**Report**

**On**

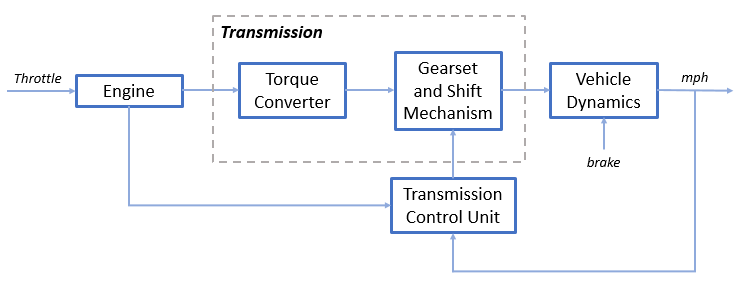
**Modeling an Automatic Transmission Controller**

### Skills that are demonstrated in this project are as follows:

1. Callbacks
2. Data Inspector
3. Solver selection strategy
4. MATLAB function block
5. Look-up table
6. Signal Builder to generate test signals.

### Analysis and Physics

The figure below shows the power flow in a typical automotive drivetrain. Nonlinear ordinary differential equations model the engine, four-speed automatic transmission, and vehicle. The model discussed in this example directly implements the blocks from this figure as modular Simulink subsystems. On the other hand, the logic and decisions made in the Transmission Control Unit (TCU) do not lend themselves to well-formulated equations. TCU is better suited for a Stateflow representation. Stateflow monitors the events which correspond to important relationships within the system and takes the appropriate action as they occur.



The throttle opening is one of the inputs to the engine. The engine is connected to the impeller of the torque converter which couples it to the transmission (see Equation 1).

**Equation 1**

$$I_{ei} \dot{N}_e = T_e -T_i $$

$$ N_e = \mbox{ engine speed (RPM)}$$

$$I_{ei} = \mbox{ moment of inertia of the engine and the impeller}$$

$$T_e, T_i = \mbox{ engine and impeller torque}$$

The input-output characteristics of the torque converter can be expressed as functions of the engine speed and the turbine speed. In this example, the direction of power flow is always assumed to be from the impeller to the turbine (see Equation 2).

**Equation 2**

$$T_i = \frac{N_e^2}{K^2}$$

$$K= f_2 \frac{N_{in}}{N_e} = \mbox{ K-factor (capacity)}$$

$$N_{in} = \mbox{ speed of turbine (torque converter output) = transmission input speed (RPM)}$$

$$R_{TQ} = f_3 \frac{N_{in}}{N_e} = \mbox{ torque ratio}$$

The transmission model is implemented via static gear ratios, assuming small shift times (see Equation 3).

**Equation 3**

$$R_{TR} = f_4(gear) = \mbox{ transmission ratio}$$

$$T_{out} = R_{TR} T_{in}$$

$$N_{in} = R_{TR} N_{out}$$

$$T_{in}, T_{out} = \mbox{ transmission input and output torques}$$

$$N_{in}, N_{out} = \mbox{ transmission input and output speed (RPM)}$$

The final drive, inertia, and a dynamically varying load constitute the vehicle dynamics (see Equation 4).

**Equation 4**

$$ I_v \dot{N}_w = R_{fd}(T_{out}-T_{load})$$

$$I_v = \mbox{ vehicle inertia}$$

$$N_w = \mbox{ wheel speed (RPM)}$$

$$R_{fd} = \mbox{ final drive ratio}$$

$$T_{load} = f_5(N_w) = \mbox{ load torque}$$

The load torque includes both the road load and brake torque. The road load is the sum of frictional and aerodynamic losses (see Equation 5).

**Equation 5**

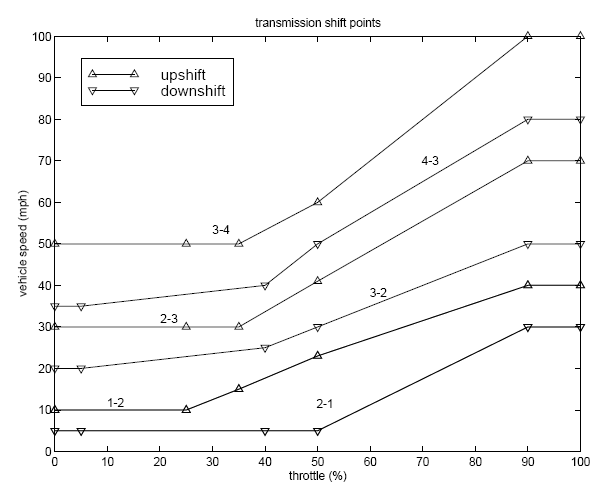
$$ T_{load} = sgn(mph) (R_{load0} + R_{load2} mph^2 + T_{brake}) $$

$$ R_{load0}, R_{load2} = \mbox{ friction and aerodynamic drag coefficients} $$

$$ T_{load}, T_{brake} = \mbox{ load and brake torques} $$

$$ mph = \mbox{ vehicle linear velocity}$$

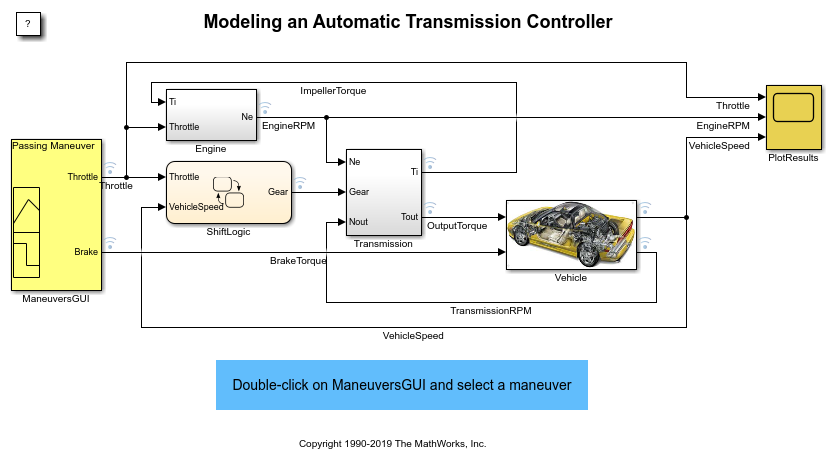
The model programs the shift points for the transmission according to the schedule shown in the figure below. For a given throttle in a given gear, there is a unique vehicle speed at which an upshift takes place. The simulation operates similarly for a downshift.



### Modeling

When you open the model, the Initial conditions are set in the Model Workspace.

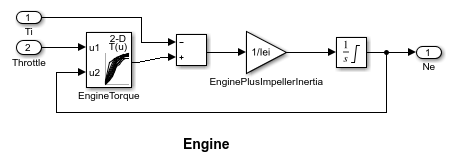
The top-level diagram of the model is shown in the figure below. To run the simulation, on the Simulation tab, click **Run**. Note that the model logs relevant data to MATLAB Workspace in a data structure called sldemo\_autotrans\_output. Logged signals have a blue indicator. After you run the simulation, you can view the components of the data structure by typing sldemo\_autotrans\_output in MATLAB Command Window. Also note that the units appear on the subsystem icons and signal lines.



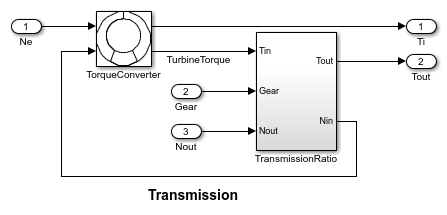
### Modeling

The Simulink model shown above is composed of modules which represent the engine, transmission, and the vehicle, with an additional shift logic block to control the transmission ratio. User inputs to the model are in the form of throttle (given in percent) and brake torque (given in ft-lb). The user inputs throttle and brake torques using the ManeuversGUI interface.

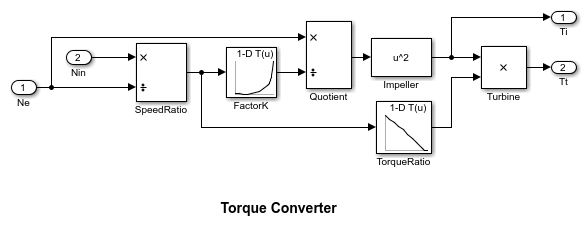
The Engine subsystem consists of a two-dimensional table that interpolates engine torque versus throttle and engine speed. The figure below shows the composite Engine subsystem. Double click on this subsystem in the model to view its structure.



The TorqueConverter and the TransmissionRatio blocks make up the Transmission subsystem, as shown in the figure below. Double click on the Transmission subsystem in the model window to view its components.



The TorqueConverter is a masked subsystem, which implements Equation 2. To open this subsystem, right click on it and select **Mask** > **Look Under Mask** from the drop-down menu. The mask requires a vector of speed ratios ( Nin/Ne ) and vectors of K-factor (f2) and torque ratio (f3). This figure shows the implementation of the TorqueConverter subsystem.



The transmission ratio block determines the ratio shown in Table 1 and computes the transmission output torque and input speed, as indicated in Equation 3. The figure that follows shows the block diagram for the subsystem that realizes this ratio in torque and speed.

**Table 1:** Transmission gear ratios

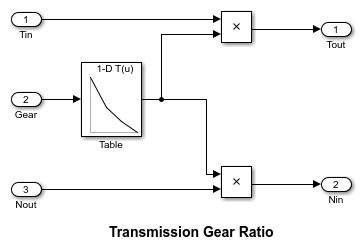
gear Rtr = Nin/Ne

1 2.393

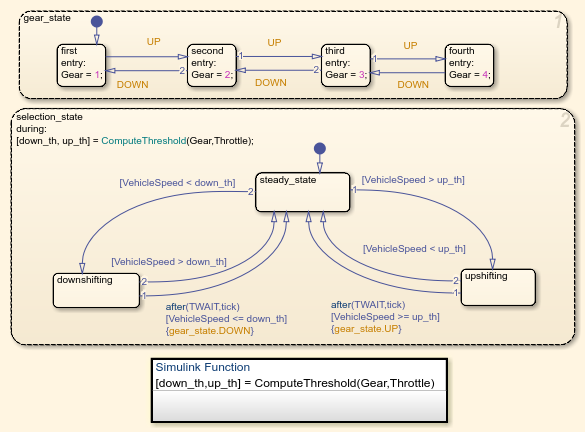
2 1.450

3 1.000

4 0.677



The Stateflow block labeled ShiftLogic implements gear selection for the transmission. Double click on ShiftLogic in the model window to open the Stateflow diagram. The Model Explorer is utilized to define the inputs as throttle and vehicle speed and the output as the desired gear number. Two dashed AND states keep track of the gear state and the state of the gear selection process. The overall chart is executed as a discrete-time system, sampled every 40 milliseconds. The Stateflow diagram shown below illustrates the functionality of the block.

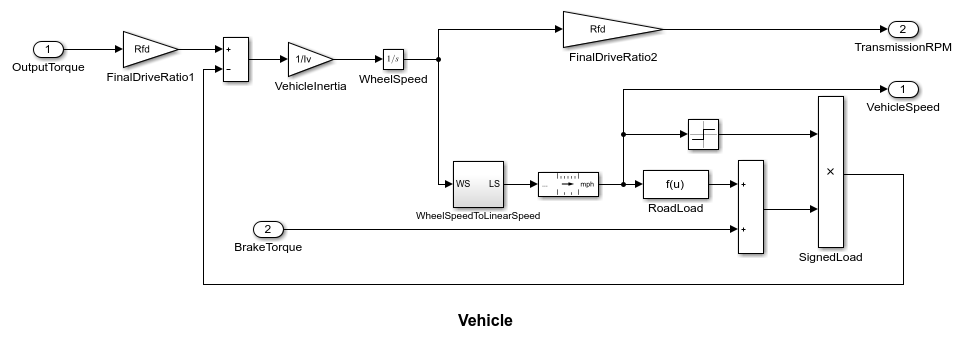


The shift logic behavior can be observed during simulation by enabling animation in the Stateflow debugger. The selection\_state (always active) begins by performing the computations indicated in its during function. The model computes the upshift and downshift speed thresholds as a function of the instantaneous values of gear and throttle. While in steady\_state, the model compares these values to the present vehicle speed to determine if a shift is required. If so, it enters one of the confirm states (upshifting or downshifting), which records the time of entry.

If the vehicle speed no longer satisfies the shift condition, while in the confirm state, the model ignores the shift and it transitions back to steady\_state. This prevents extraneous shifts due to noise conditions. If the shift condition remains valid for a duration of TWAIT ticks, the model transitions through the lower junction and, depending on the current gear, it broadcasts one of the shift events. Subsequently, the model again activates steady\_state after a transition through one of the central junctions. The shift event, which is broadcast to the gear\_selection state, activates a transition to the appropriate new gear.

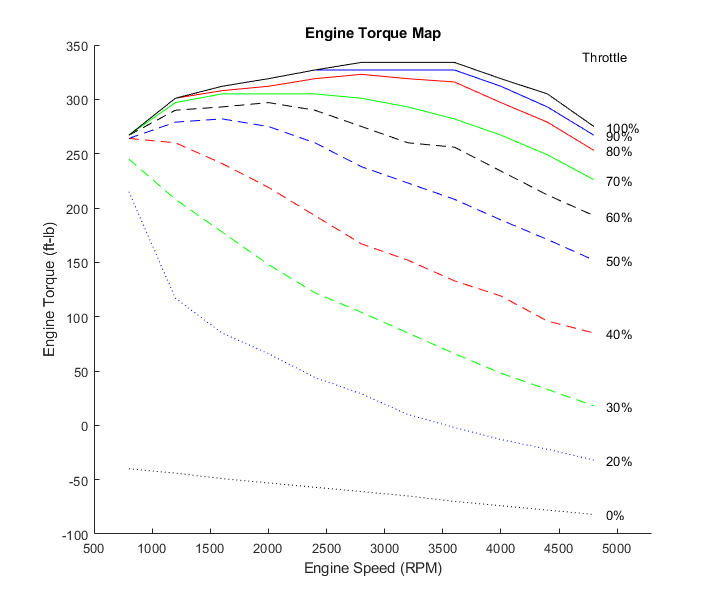
For example, if the vehicle is moving along in second gear with 25% throttle, the state second is active within gear\_state, and steady\_state is active in the selection\_state. The during function of the latter, finds that an upshift should take place when the vehicle exceeds 30 mph. At the moment this becomes true, the model enters the upshifting state. While in this state, if the vehicle speed remains above 30 mph for TWAIT ticks, the model satisfies the transition condition leading down to the lower right junction. This also satisfies the condition [|gear == 2|] on the transition leading from here to steady\_state, so the model now takes the overall transition from upshifting to steady\_state and broadcasts the event UP as a transition action. Consequently, the transition from second to third is taken in gear\_state which completes the shift logic.

The Vehicle subsystem uses the net torque to compute the acceleration and integrate it to compute the vehicle speed, per Equation 4 and Equation 5. The Vehicle subsystem is masked. To see the structure of the Vehicle block, right click on it and select **Mask** > **Look Under Mask** from the drop-down menu. The parameters entered in the mask menu are the final drive ratio, the polynomial coefficients for drag friction and aerodynamic drag, the wheel radius, vehicle inertia, and initial transmission output speed.

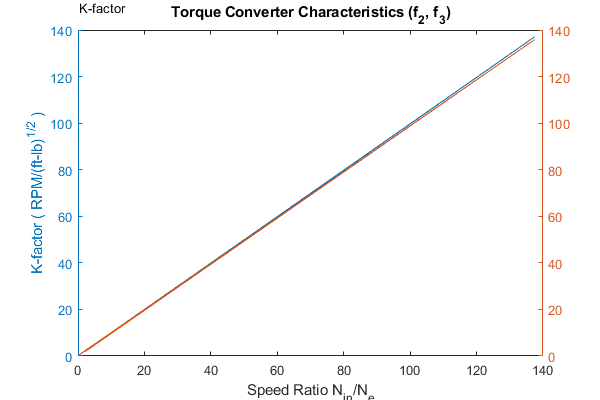


### Results

The engine torque map, and torque converter characteristics used in the simulations are shown below.



Get the FactorK (second row) and the TorqueRatio (third row) vs SpeedRatio(first row)



The first simulation (passing maneuver) uses the throttle schedule given in Table 2 (this data is interpolated linearly).

**Table 2:** Throttle schedule for first simulation (passing maneuver)

Time (sec) Throttle (%)

0 60

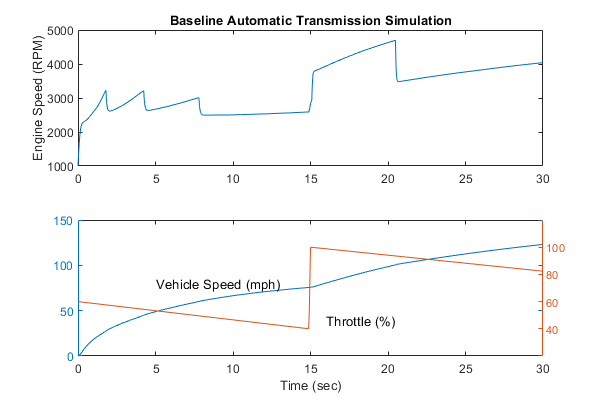
14.9 40

15 100

100 0

200 0

The first column corresponds to time; the second column corresponds to throttle opening in percent. In this case no brake is applied (brake torque is zero). The vehicle speed starts at zero and the engine at 1000 RPM. The following figure shows the plot for the baseline results, using the default parameters. As the driver steps to 60% throttle at t=0, the engine immediately responds by more than doubling its speed. This brings about a low speed ratio across the torque converter and, hence, a large torque ratio. The vehicle accelerates quickly (no tire slip is modeled) and both the engine and the vehicle gain speed until about t = 2 sec, at which time a 1-2 upshift occurs. The engine speed characteristically drops abruptly, then resumes its acceleration. The 2-3 and 3-4 upshifts take place at about four and eight seconds, respectively. Notice that the vehicle speed remains much smoother due to its large inertia.



At t=15sec, the driver steps the throttle to 100% as might be typical of a passing maneuver. The transmission downshifts to third gear and the engine jumps from about 2600 RPM to about 3700 RPM. The engine torque thus increases somewhat, as well as the mechanical advantage of the transmission. With continued heavy throttle, the vehicle accelerates to about 100 mph and then shifts into overdrive at about t = 21 sec. The vehicle cruises along in fourth gear for the remainder of the simulation. Double click on the ManeuversGUI block and use the graphical interface to vary the throttle and brake history.

### Conclusions

You can enhance this basic system in a modular manner, for example, by replacing the engine or transmission with a more complex model. You can build large systems within this structure via step-wise refinement. The seamless integration of Stateflow control logic with Simulink signal processing enables the construction of a model that is efficient and visually intuitive.