MATLAB-Simulink-Simscape Model Documentation

Version 1.0, Last Updated 5/4/2023

Steps to run the simulation:

1. Uncomment 'Autonomie data' block (for discharge), or 'Charging Algorithm' block (for charging), or keep both commented (for precooling)
2. Uncomment either the 'Refrigeration Loop' block if using active cooling or the 'radiator' block if using passive cooling.
3. In the parameter_initialisation.m set the ambient temperature by changing 'ambient_temp' variable
4. Set the initial temperature of the battery (T_init_bat) to ambient temperature if battery is not precooled and if precooled set to the precooled temperature.
5. Run the parameter_initialisation.m
6. Run The Simulink model

Five different possible cases to run in this model are:

1. Precooling with active cooling (refrigeration loop): uncomment 'Refrigeration Loop' block
2. Charging with active cooling: uncomment 'Refrigeration Loop' block and 'Charging Algorithm' block
3. Discharge with active cooling: uncomment 'Refrigeration Loop' block and 'Autonomie data' block
4. Charging with passive cooling (radiator): Uncomment 'radiator' block and 'Charging Algorithm' block
5. Discharge with passive cooling: Uncomment 'radiator' block and 'Autonomie data' block

1. Autonomie data:

When discharging the battery, US06 drive cycle data (speed and power) is fed to the model by data processed from Autonime software. There are many other signals including heat generation from different blocks that are not used in this model. We only use the power signal to calculate the current being drawn from the battery and speed signal to use for air speed in 'condenser' and 'radiator' blocks. The data file containing the data and the drive cycle should be selected inside this block as shown in figure 2.

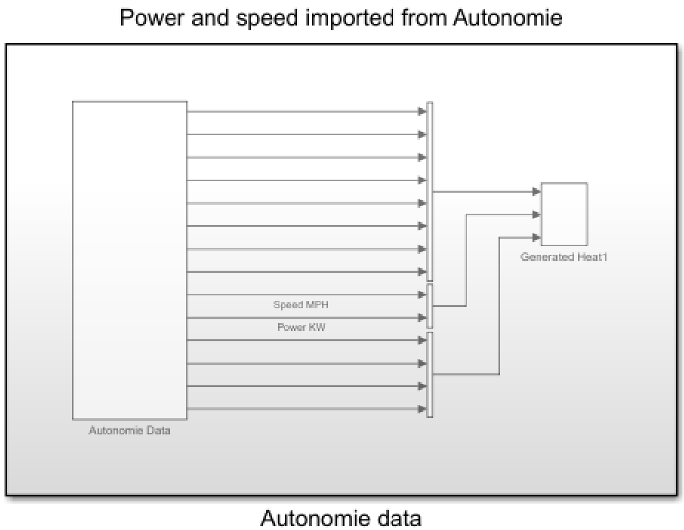

Figure 1

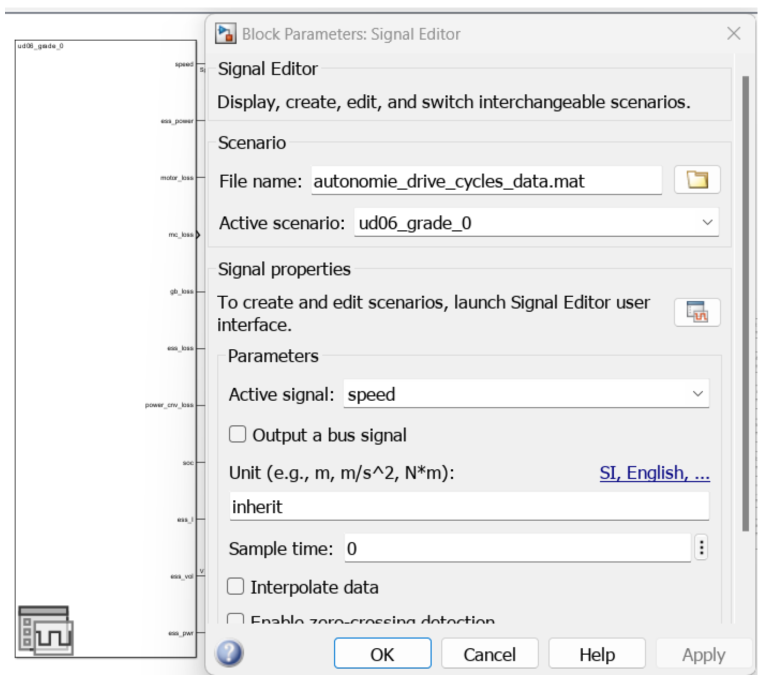

Figure 2

2. Ambient Conditions

The ‘Ambient Conditions’ block provides the weather conditions signals for temperature, pressure, relative humidity, and CO_2 fraction.

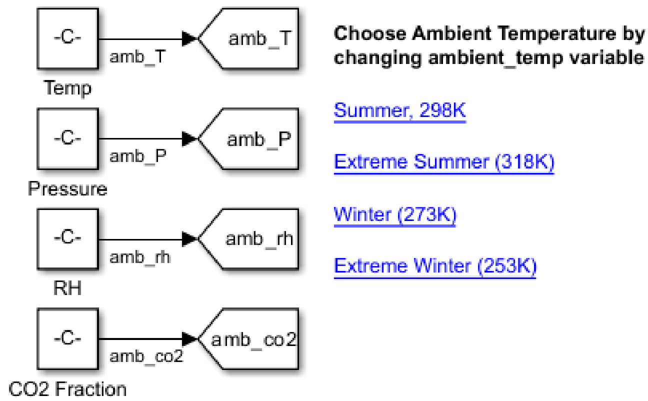

Figure 3

3. Charging algorithm

This block calculates the current for constant current-constant voltage (CC-CV) charging protocol that will be fed into the battery block. This block will use the c-rate that we set in the parameter_initialisation.m file.

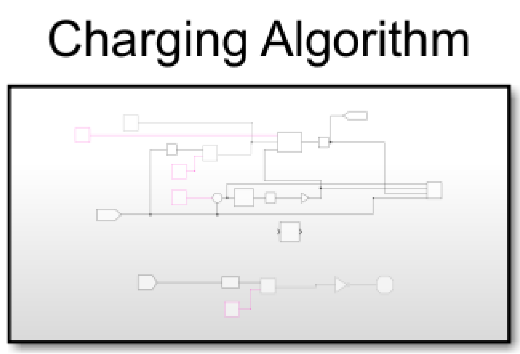

Figure 4

4. Refrigeration Loop :

The refrigeration loop is composed of a Chiller, PID controlled compressor, Condenser, and TXV or expansion valve. The Chiller receives the coolant liquid (here water-EG mixture) coming out of the battery and cools it down by absoring the heat to the refrigerant line(here R134-a), so basically the chiller is the evaporator for refrigeration cycle. Next, a compressor block with PID controlled mass flow rate receives the superheated refrigerant from the chiller and increases the pressure to the condenser pressure. Next, the high-pressure high-temperature refrigerant enters an air-cooled condenser and condenses into liquid. Finally, the expansion valve whose main function in the cycle is to ensure the compressor inlet is superheat reduces the pressure of the refrigerant in an iso-enthalpic process.

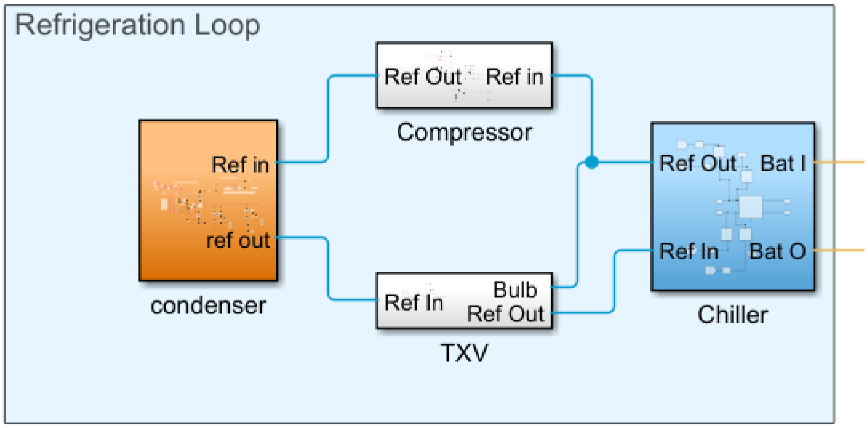

Figure 5

5. PH diagram

Uncomment this block when using the refigeration block to see the PH diagram of the refrigeration cycle..

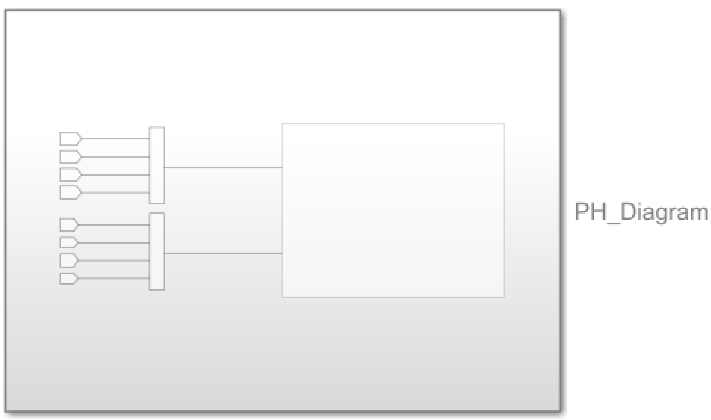

Figure 6

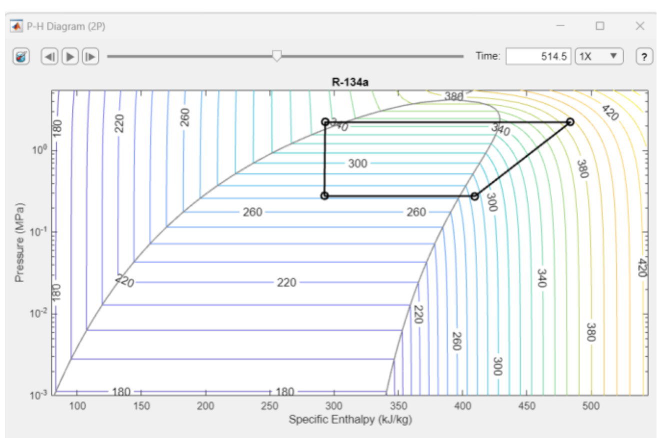

Figure 7

6. Radiator:

The radiator receives the coolant from battery and cools it down by rejecting the heat to the atmosphere.

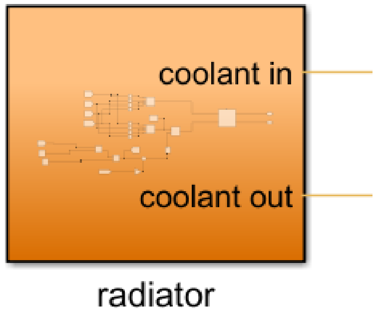

Figure 8

7. Battery and cold plate

The battery has four packs whose temperature (T1 to T4) can be seen in figure 9. The current coming from either the 'Autonomie data' block or 'Charging algorithm' block is fed into the equivalent-circuit model of the battery shown in figure 10. The heat generation in each pack is then connected to a thermal mass that is connected to cold plates under the battery which exchanges heat with the battery by running a coolant (water-EG) through pipes inside the cold plate. Both the convection and conduction parts of heat transfer is modeled in these blocks.

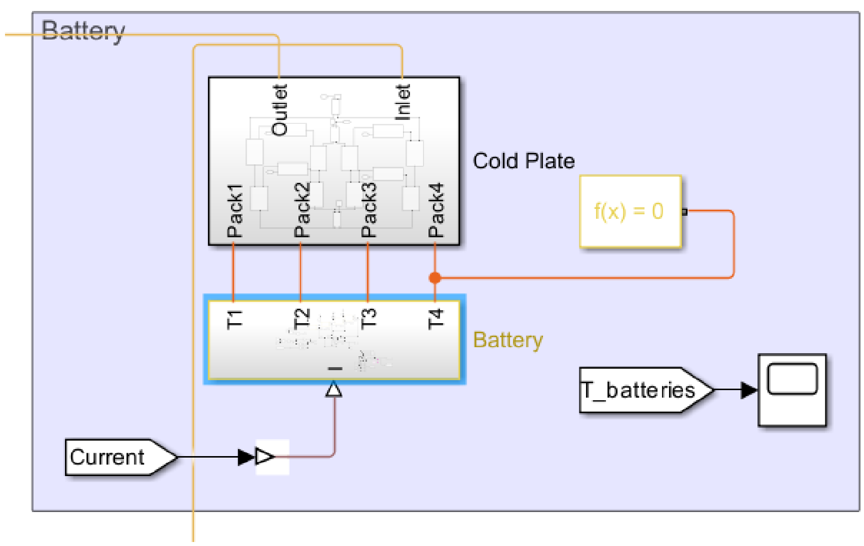

Figure 9

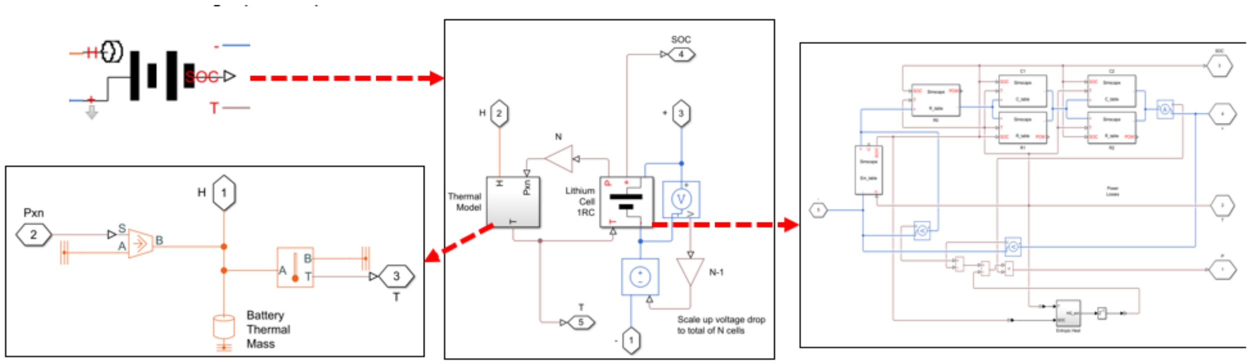

Figure 10

8. Aging Function

The MATLAB file aging_fun.m contains the combined aging model that uses the principle of cumulitive damage to determine the number of cycles to end-of-life which is basically when cell capacity reaches 80 percent of its original capacity.

Once running the simulation save the current and temperature profiles by running either charging_profiles.m or discharge_profiles.m, then load both the charge and discharge profiles in MATLAB workspace and run the aging funtion.