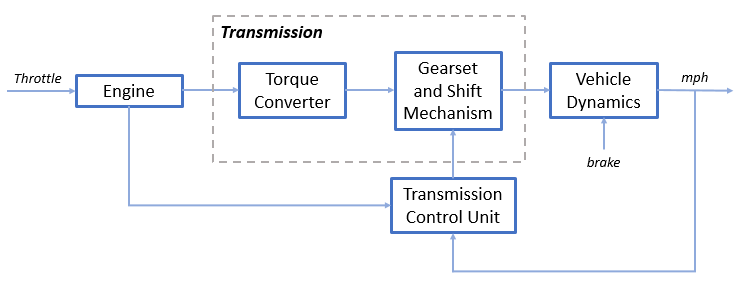
**Modeling an Automatic Transmission Controller**

In this project, Stateflow shows its strength by performing the function of gear selection in an automatic transmission. This function is combined with the drivetrain dynamics in a natural and intuitive manner by incorporating a Stateflow block in the Simulink block diagram.

### 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 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.

### Equations used in the project

**Equation 1**

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

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

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

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

**Equation 2**

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

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

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

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

**Equation 3**

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

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

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

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

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

**Equation 4**

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

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

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

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

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

**Equation 5**

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

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

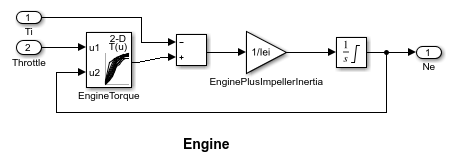
$$ T_{load}, T_{brake} = \mbox{ load and brake torques} $$

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

### Modeling

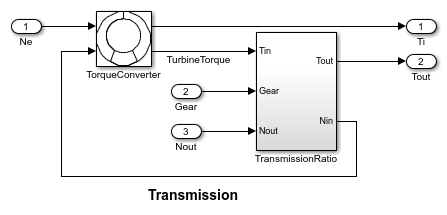
**Engine:**

The Engine subsystem consists of a two-dimensional **lookup table** that interpolates engine torque versus throttle and engine speed. The figure below shows the composite Engine subsystem.



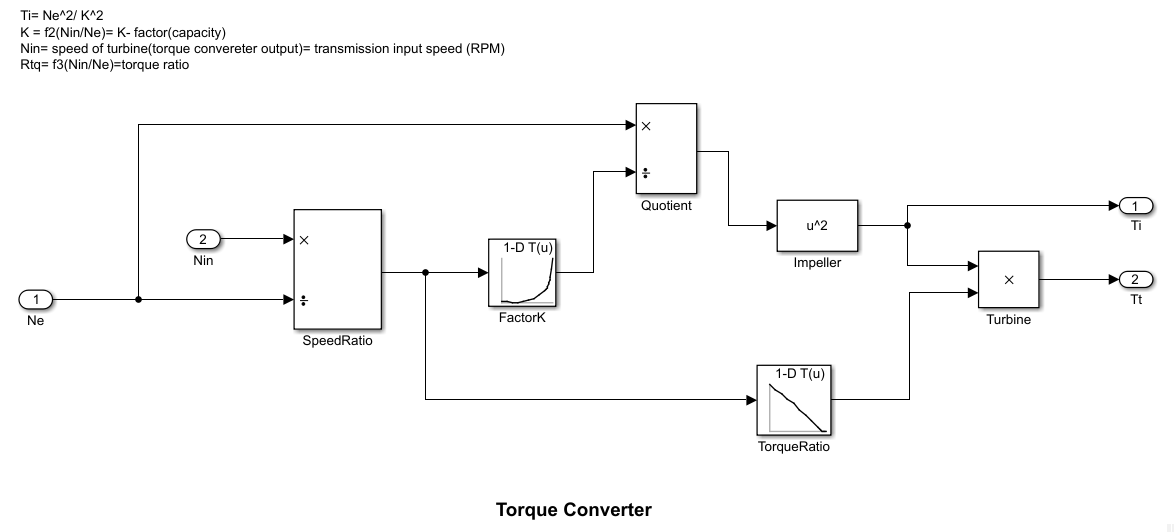
**Transmission:**

The TorqueConverter and the TransmissionRatio blocks make up the Transmission subsystem, as shown in the figure below.



**TorqueConverter:**

The TorqueConverter is a masked subsystem, which implements Equation 2. The mask requires a vector of speed ratios (Nin/Ne) and vectors of K-factor (f2) and torque ratio (f3). These vectors are passed through 1-D **lookup table** as shown. This figure shows the implementation of the TorqueConverter subsystem.



**Transmission Gear Ratio:**

The transmission ratio block determines the ratio shown in Table 1 and computes the transmission output torque and input speed with the help of a 1-D **lookup table**, 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

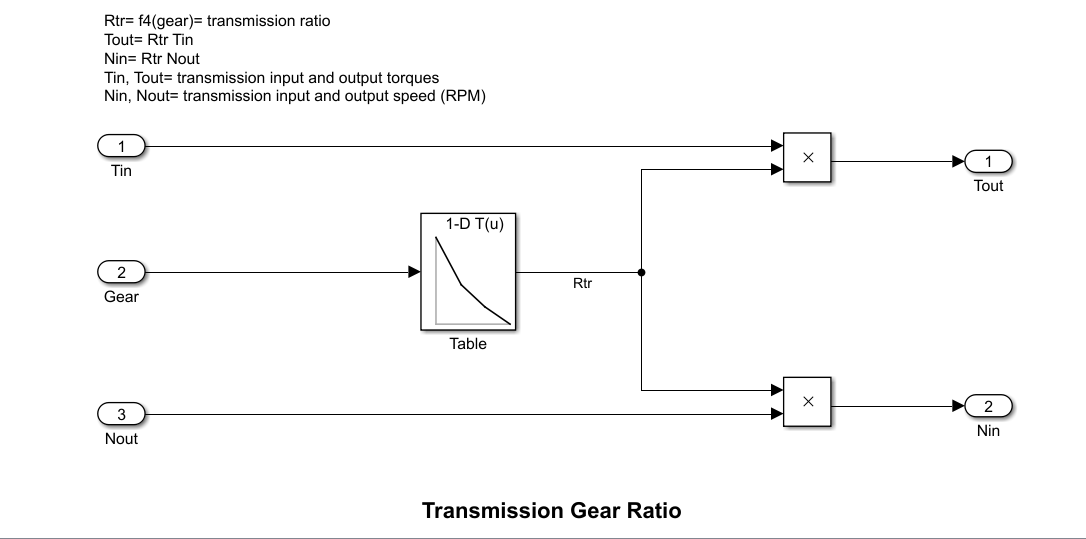
Gear Rtr = Nin/Ne

1 2.393

2 1.450

3 1.000

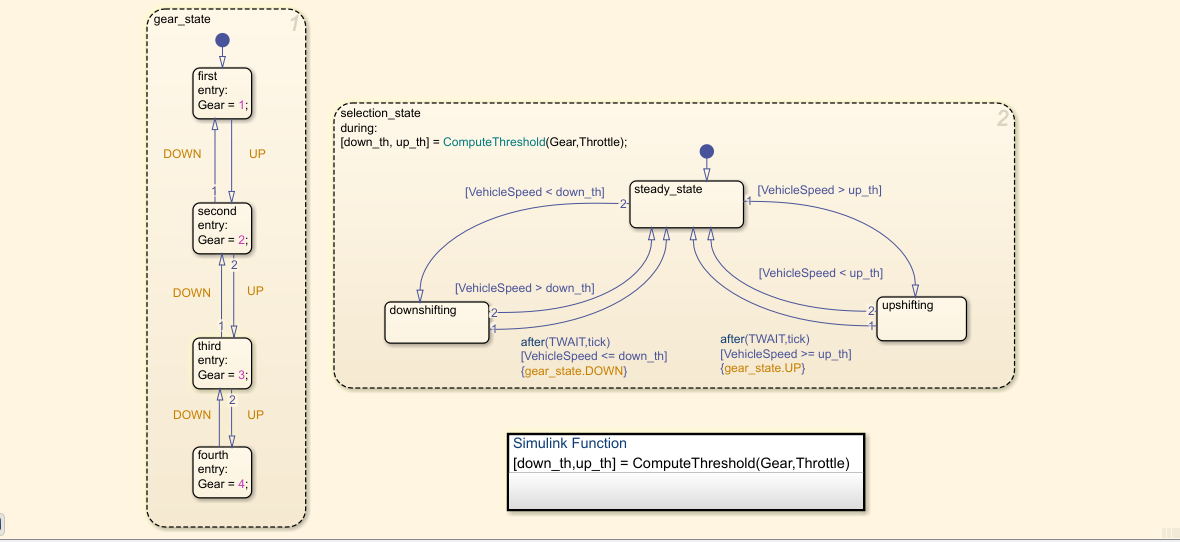
4 0.677



**ShiftLogic:**

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.