**Michigan Tech**
**MEEM/EE 4295: Introduction to Propulsion Systems for Hybrid Electric Drive Vehicles**

**HW_1**
**Topics: Unit Conversions, Power Requirement, Introduction to Matlab and Model Based Design (Simulink)**
**See Syllabus for due date**

HW -1 is used to introduce the equations of motion of the vehicle, variations in power requirements over a **simple** drive cycle and Model Based Design. Starting at the tractive force at the tires, the rolling resistance and aerodynamic drag forces will be used to calculate the power and energy required for a drive cycle.

EPA drive cycles such as the UDDS, limits on powertrains, batteries, e-drives, etc. will be covered in later home work assignments.

The following example was used in EcoCAR to demonstrate modeling techniques, Matlab, tractive force and power requirements using a very basic drive cycle. We are given the following information about a simple drive cycle. Using the given information derive the equations that define the force and power required throughout the entire drive cycle.

Figure 1 shown below is the velocity versus time of our drive cycle. The cycle is as follows

1) WOT from 0-60
2) Steady state at 60
3) Apply brakes and decelerate to 40 (now behind a slow driver)
4) Steady state at 40
5) WOT from 40-70 to pass
6) Steady state at 70 to complete the pass
7) Decelerate to 55
8) Steady state at 55
9) Maintain 55 while going uphill, the elevation curve (road slope) is shown in Figure 2.

    i) As we crest the hill, we maintain 55 to the bottom of the hill

    ii) Steady state at 55 to almost the end of the drive cycle, apply the brakes and stop the vehicle.

The drive cycle is being used to demonstrate basic math modeling methods and how to formulate those in Matlab. It is NOT representative of "reasonable" driving.

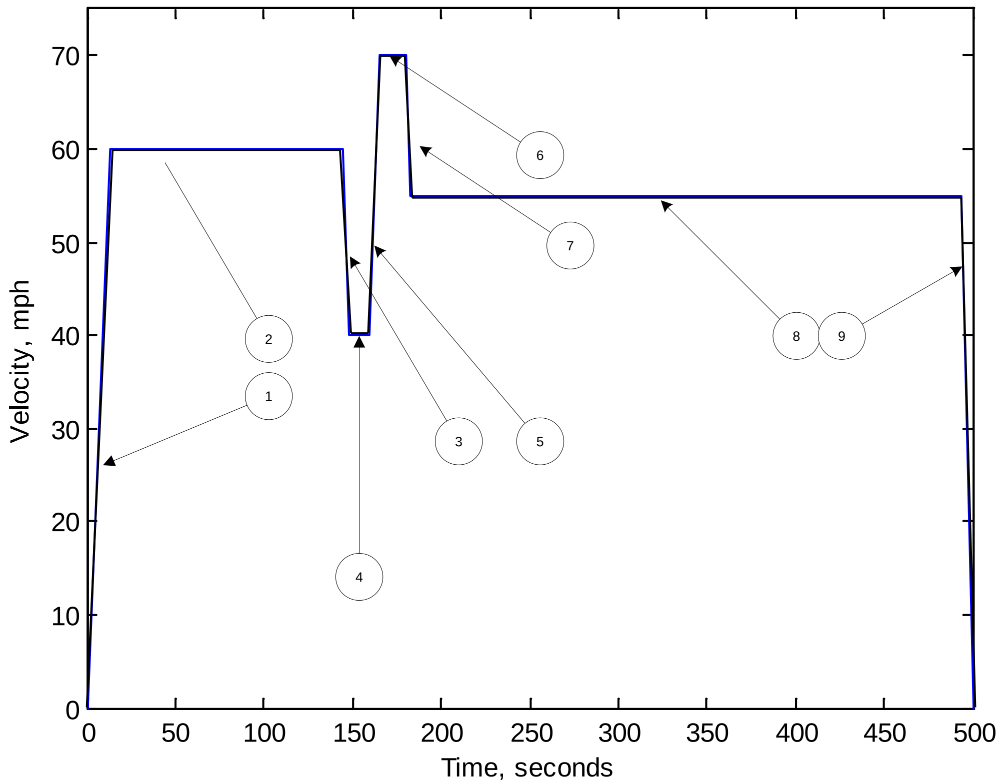


**Figure 1: Vehicle velocity versus time for the drive cycle**

Figure 2 shown below is the grade of road over our drive cycle. You may notice we are driving on level ground most of the time. At a time of 300 seconds, we start up an incline of 4.0 degrees. Once we reach the top of the hill the slope is now a -4.5 degrees. We maintain the 55 mph speed down the hill and by good use of our brakes and our magic regeneration system. Once we meet the bottom of the hill we maintain a constant speed for the remainder of the drive cycle until braking at the end. The entire drive cycle has taken 500 seconds.

For the given system we first model the vehicle. Figure 3 is our standard 2-D car model[1]. The forces acting on the vehicle are summed in the X-direction, our direction of motion. For most of the cycle the direction of travel is horizontal.

[1] Fundamentals of Vehicle Dynamics, T. D. Gillespie, SAE Publication, 1992. Same equations in the Modern Electric, Hybrid Electric and Fuel Cells Vehicles textbook. The road slope angle will be defined as $\theta$. The text book by Ehsani, Gao and Emadi has the same basic FBD in Chapter 2.

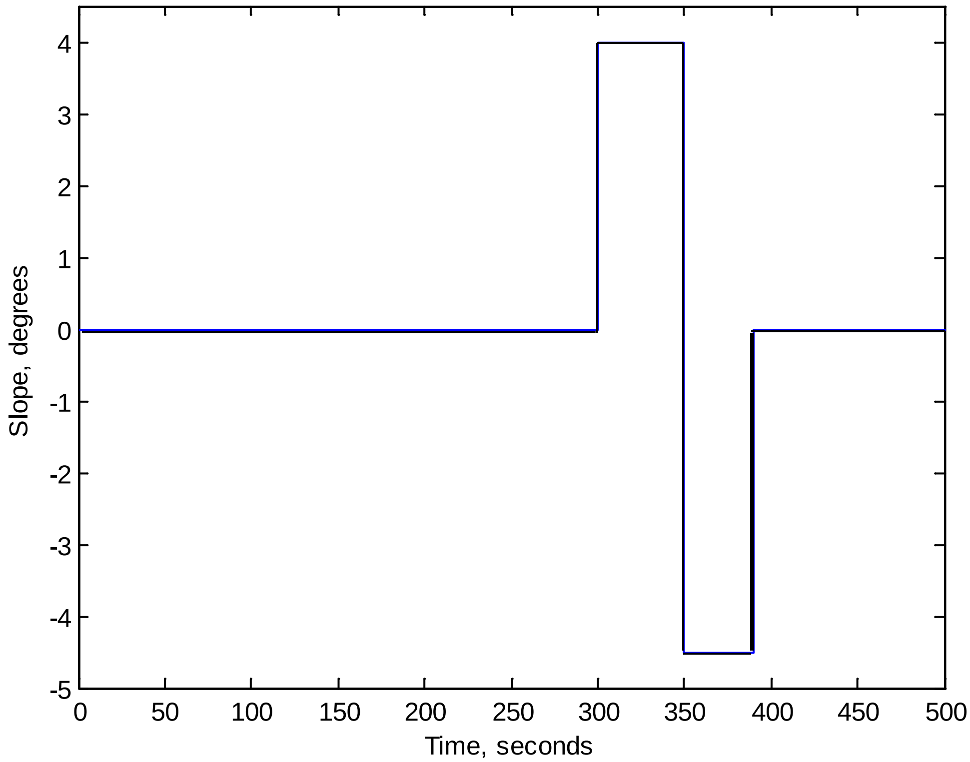


**Figure 2: Road slope for the drive cycle.**

For this problem include rolling resistance and wind drag, but neglect other power losses. The $C_d$ for this model is 0.40. You may assume a constant temperature of 72$^o$ F.

| Time | Velocity, mph | Slope, degrees |
|---|---|---|
| 0 | 0 | 0 |
| 13.0 | 60 | 0 |
| 145.0 | 60 | 0 |
| 148.0 | 40 | 0 |
| 160.0 | 40 | 0 |
| 165.5 | 70 | 0 |
| 180.0 | 70 | 0 |
| 183.0 | 55 | 0 |
| 300.0 | 55 | Starts up the hill, slope of 4 degrees |
| 350.0 | 55 | Apex or top of hill, starts down |
| 390.0 | 55 | Bottom of hill, slope returns to zero |
| 493.6 | 55 | 0, Start to brake |
| 500.0 | 0.0 | 0, vehicle at rest |

Table 1: Data use to generate Figure 1.

We are working some of the data in SAE units, convert them in the Matlab code.
Table 2:

| Item | Value | Units |
|---|---|---|
| Weight | 4,100 | Lbf |
| $\rho$ | 0.00236 | Lbf-sec$^2$/Ft$^4$ |
| g | 32.17 | Ft/sec$^2$ |
| $C_d$ | 0.4 | Unit less |
| A | 2.7 | meters$^2$ |
| $\theta$ | Given in table and figures | degrees |

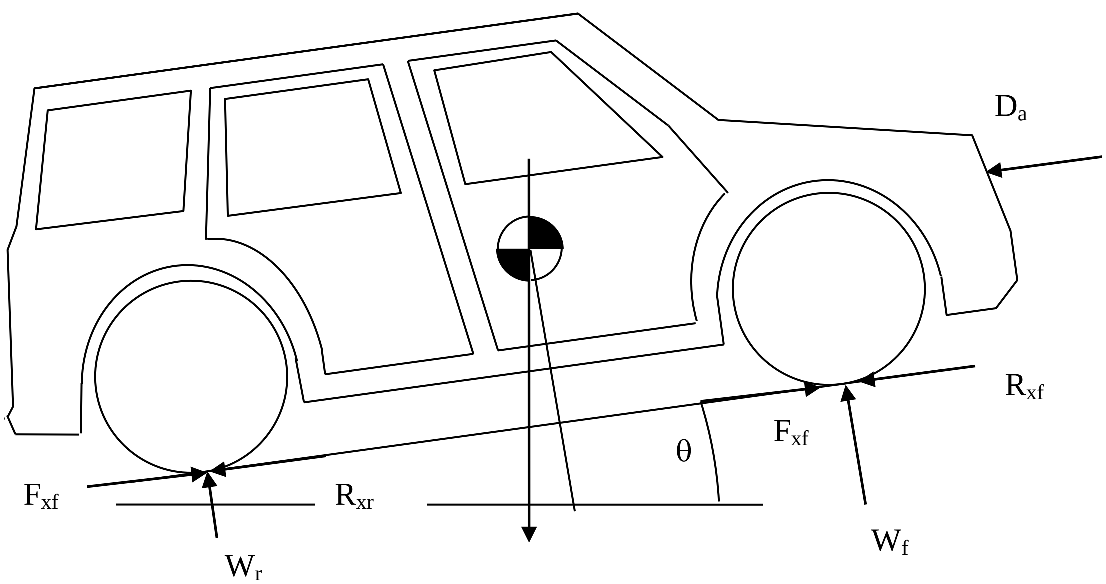


**Figure 3: Basic model of the vehicle and the external forces.**

| **Symbol** | **Definition** | **Units (fill in blank)** |
|---|---|---|
| $\mathbf{W_f}$ | **Front wheel load** | |
| $\mathbf{W_r}$ | **real wheel load** | |
| **ha** | **distance to location of drag forces** | |
| **h** | **distance to cg** | |
| **L** | **wheelbase** | |
| **b** | **distance to front wheels from cg** | |
| **c** | **distance to rear wheels from cg** | |
| $\mathbf{R_{xf}}$ | **rolling resistance, front wheels** | |

| | | |
|---|---|---|
| **$R_{xr}$** | **rolling resistance, rear wheels** | |
| **$F_{xf}$** | **tractive force, front wheels** | |
| **$F_{xr}$** | **tractive force, rear wheels** | |
| **$\theta$** | **Road slope** | |

When we sum the forces in the X-direction we get the following equation.

$$\sum F_X = m\ddot{x} = F_t - R_f - D_a - mg\sin(\theta)$$

Rolling resistance may be approximated as follows if we use a function that approximates the friction losses as a linear function of speed. It is commonly written as:

$$R_f = f_r \cdot W\left(1 + \frac{V}{100}\right)$$

f=0.01 and W is the weight in Lbf, V is in mph. Since the calculations in Gillespie are in Ft and Lbf., convert the equation to the form

$$R_f = f_r \cdot W\left(1 + \frac{\dot{X}}{K_{fv}}\right)$$

and determine the appropriate value of $K_{fv}$ for the velocity in Ft/sec. The variable $K_{fv}$ is the conversion to allow you to use Ft/sec for the velocity and replaces the 100 from the Gillespie book. The Ehsani, Gao and Emadi book 160 when the velocity units are kph.

Wind drag may be approximated by the following.

$$D_a = \frac{1}{2}\rho \cdot C_d \cdot A \cdot \dot{X}^2$$

The variables are
$\rho$=density of air, Lbf-sec$^2$/Ft$^4$ or kg/meter$^3$
$C_d$=drag coefficient, unit less
A=cross-sectional area, meter squared or feet squared

Now we have a fairly complete model of the external force acting on the vehicle and we can approximate the tractive forces needed to meet the drive cycle. When we sum the forces in the X-direction we derived the following equation in class.

$$\sum F_X = m\ddot{x} = F_t - R_f - D_a - mg\sin(\theta)$$

Using the drive cycle given, do the following.
**Home Work Assignment**

**Part I: Introduction to Matlab**

1) Write a Matlab code to:
   a) Put the time, velocity and road slope (seconds, MPH, degrees) in arrays. They will appear in the Workspace once you execute the .m file
   b) List the variables needed, writes to the Workspace.
   c) Enter the conversion factors you need.
2) Plot the drive cycle for validation. A basic 2-D plot with labels should be adequate.

**Part II: Using Matlab**

**Note:** In this section you will manipulate the data between the given points (linear interpolation) so you may calculate the forces between the given points. In later home works, this will be done in Simulink.

1) Using the interpolator function, fill in the time and velocity values between the given points. Use at least 50 steps between points. Use names of velocity and time that are meaningful. Do the same for slope. See both caveats below. The description from Matlab is shown below.

> "`yi = interp1(x,Y,xi)` interpolates to find `yi`, the values of the underlying function `Y` at the points in the vector or array `xi`. `x` must be a vector. Y can be a scalar, a vector, or an array of any dimension, subject to the following conditions:"

In "yooperese," if you have two arrays (x,Y) like the ones given with only several points and you have an array xi with the same starting and ending values of x and with many more evenly distributed points, you will get back yi as the interpolation of Y.

2) Use the "new arrays" of time and velocity to determine the acceleration at each step. You may use the Matlab function, "diff" to determine the delta T's and delta V's for an array. The description from Matlab is shown below.

> "`Y = diff(X)` calculates differences between adjacent elements of `X`.
>
> If `X` is a vector, then `diff(X)` returns a vector, one element shorter than `X`, of differences between adjacent elements:"

To determine acceleration, in Matlab you may use

Accel=diff(Vel)./diff(Time); Or in equation form, $a_X = \frac{\Delta v}{\Delta t}$

I will demonstrate in class why the "period" in the above equation.

***As a caveat, if you have a vector (array) of n-values, it will return n-1 differences.***

***If you use*** `interp1` ***to fill in the slope arrays, the infinite change of slope at given point needs to be considered. There are several methods to accomplish this.***

3)

a) Using the "new" arrays for time, velocity, acceleration and slope, determine the rolling resistance, aerodynamic drag and tractive force for the drive cycle. Plot them versus time. See note below on units.

b) Calculate the power required for the rolling resistance, aerodynamic drag and tractive force. Plot them versus time.

4) Validate the solution at several points. Validate means an "independent" check of the results. Hint: Think of ways to arrive at approximate values of the variables using independent derivations.

**Units**

The inputs are in mixed units. In your Matlab model you should convert all your variables to SI, In the plots, show as Lbf and Newtons, horsepower and kilowatts (or watts) and output velocity in MPH since we still drive using MPH, kph can be added if you want. Once the "data" or variables are imported into the Simulink model in the newt homework, keep them in the same units. In later problems, you may want to convert rpm to rad/sec inside of Simulink.

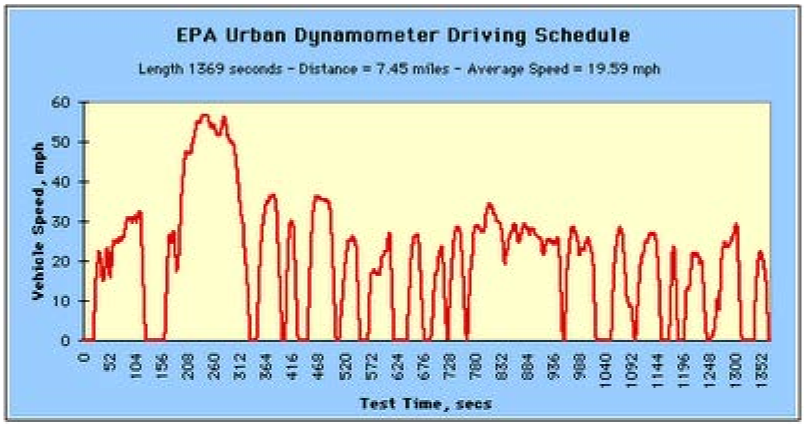


**Figure 4: The EPA Urban Dynamometer Driving Schedule, commonly known as UDDS**