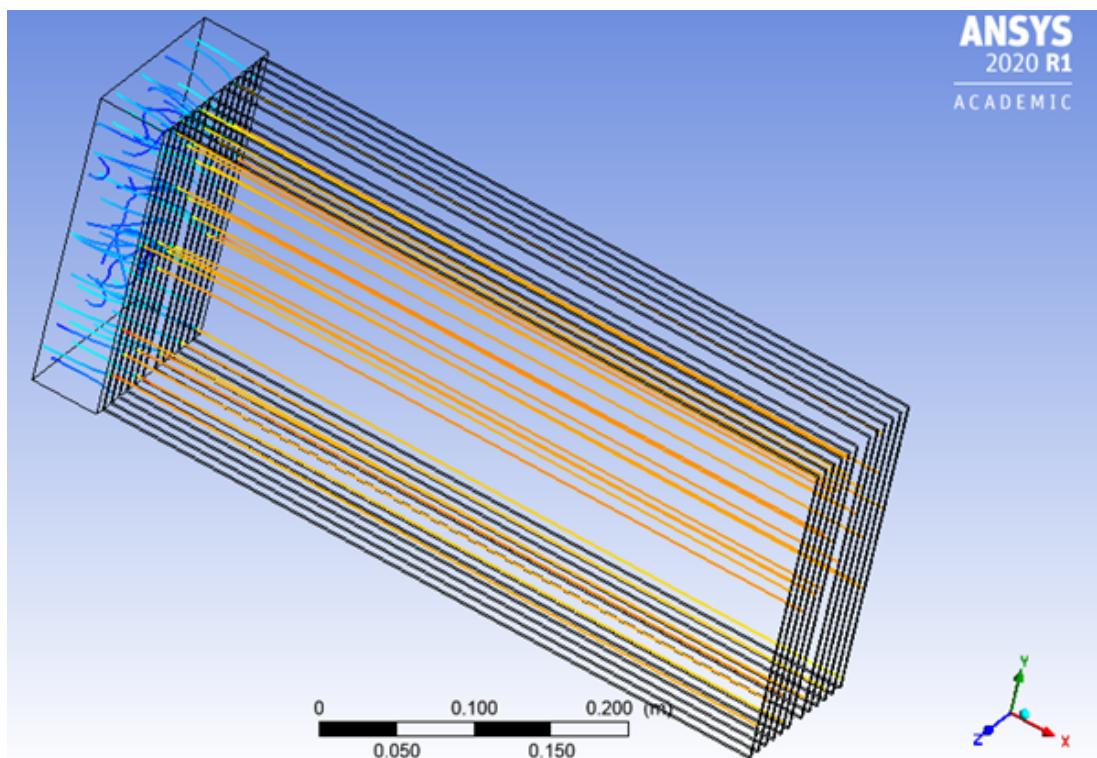


Fig :Visualization of the flow through streamlines.

#### 4.9.6.5 Validation of Simulation:

The primary validation of the simulation comes from the convergence of residuals. The convergence of the residuals shows that the linearization error is minimized and the solution is well converged. The figure below showcases the convergence of the residuals. As is seen all residuals are well below 1e-4 and thus the solution has satisfactorily converged.

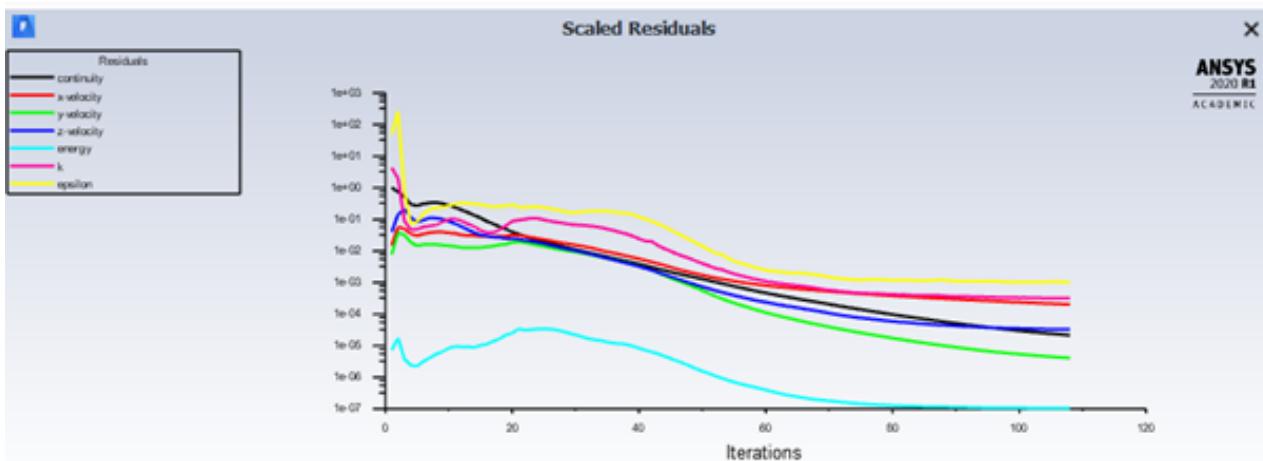


Fig:Residual Convergence for the simulation

The discretization error can be checked through a mesh independence study. A finer mesh with increased number of elements was constructed (1.5x initial) and the model was simulated. The results of this simulation matched very well with coarser mesh. Thus we can be sure that the discretization error is also minimized.

Furthermore the operating point although deviated from the hand calculations followed the expected trend. The Temperature, heat transfer coefficient of the wall and the friction factor at the wall also correlated with well known correlations.

Thus we can be confident that the simulation is a somewhat accurate model of reality if the assumptions taken are true.

## 5. Control and Safety Circuits

### 5.1 Central Network

The central network is divided into 3 further networks:

- 1) Central CAN network connecting the DCM, Motor controller, BMS and the DAQ
- 2) DAQ network that involves other mechanical sensors like IMU. This network also connects the DAQ to the dashboard to display various signals ( mainly Speed and SOC ).
- 3) BMS to Motor controller network for varying power transmission and limiting the current limits based on real time calculations.

#### 5.1.1. Drive Control Module

For the purpose of Drive Control Module, STM32F42Z Nucleo MCU is used. The board has 2 in-built CAN 2.0 B controllers for connecting it to our CAN network.

The module's main I/O pins include:

Inputs :

ADC : APPS

GPIO Pins : RTD button

Status signals : Brake Pedal signal, Shutdown signal

Outputs : CAN network ( in output/transmit-only mode) , RTD buzzer, Standby LED.

The module's main functions include the following:

- 1) Torque Encoder - 2 X12 Bit ADC converter of the module will receive the APPS signals in two different channels and the customized coded logic will turn this into an appropriate signal output which is then provided to the Motor controller/driver. Given that there is no implausibility, the logic will perform a weighted average of the APPS value, which will then be scaled as per the Motor controller's requirement. The signal is sent to the main CAN network and will also be provided to the DAQ via the same channel.

The Brake status signal will disable the Torque encoder(software logic) while the brakes are applied. This will further improve the entire breaking process. This will be achieved using an software interrupt for the braking process.

The Software will also perform error check in the following ways :

- Short circuit to ground and short circuit to supply, comparing the corresponding digital values from the APPS to hardcoded values that indicate ground and supply respectively, and values that correspond to impossible angles of the pedal.
- Checking the 10% implausibility, the values are sampled every 1ms, sampling is hardware triggered from a timer on the microcontroller.

Average of 5 consecutive values are taken for both the analog channels respectively corresponding to the two APPS, and for these averaged values the 10% error is checked.

Iterative process of 20 times occurs to check for implausibility, if at the end of the iteration , there is a majority that has positive indication to the 10% implausibility then the DCM comes back to stand-by mode. The driver would have to bring the car back to RTD by dedicated action.

Furthermore, an unique "APPS error" message will be sent over the CAN network ( to the motor controller and the DAQ ).

- 2) RTD ( Ready to Drive) - The car, by default would be in stand-by mode and the dashboard would contain a LED to indicate this status (the software equivalent of this logic would be on the DCM ). A dedicated RTD button on the dashboard would be directly connected to the DCM via its GPIO pin. The Brake pedal signal would be checked by the DCM before changing to RTD mode. Also, the RTD buzzer would be activated at the same time. The CAN network would also be notified about this.

The car will turn back to stand-by mode in case of following:

- a) APPS error has occurred ( DCM has customized logic for this ).
- b) Shutdown is initiated (Shutdown signal ).

- 3) **Post - Shutdown sequence** - After the hardware shutdown has been triggered, the circuit would be sending the shutdown init status to the DCM. This would be followed by an unique "Shutdown init" CAN message to both BMS and DAQ. This message would kill the power supply process from the battery by signaling the BMS to turn the Voltage low ( Hardware Shutdown system opens the circuit but this additional feature would ensure that no voltage is applied to the open circuit as well, thus removing chances of charge - arching ).

This series of events would also turn the car to stand-by mode automatically.

### 5.1.2. Data Acquisition System

The DAQ is made using commercially available Kvaser Memorator Pro 2xHS V2 USB to CAN Data logger. It stores the main CAN network's data and also stores the data from other mechanical sensors. The dual CAN input bus helps it intake the 2 different sets of data simultaneously. The features are explained in detail :

- 1) **Dashboard connectivity** - The DAQ will be connected to an additional controller that decrypts the CAN messages and sends some chosen signals to the dashboard display. This includes APPS signals averaged with the Speed calculated by the Motor controller, and the SOC and SOH from the BMS. This will additionally contain a wireless module to send the real time data to Laptop( in the pit ) for further analysis. The Kvaser product provides the functionality to log and stream data at the same time.
- 2) **Second CAN channel** - This will include the data coming from various other sensors used for mechanical data analysis. This would include IMU and GPS for road data ( and would be sent sequentially ). It will also include Motor speed and would be directly fed to the dashboard connected controller.
- 3) **DCM message priority shift** - Any priority based controller has a problem of network starvation. To avoid this, the highest priority node will have an additional logic to shift its priority based on certain features. This would be triggered in case of : constant signal(high signal) is maintained ( 5% change allowed ) for more than 3s, or the low priority signal is at critical condition. Other than these, the low priority message would be triggered after constant intervals as well.
- 4) **Additional Application Layer Protocol** - This would be necessary because the starved message should not lose any information. In case of starvation of following:
  - a) BMS - The SOH and SOC value would be passed in the same message. Also, the average of all the values of SOH/SOC would be sent ( starting from the last sent message )
  - b) DCM - The first bit of the message would be the control bit. Thereafter, if the control bit indicates the starved message, the initial half dataframe would be the time-average starved message.

The commercial DAQ has additional features that involve message filtering based on a trigger. These logics are written as the Kvaser - t-scripts. Additionally Kvaser's CanKing and CANalyzer simulations and LabView simulations can be done with help of proper API based Frameworks.

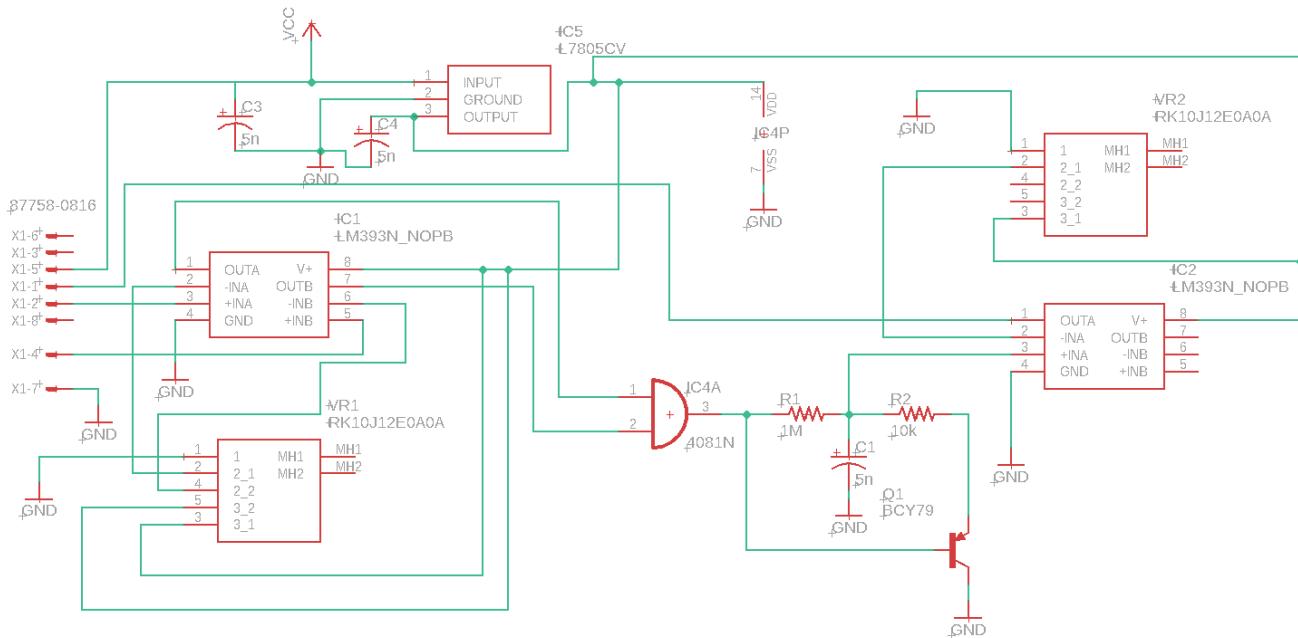
## 5.2. Safety Circuits and Sensors

### 5.2.1 Brake System plausibility device

## **Brake System plausibility device**

The Brake System Plausibility Device (BSPD) essentially ensures that the acceleration pedal and brake pedal aren't pressed simultaneously for more than 500ms. Signals from the accumulator current sensor and one of the brake pressure transducers are taken and compared against a threshold value. The threshold voltage of the main accumulator current sensor corresponds to 5kW power being delivered to the motors.

Schematic:-



An AND gate will check if both the inputs are above the threshold voltages after the signals pass through a comparator. In case both the signals are above the threshold values, the circuit checks for persistence of this signal by charging a capacitor. After 0.5s, the RC circuit will repach a voltage above the threshold voltage set by the potentiometer. The output signal is sent to a BPS Latch.

### 5.2.2. Shutdown Circuit

## Shutdown Circuit

**Schematic:-**

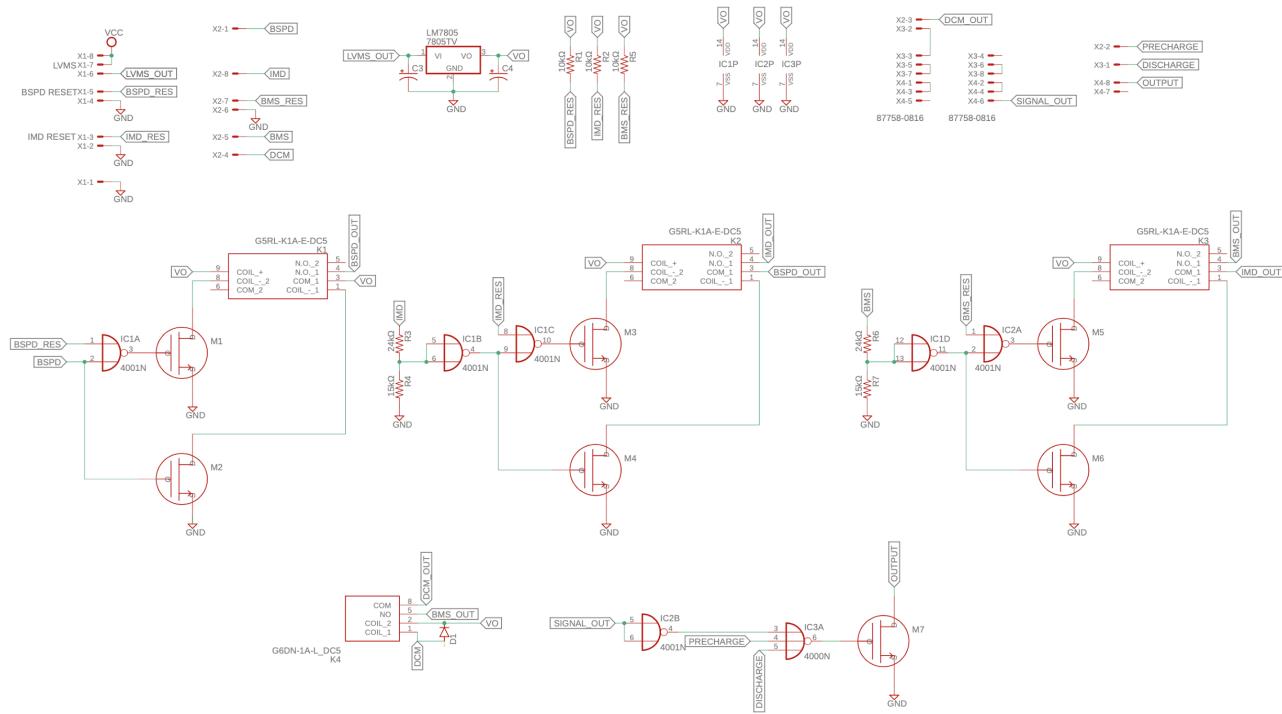


Fig. Schematic for shutdown circuit

### 5.2.3. Pre-Charge - Discharge

#### Precharge Circuit

The precharge circuit is a protection circuitry for the motor controller.

Consider a case where there is no precharging mechanism present:

The motor controller has an intermediate capacitance that is charged up before the inverter (controller) starts functioning. The low series resistance as well as no back EMF of the Tractive System would lead to sudden inrush of a large value of current into the motor controller capacitor causing the tractive system fuse to blow. This would happen every time, and even after replacing the fuse , we won't be able to start the car as the fuse may blow again.

Considering the natural response of the capacitor:

$$i_C = \frac{V_S (1 - e^{\frac{-t}{RC}})}{R}$$

Where:

I<sub>c</sub> - Capacitor current

V<sub>s</sub> - Accumulator Voltage

R - Circuit Resistance

C - Intermediate Capacitance

Since the value of the resistance is very low, the current flowing into the capacitor would be very high, blowing the Tractive System fuse.

Thus, to ensure that the fuse does not blow out at regular intervals we intend to use a resistor which increases the series resistance so that the intermediate capacitance of the motor controller can be charged to a safety limit as specified by the Formula Bharat Rulebook.

The circuitry used is called a **pre-charge circuit..**

Calculations:

Rule book states that the precharge circuit should charge the immediate capacitance by at least 95% before closing of AIR.

We know that the capacitance of the circuit is 320uF and the resistance chosen is equal to 1k ohm.

Therefore,

Time Constant(T) = 0.32s

Tractive System voltage = 320V

95% of Tractive System voltage = 304V

Time required to charge up to 95% can be calculated by:-

$$304 = 320(1 - e^{(-t/T)})$$

$$t = 0.958s$$

Therefore we have chosen t = 1 sec.

$$\begin{aligned} \text{Heat generated by resistor during this period} &= \frac{1}{2} CV^2 \\ &= 16.384J \end{aligned}$$

$$\begin{aligned} \text{Rating of resistance chosen} &= 35W * 1s \\ &= 35J \end{aligned}$$

which is greater than heat generated in the charging process.

For timer :-

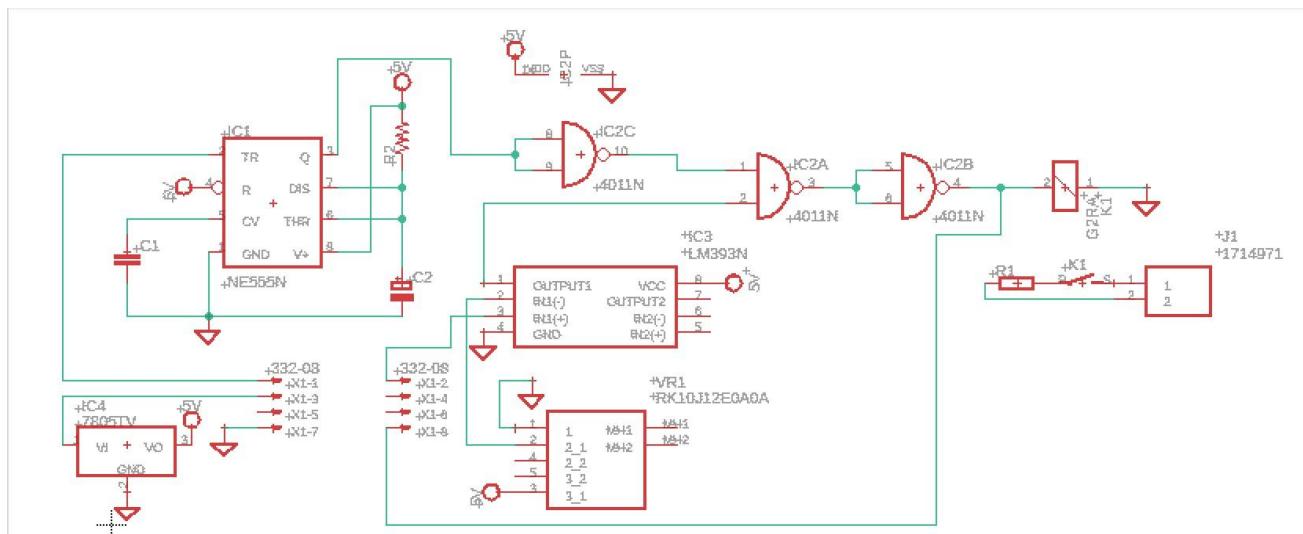
$$t = (1.1)RC$$

$$t = 1 \text{ sec}$$

Therefore R = 1kOhm (**Pre-Charge Resistor**)

$$C = 320\mu\text{F}$$

### Schematic:



## Discharge Circuit

When both the AIRs are closed and the Tractive System is active, intermediate capacitance of the motor controller stores energy. When the Drive Control Module shifts to the StandBy mode, the discharge circuit discharges the energy stored in the capacitor.

Discharging the intermediate capacitance protects the engineer working on the car when it has been switched to standby mode and the TSAL is lit green.

It involves the usage of a resistor connected in series with the capacitor. The discharge relay receives a control signal to switch to the ON position, allowing intermediate capacitor's stored energy to be removed as heat.

$$i_C = \frac{V_S \left( e^{\frac{-t}{RC}} \right)}{R}$$

## Natural Response of Discharge Circuit

### **Calculation:**

We know that the capacitance of the circuit is  $320\mu\text{F}$ . Resistance chosen is equal to  $1\text{k ohm}$ .

Therefore,

Time Constant( $T$ ) = 0.32s

Tractive System voltage = 320V

Time required to discharge upto 95% can be calculated by:

$$16 = 320(e^{-t/T})$$

$t = 0.958s$

Therefore we have chosen  $t = 1$  sec.

$$\text{Heat generated by resistor during this period} = \frac{1}{2} CV^2 \\ = 16.384 \text{ J}$$

$$\text{Rating of resistance chosen} = 35 \text{ W} * 1 \text{ s} \\ = 35 \text{ J}$$

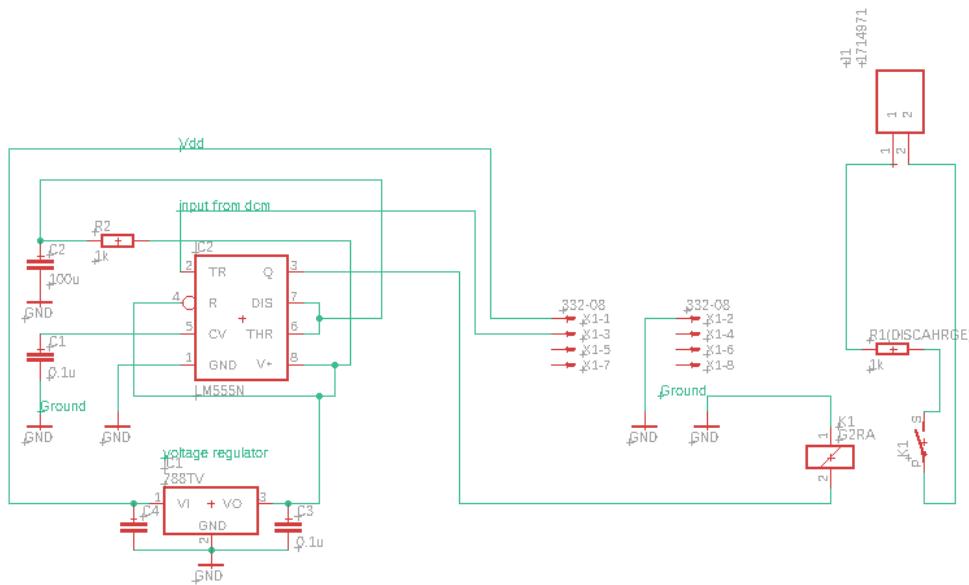
which is greater than heat generated in the discharging process.

For timer :-  $t = (1.1)RC$

$$t = 1 \text{ sec}$$

Therefore  $R = 1\text{k}\Omega$  (**Discharge Resistor**)

### Schematic:



## 5.2.4. TSAL - AIL

### Tractive System Active Light

The tractive system is active when any of the following conditions are true:-

1. An Accumulator Isolation Relay is closed.
2. The pre-charge relay is closed.
3. The voltage outside the accumulator container exceeds 60 V DC or 25 V AC RMS. This implies that at least the voltage of all DC-link capacitors needs to be measured even with the HVD removed.

Active means TSAL must blink continuously in **red** color.

Otherwise, TSAL must remain in **green** color.

Schematic :-

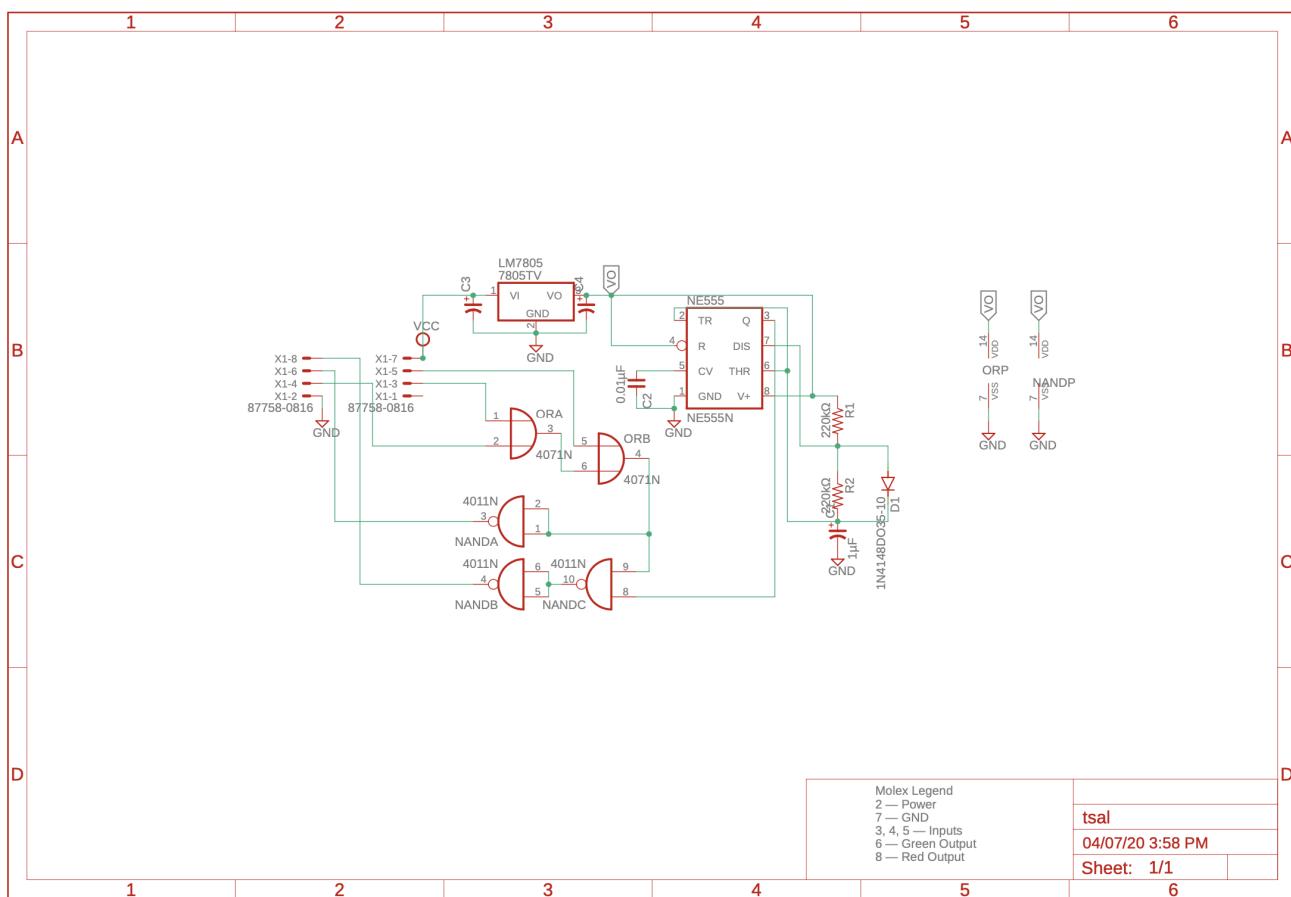


Fig. Schematic for TSAL

## Accumulator Indicator Light (AIL)

We need an indicator to show whether voltage greater than 60V (DC) is present behind AIR or not so that the person about to work on the vehicle is aware of the high voltage.

Schematics:-

Voltage Sensor:

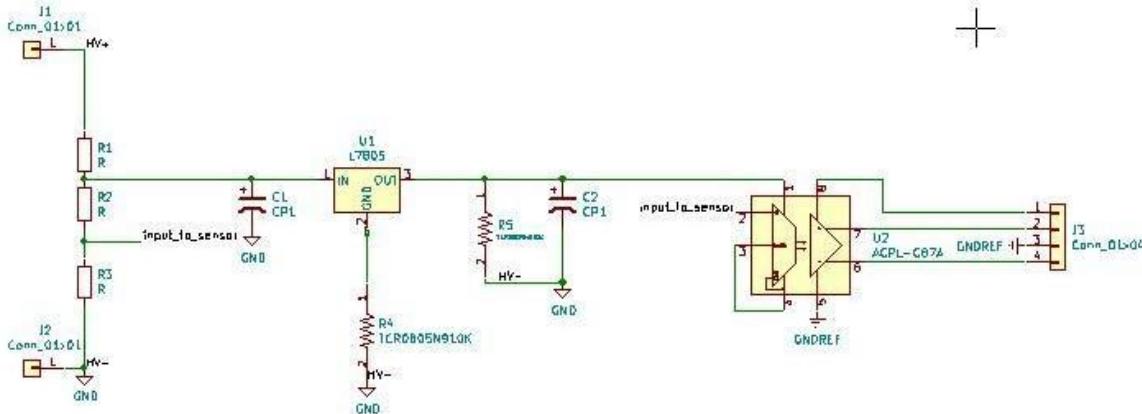


Fig. Schematic for Voltage Sensor

Control:-

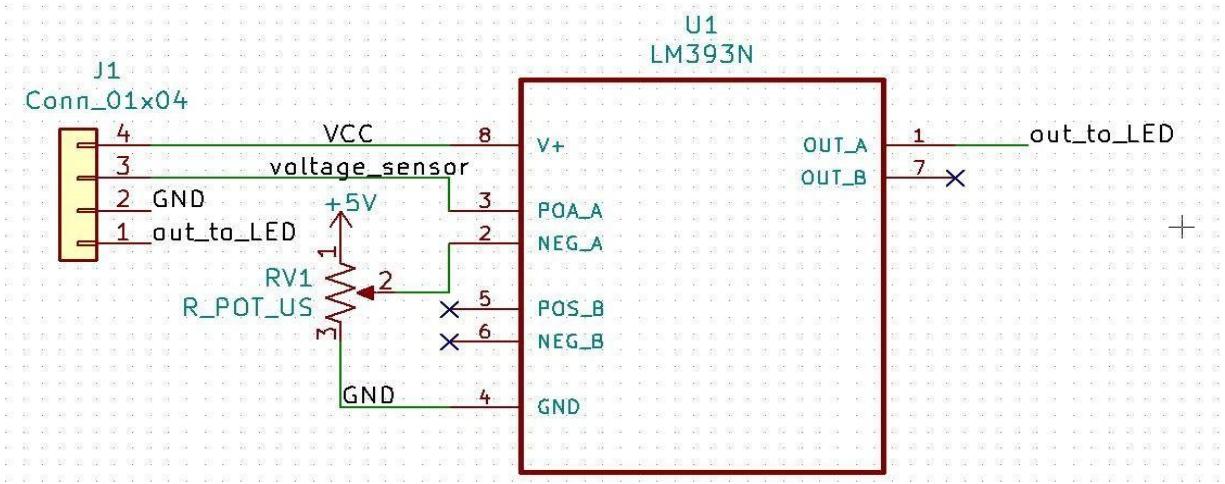


Fig. Schematic for AIL

PCB Location-

The AIL is present inside the battery container.