Automotive Powertrain and Alternator Modeling

Objectives:

1. To learn mixed domain modeling
2. Simple Model of Powertrain, a mechanical system
3. Simple Model of Alternator, a mixed system which transforms mechanical energy to electrical energy

I. Powertrain Modeling

Powertrain refers to engine and transmission combination. In general, it is a name applied to a group of components used to transmit engine power to the driving wheels. It can consist of engine, clutch, transmission, universal joints, drive shaft, differential gear, and axle shafts. Powertrain components are matched according to driver needs such as high torque, fuel economy, or convenience. [www.Radleyauto.com]. A simple model is simply represented by the typical driving speed profile in city, highway, and freeway driving. The speed profile is represented using revolution per minute (rpm), the output converts the unit to rad/sec.

| S(rpm) | 0 | 550 | 1800 | 550 | 1005 | 2144 | 550 | 1005 | 1505 | 2150 |
| T(ms)  | 1000 | 25000 | 20000 | 25000 | 10000 | 30000 | 55000 | 5000 | 10000 | 20000 |

| S(rpm) | 1505 | 550 | 1005 | 2144 | 550 | 1005 | 1505 | 2150 | 1505 | 550 |
| T(ms)  | 20000 | 70000 | 5000 | 30000 | 40000 | 5000 | 5000 | 20000 | 15000 | 75000 |

| S(rpm) | 1005 | 2144 | 550 | 1005 | 1505 | 2150 | 1505 | 550 | 0 |
| T(ms)  | 5000 | 30000 | 35000 | 5000 | 10000 | 20000 | 5000 | 5000 | stop |

**TASK 1.** Complete the outline VHDL-AMS code of the powertrain presented in LISTING 1 that implements the speed profile given in TABLE 1. Edit its symbol to look like the symbol shown in Figure 2. Then simulate and compare your result with the waveform shown in Figure 1.

**LISTING 1**

LIBRARY IEEE; use IEEE.math_real.all;
USE IEEE.electrical_systems.ALL;
USE IEEE.mechanical_systems.ALL;

ENTITY MyPowertrain IS
    PORT (TERMINAL n_out : rotational_v);
END ENTITY MyPowertrain;

ARCHITECTURE behav OF mypowertrain IS
    QUANTITY vel ACROSS tor THROUGH n_out TO ROTATIONAL_V_REF;
    signal vhe_rpm: real := 0.0;
II. Alternator Modeling

Alternator transforms the mechanical energy to electrical energy. A simple model assumed to have no inertia, that is the torque depends on the input speed and the electrical power output. TABLE 2 shows the mechanical to electrical conversions characteristics of the alternator. In the implementation, the mechanical speed is in rad/sec rather than rpm. The complete VHDL-AMS model is given LISTING 2.

<table>
<thead>
<tr>
<th>S(rpm)</th>
<th>0</th>
<th>1000</th>
<th>1500</th>
<th>2000</th>
<th>3000</th>
<th>4000</th>
<th>6000</th>
<th>10000</th>
<th>14000</th>
<th>∞</th>
</tr>
</thead>
<tbody>
<tr>
<td>I(A)</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>60</td>
<td>70</td>
<td>85</td>
<td>90</td>
<td>100</td>
<td>105</td>
<td>105</td>
</tr>
</tbody>
</table>

CONSTANT OM2N : REAL := 9.549296; -- This is equal to 60/2*pi
The model calculates a torque output based on the following equations:

Torque* Omega = Voltage * Current
Torque = Voltage * Current / Omega

--LISTING 2.
LIBRARY IEEE;
USE IEEE.ELECTRICAL_SYSTEMS.ALL;
USE IEEE.MECHANICAL_SYSTEMS.ALL;
ENTITY Alternator IS
    PORT (TERMINAL rot_in : ROTATIONAL_V;
TERMINAL p_out, m_out : ELECTRICAL;
QUANTITY speed_out : OUT REAL;
QUANTITY current_out : OUT REAL;
QUANTITY torque_out : OUT REAL);
END ENTITY Alternator;

ARCHITECTURE behav OF alternator IS
  CONSTANT OM2N : REAL := 9.549296; -- This is equal to 60/2*pi
  QUANTITY v ACROSS i THROUGH p_out TO m_out;
  QUANTITY omega ACROSS torque THROUGH rot_in TO ROTATIONAL_V_REF;

  FUNCTION CH (x: REAL) RETURN REAL IS
    CONSTANT X1 : REAL := 1000.0;
    CONSTANT Y1 : REAL := 0.0;
    CONSTANT X2 : REAL := 1500.0;
    CONSTANT Y2 : REAL := 30.0;
    CONSTANT X3 : REAL := 2000.0;
    CONSTANT Y3 : REAL := 60.0;
    CONSTANT X4 : REAL := 3000.0;
    CONSTANT Y4 : REAL := 70.0;
    CONSTANT X5 : REAL := 4000.0;
    CONSTANT Y5 : REAL := 85.0;
    CONSTANT X6 : REAL := 6000.0;
    CONSTANT Y6 : REAL := 90.0;
    CONSTANT X7 : REAL := 10000.0;
    CONSTANT Y7 : REAL := 100.0;
    CONSTANT X8 : REAL := 14000.0;
    CONSTANT Y8 : REAL := 105.0;
    VARIABLE result: REAL :=0.0;
    BEGIN
      IF X < X1 THEN result := Y1;
      ELSE
        IF X < X2 THEN result := Y1+(X-X1) / (X2-X1) * (Y2-Y1) ;
        ELSE
          IF X < X3 THEN result := Y2+(X-X2) / (X3-X2) * (Y3-Y2) ;
          ELSE
            IF X < X4 THEN result := Y3+(X-X3) / (X4-X3) * (Y4-Y3) ;
            ELSE
              IF X < X5 THEN result := Y4+(X-X4) / (X5-X4) * (Y5-Y4) ;
              ELSE
                IF X < X6 THEN result := Y5+(X-X5) / (X6-X5) * (Y6-Y5) ;
                ELSE
                  IF X < X7 THEN result := Y6+(X-X6) / (X7-X6) * (Y7-Y6) ;
                  ELSE
                    IF X < X8 THEN result := Y7+(X-X7) / (X8-X7) * (Y8-Y7) ;
                    ELSE result := Y8;
                  END IF;
                END IF;
              END IF;
            END IF;
          END IF;
        END IF;
      END IF;
    RETURN result;
END FUNCTION CH;
BEGIN
  i == -CH(omega*OM2N);
  torque == -i*v / (omega + 1.0E-12);
  speed_out == omega*OM2N;
  torque_out == torque;
  current_out == i;
END ARCHITECTURE behav;

**TASK 2.** Create the VHDL-AMS code of the alternator shown in LISTING 2. Edit its symbol to look like the symbol shown in Figure 2.

**TASK 3.** Draw the schematic shown in Figure 2 using your powertrain and alternator models. Simulate for 600 sec and display the powertrain vhe_rpm and alternator current_out, and compare with Figure 3.

![Figure 2. Powertrain and Alternator Schematic](image)

![Figure 3(a) Powertrain speed profile in rpm](image)

![Figure 3(b) Alternator output current.](image)
**TASK 4.** Modify the FUNCTION CH in LISTING 2 to avoid the long chain of IF –THEN-ELSE statement. Using a two dimensional array to represent TABLE 2, search the table to find the two entries where interpolation is to be done. NOTE the infinity entry is represented by the largest positive real number in Simploter, 9.8e99. Repeat TASK 3 and show that your implementation resulted in identical response as shown in Figure 3(b).

**REFERENCES**