MECH545 - Hybrid Electric Vehicle Propulsion
Project 05 – Simulink Introduction – Tractive Power
Dr. Craig Hoff

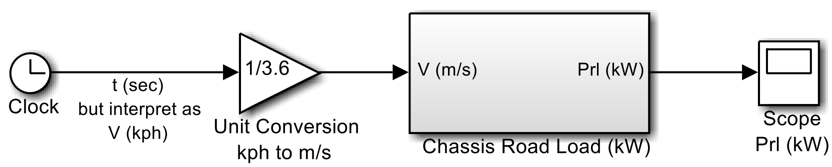
HEV Objective: To determine tractive power requirements for a vehicle.

MATLAB/Simulink Objective: To learn the fundamentals of Simulink programming.

Reminder: Prior to starting this project, take a Simulink tutorial (see separate document)

All Students - Problem Statement:

<u>Part 1 Build a Simulink model to determine the road load on a vehicle</u>
Create a Simulink model that will calculate the **<u>road load power</u>** requirement for the **<u>original</u> vehicle used for Project 1**[1]. Verify that the Simulink model yields the same results as your MATLAB model. The form of your solution should be:



Notes:
- As shown in the above figure, use the clock output (time) to represent the velocity in kph (or alternatively, use a ramp function to generate velocity values)
- Use a fixed-time step integrator with a small enough time step to produce smooth results

<u>Part 2 Build a Simulink model to evaluate the power demands for a given drive cycle</u>
Create a Simulink model will calculate the **<u>tractive power</u>** requirement for the **<u>original vehicle used in Project 1</u>**. Use your model to determine the maximum engine power and maximum braking power for the vehicle is driven on the following drive schedules:

(a) US EPA 'City' cycle
(b) US EPA 'Highway' cycle
(c) US EPA 'Aggressive' cycle

You will use a **1-D lookup table** to input the driving schedule (see below for more details).

[1] i.e. the vehicle weight used for this assignment is the original vehicle weight (not the weight including the EREV conversion.) We're using the original mass for the next two assignments, so that we can compare our results to known test data (in Project 6).

Recall from lecture that the tractive effort can be found from:

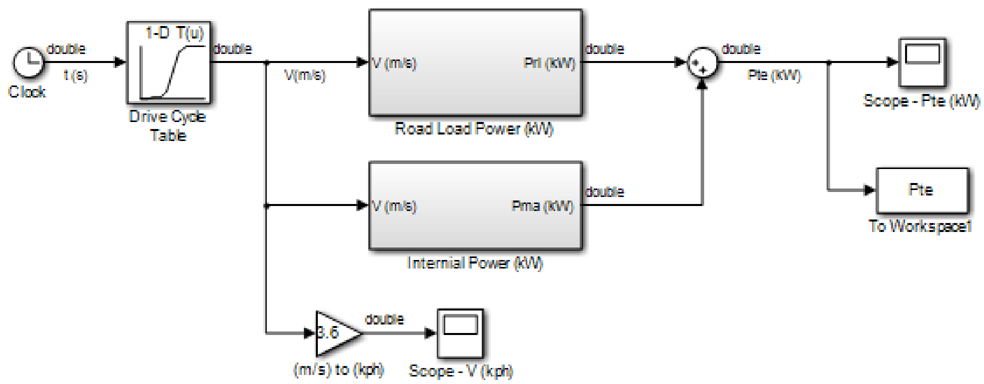
$$P_{te} = P_{rl} + m_e aV$$

And that the effective mass for the vehicle (m_e) represents the combined linear inertia of the vehicle and the rotational inertia of the driveline components. The effective mass can be found from:

$$m_e = k_m \cdot m$$

Assume that the mass factor (k_m) for this vehicle is 1.15.

Your model should have the following form:



Submit:

- For Part 1 – Plot of *road load power*, the model file, and verify the model is correct
- For Part 2 – Plots of *tractive power* for each drive cycle, a summary table of peak positive (engine) and peak negative (braking) values, discussion of the results, your model.
- MECH-691 Students: A plot of power components for the FUDDS driving cycle. What is the largest value for each component?

Due:

- One Week

Please Note: See below for (a) how to load the test drive cycle data, (b) how to setup lookup tables in Simulink, and (c) how to pass results from Simulink to MATLAB

MECH691 Students – Additional Problem Statement:

<u>Part 3 – Contributions of Components</u> Save your model from Part 2 for later use. Create a modified version of your model from Part 2 which will allow you plot each component of power [aerodynamic drag (P_{ad}), rolling resistance (P_{rr}), and inertia (P_{ma})] separately. Provide a summary table for your results which shows peak values for the components and discuss your results. Note the hill climbing force is zero on a level grade so it can be ignored. Hint: You might want to consider using the **Mux Block** to collect your signals.

MATLAB/Simulink Issues

MATAB Workspace files: The data for the driving schedules can be downloaded from BlackBoard as a .zip file. There are four files in the folder:

- sch_fuds1.mat , which contain data for the US City Test
- sch_fhds1.mat, which contain data for the US Highway Test
- sch_us06.mat, which contain data for the US 06 Test

The data for each driving schedule is stored in a **MATLAB workspace (*.mat) file**.

Data from workspace files can be loaded into memory by either of two ways:

- Double-click on the *.mat file in the "Current Directory" window
- Use the MATLAB command: load filename.mat. For example to load the city cycle type: load sch_fuds1.mat

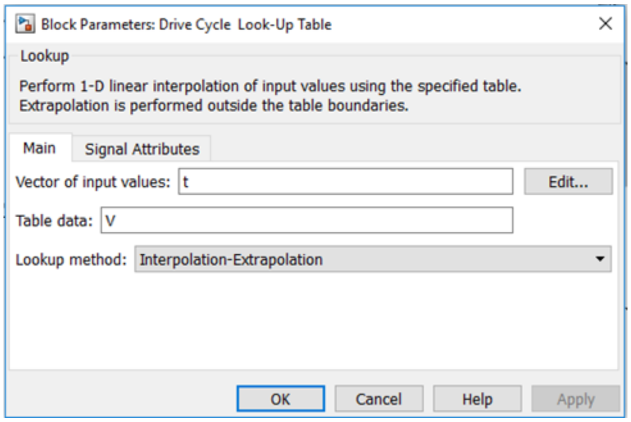
Each *.mat file contains several workspace variables to define the drive schedule. The variable *sch_cycle* is a n x 2 matrix containing the drive cycle data. The first column in the matrix is the time data (in seconds) and the second column in the matrix is **the vehicle speed (in m/s)**.

Extracting columns from a larger matrix: To extract the two columns into independent vectors, use the following commands:

> **t = sch_cycle (:,1);**
> **V = sch_cycle (:,2);**

Using lookup tables: To use the data in the Lookup Table, double-click on the Lookup Table icon to bring up the editing window.

Table 1 MATLAB 2016a Dialog box for 1-D Lookup Table



Settings:

- Vector of input signals: the lookup indices, i.e the time values
- Table data: the lookup values, i.e. the velocity value
- Lookup Method: Use Interpolation-Extrapolation

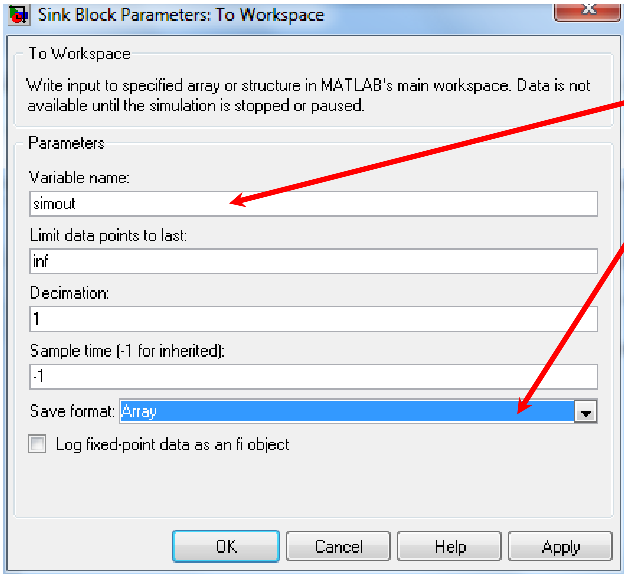
When switching drive schedules: Each time you load a new data set, it is a good idea to delete (clear) the previous data set first; otherwise unexpected things can happen. For this example,

- To clear all the variables from memory, type: clear
- To clear all the variables starting with the letters sch, type: clear sch*

Creating Output Plots: There is no ‘Copy’ command in the menu options of the Scope output window. Scopes are only used look at signal values. To create a proper plot of your signal, you must send the data back to MATLAB using a *To Workspace* block.

Configuring the ‘To Workspace’ block: The ‘To Workspace’ block (found in the ‘Sinks’ folder) is used to a pass variable from Simulink to the MATLAB Workspace. The dialog box for setting up the transfer is shown in the Table 2.

Table 2 Dialog box for 'To Workspace' block



Put the variable name that will be used in MATLAB here. Use any name.

Set the 'Save format' to 'Array'.

Also, note that when you run the Simulink model, the time vector (tout) is automatically sent to the MATLAB Workspace. Also note that variables that are defined in MATLAB are automatically passed to Simulink.