**REPORT**

**Automatic Gear Transmission Control**

Week 3 Problem 3 Milestone

Done By

Vignesh Babu A\_2005412

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **SNO** | **Content** | **Page** |
| 1 | Introduction |  |
| 2 | System Design |  |
| 3 | Solver Selection |  |
| 4 | Result |  |
| 5 | Skills Implemented |  |

**INTRODUCTION**

Automatic Transmission Controller is designed in this project. Gear shift happens when there is change in vehicle speed from a specified threshold.

**SYSTEM DESIGN**

Torque converter, gearset, shift mechanism and vehicle dynamics are modelled in differential equations. 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.

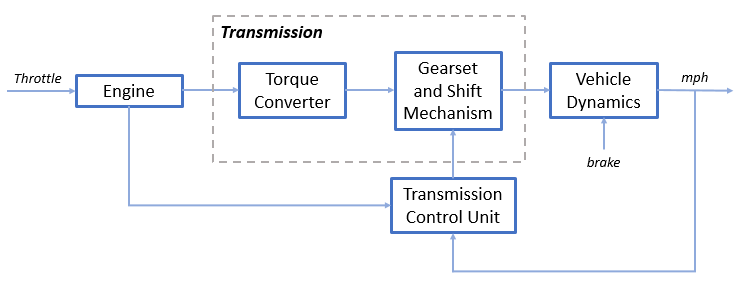
****

Figure 1 Block Diagram of the System

**Equations representing the system:**

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 Figure 2).

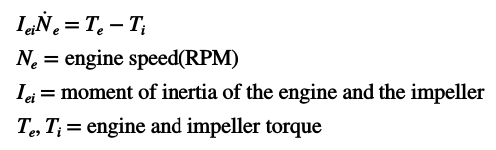


Figure 2 Equation 1

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 Figure 3).

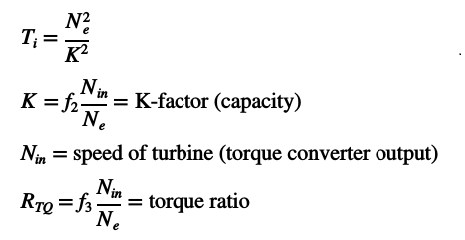


Figure 3 Equation 2

The transmission model is implemented via static gear ratios, assuming small shift times (see Figure 4).

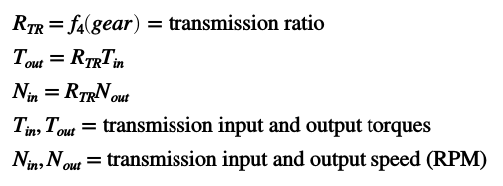


Figure 4 Equation 3

The final drive, inertia, and a dynamically varying load constitute the vehicle dynamics (see Figure 5).

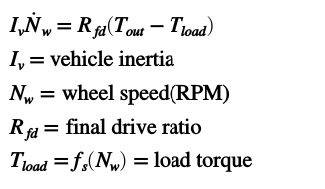


Figure 5 Equation 4

The load torque includes both the road load and brake torque. The road load is the sum of frictional and aerodynamic losses (see Figure 6).

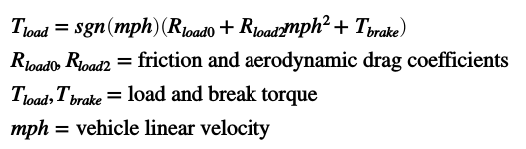


Figure 6 Equatio 5

**Modelling:**

The entire system consists of shift logic block for automatic gear change, transmission block to represent the real transmission system, Engine represents the engine of the system, and vehicle block represents the entire vehicles dynamics.

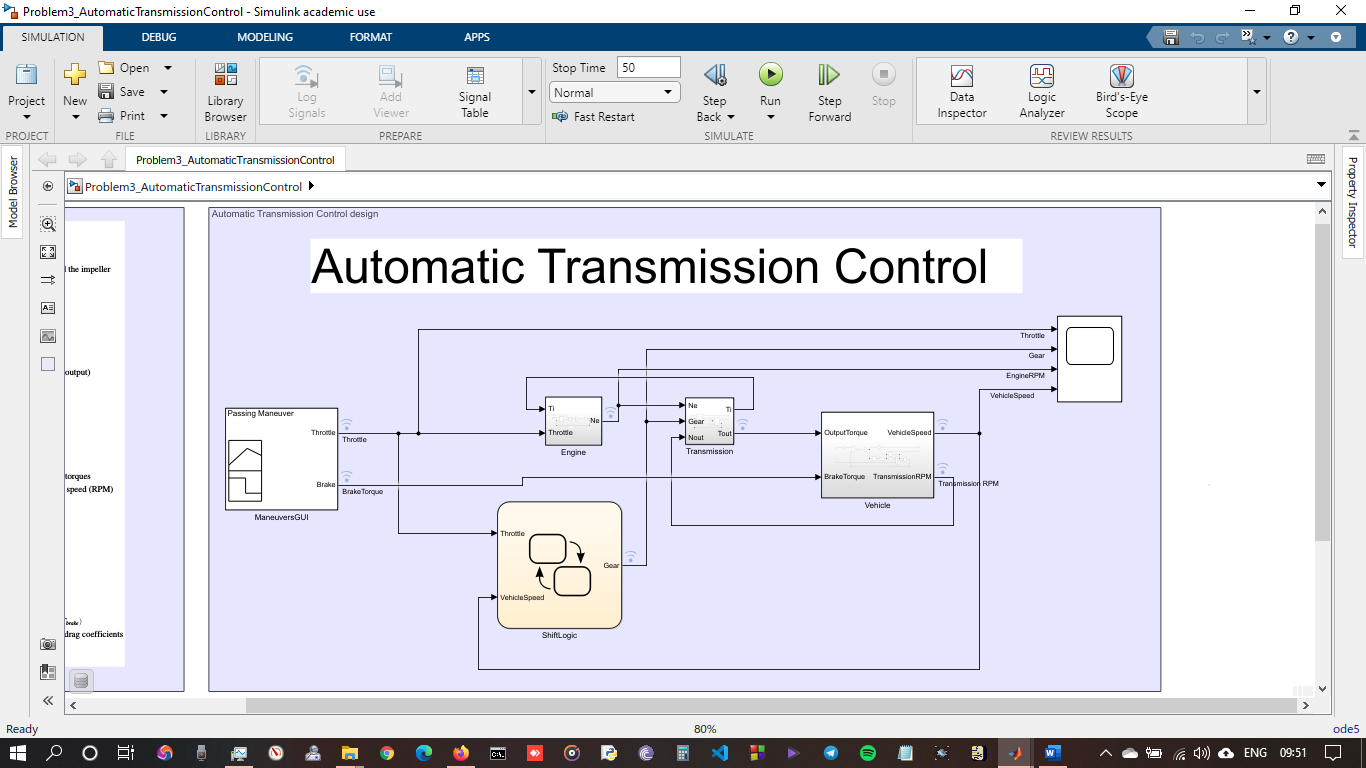


Figure 7 System Architecture

**Engine Subsystem:**

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.

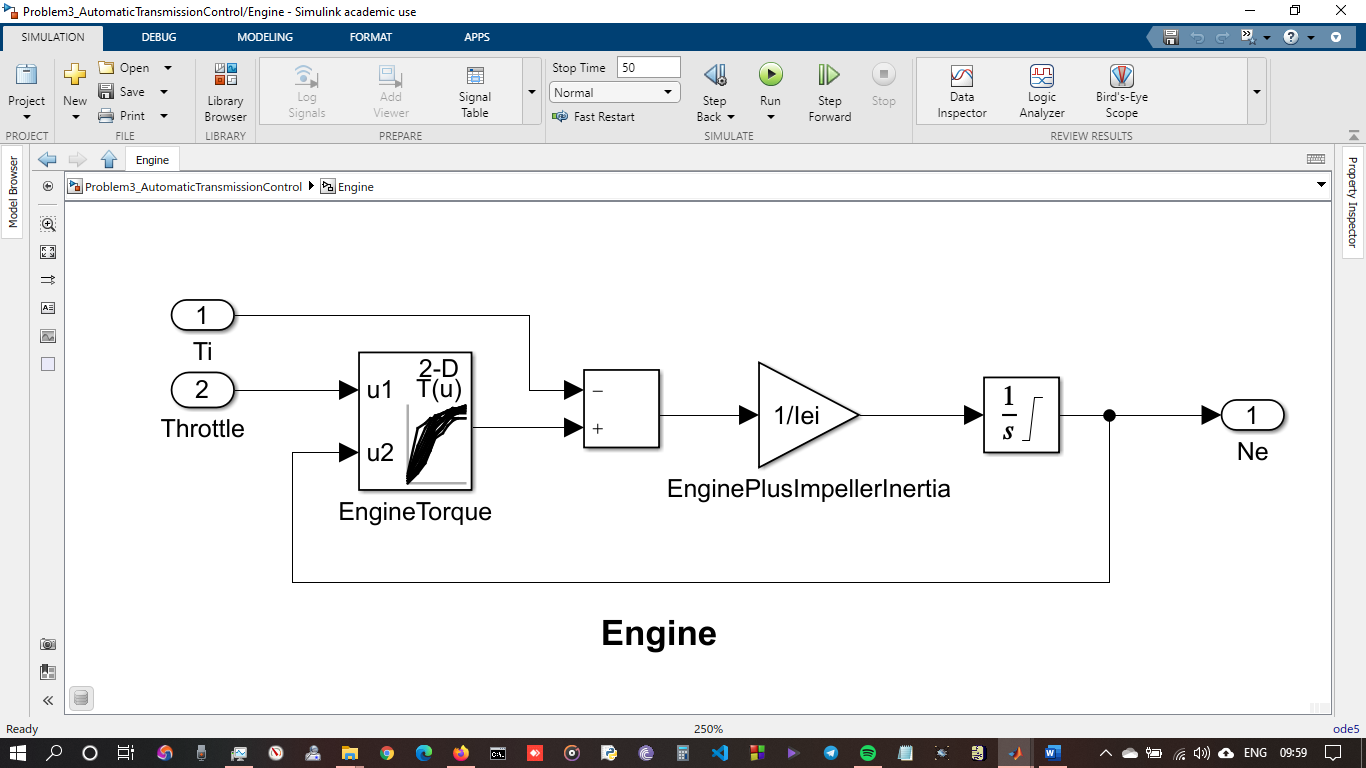


Figure 8 Engine

**Transmission Subsystem:**

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

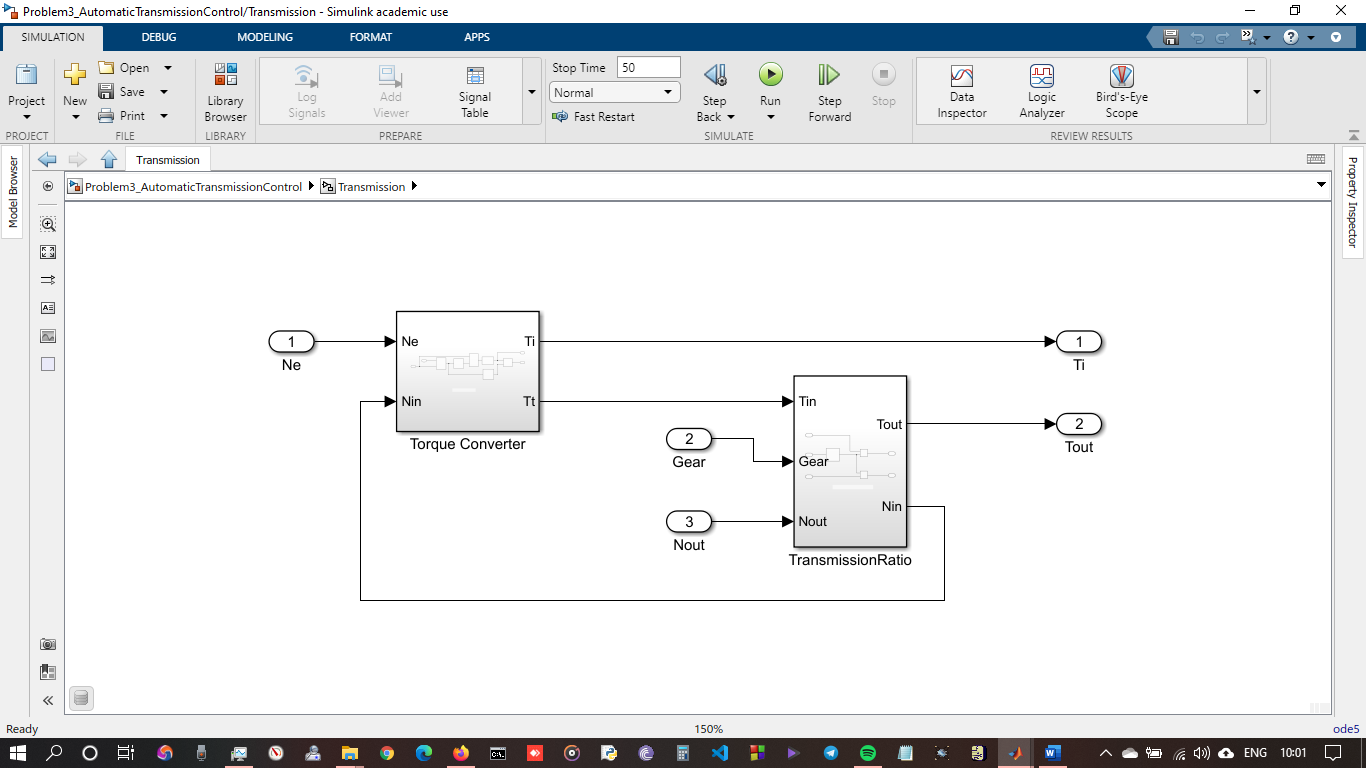


Figure 9 Transmissio System

**Torque Converter:**

The subsystem requires a vector of speed ratios ( Nin/Ne ) and vectors of K-factor (f2) and torque ratio (f3) for the lookup tables.

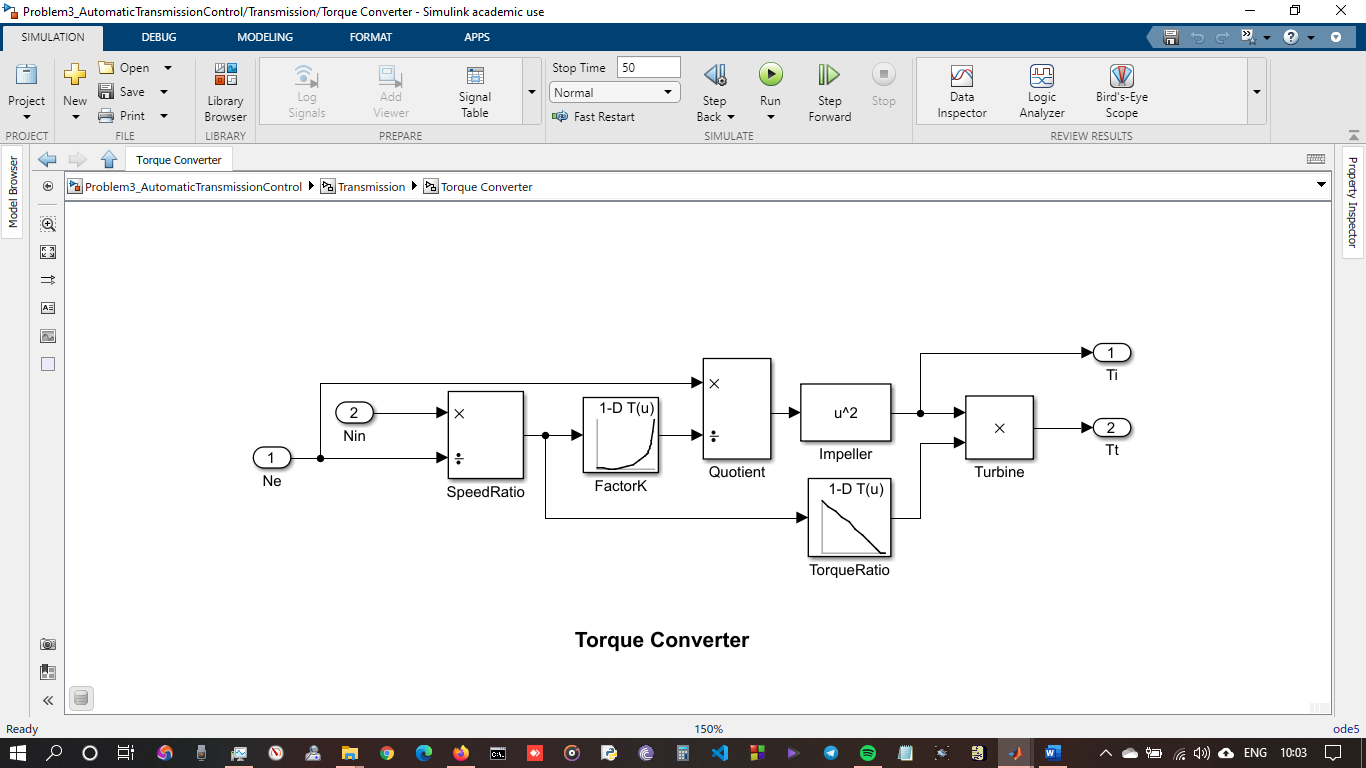


Figure 10 Torque Converter

**Transmission Ratio:**

The transmission ratio block determines the ratio computes the transmission output torque and input speed, as indicated in Equation 3.

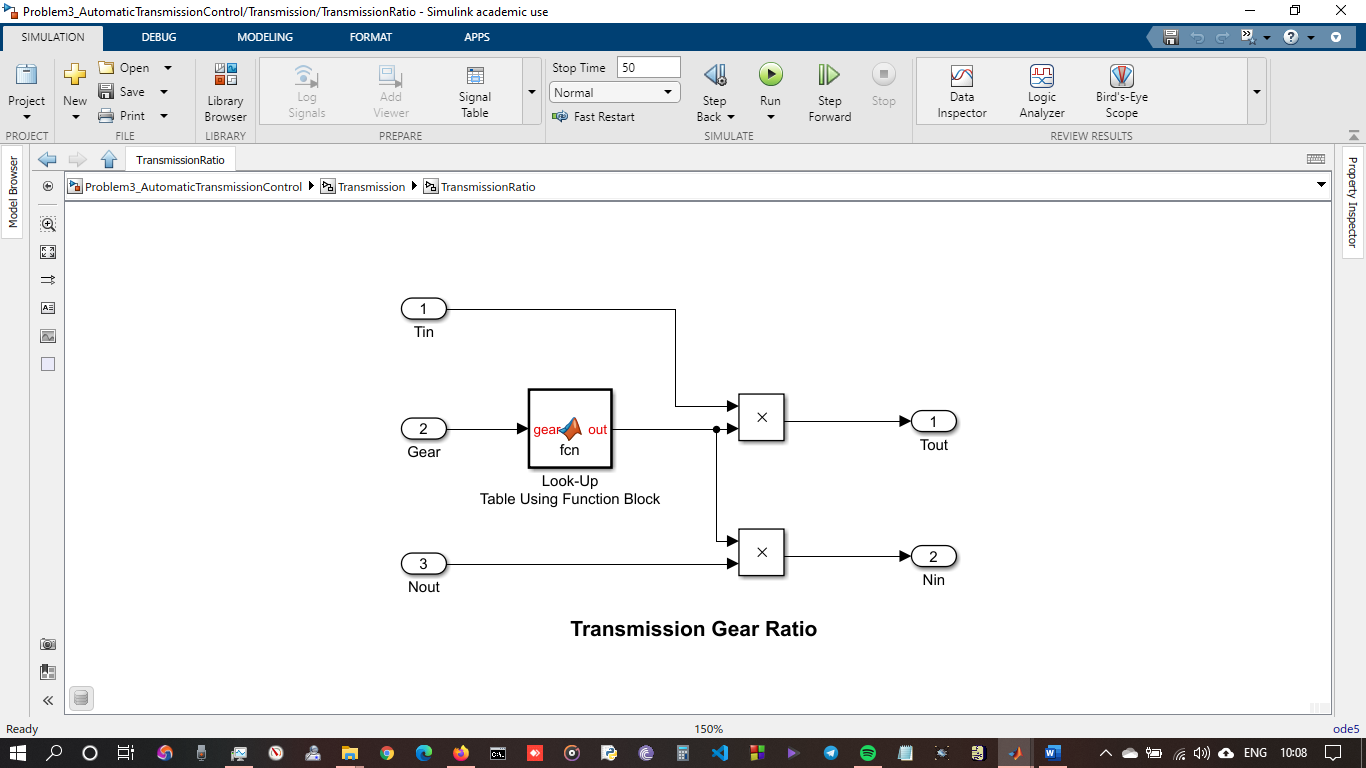


Figure 11 Transmission ratio

**Shift Logic Chart:**

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.

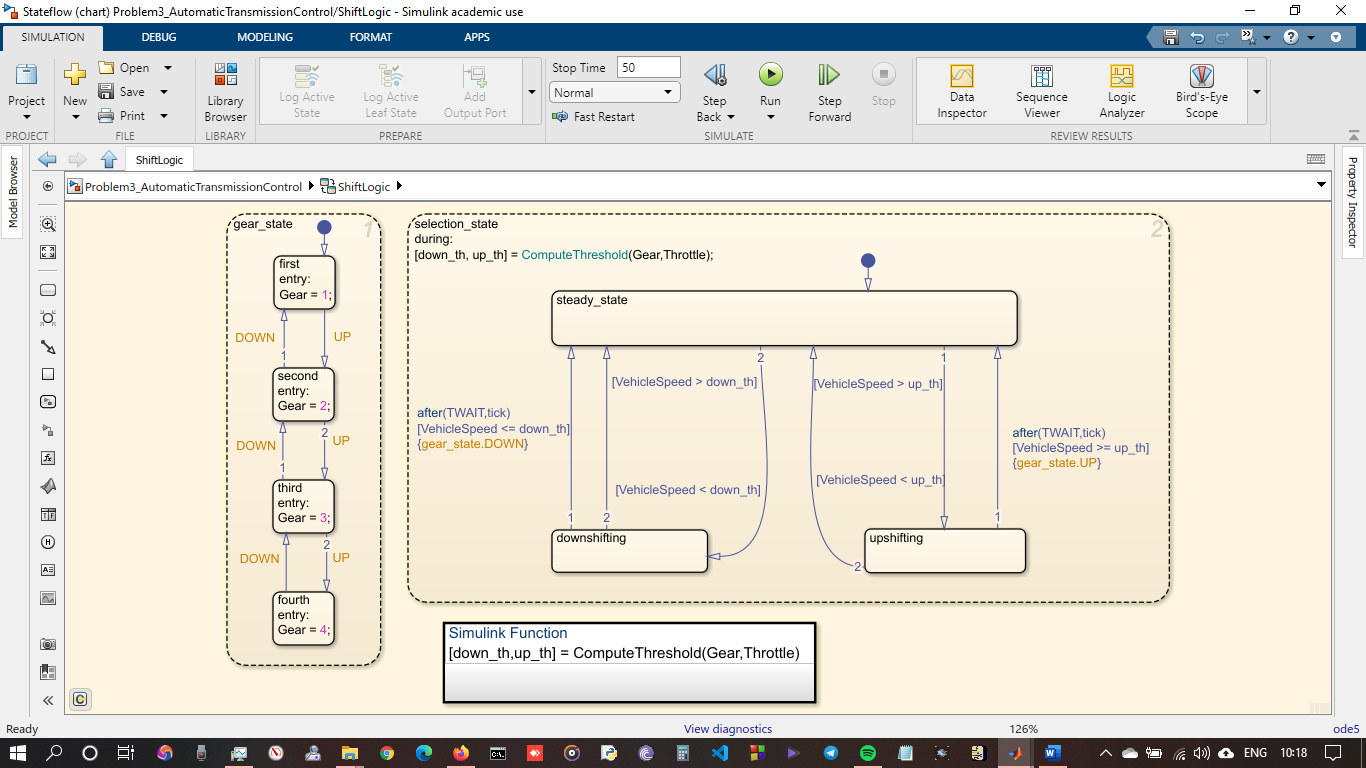


Figure 12 Shift logic Stateflow Diagram

**Vehicle:**

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 parameters entered are the final drive ratio, the polynomial coefficients for drag friction and aerodynamic drag, the wheel radius, vehicle inertia, and initial transmission output speed.

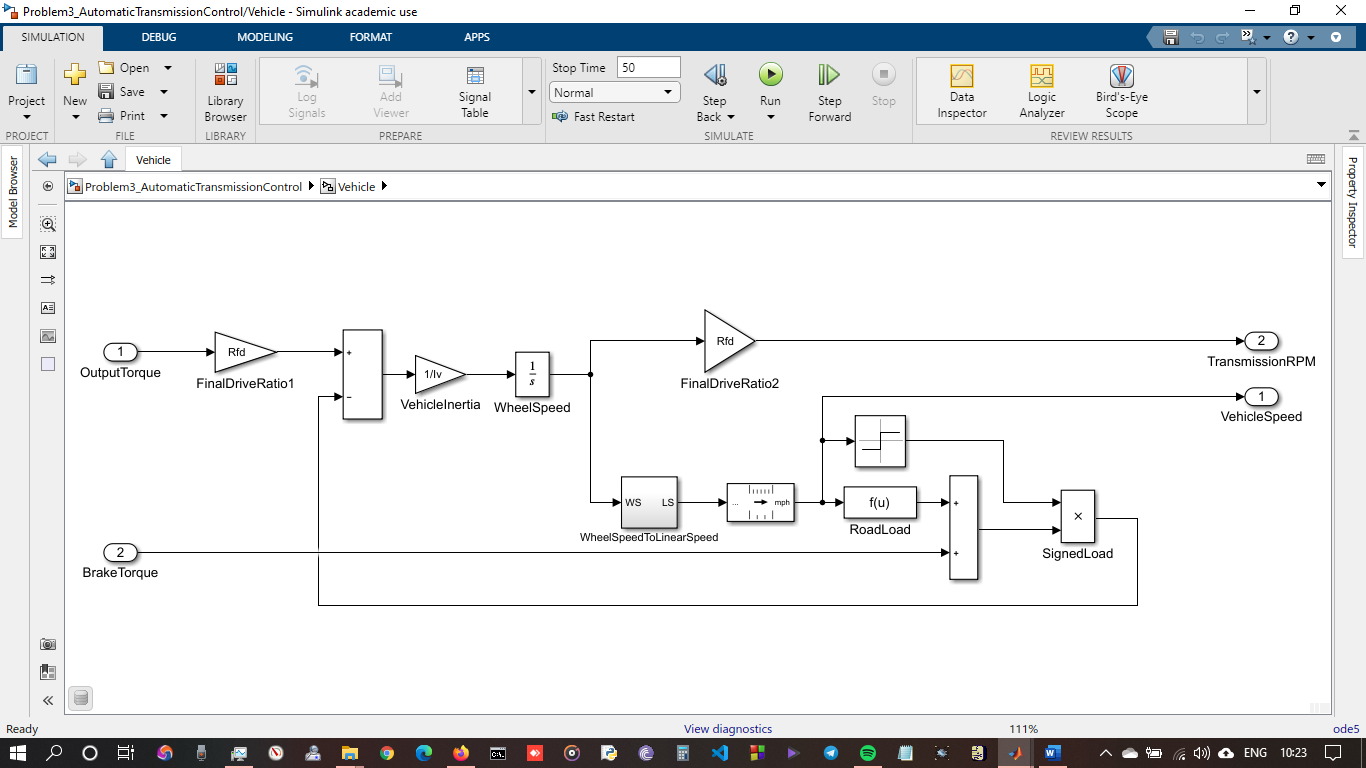


Figure 13 Vehicle Subsystem

**SOLVER SELECTION**

The appropriate solver for simulating a model depends on these characteristics:

* System dynamics
* Solution stability
* Computation speed
* Solver robustness

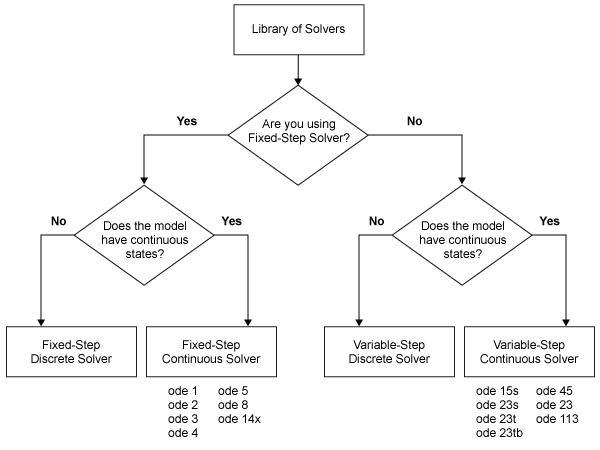


Figure 14 Solver selection flow chart

According to the above figure, this system does not contain any discrete blocks, so **continuous solvers** need to be selected.

Fixed Step Solver is used because it took **least number** of steps to simulate compared to variable step solvers. See the blow figure 15.

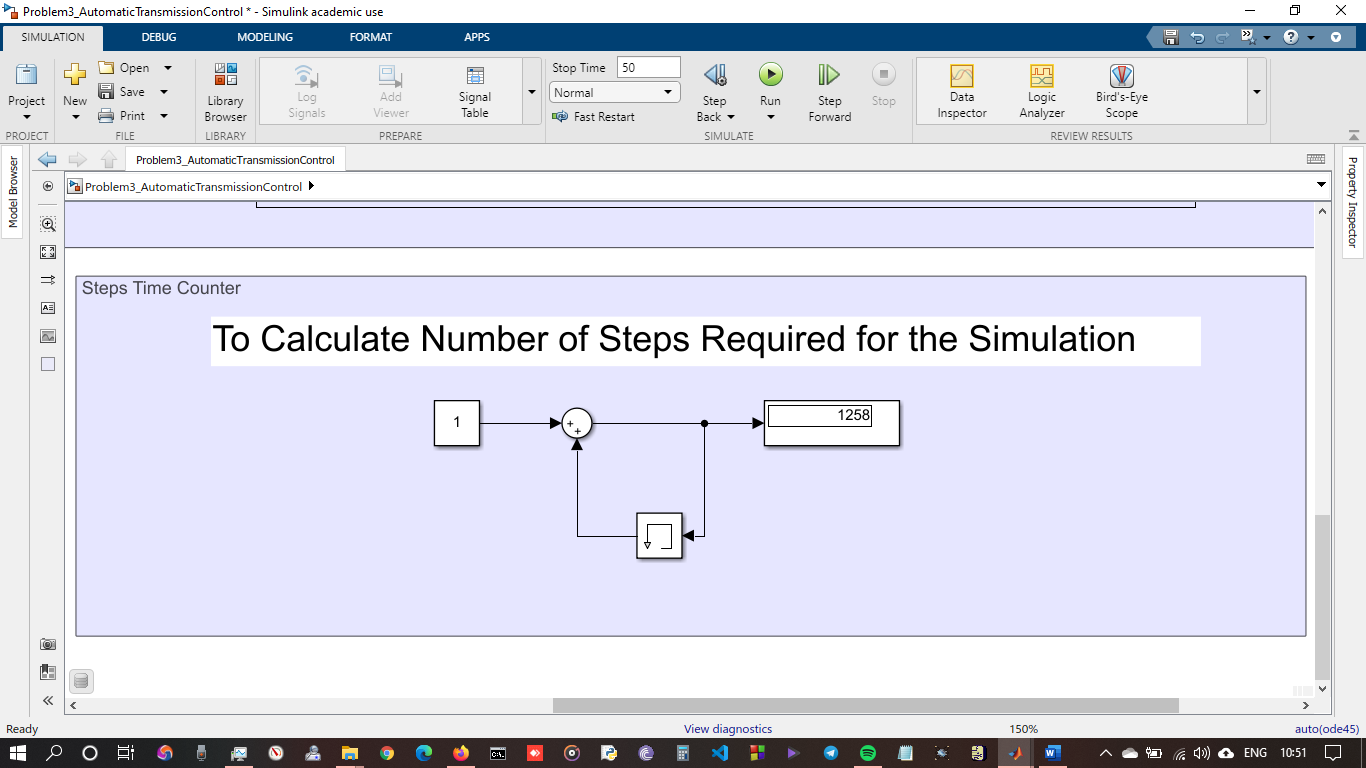


Figure 15 Steps count for Variable step Solvers

Variable step solvers took **1258 steps** to completed the simulation. But Fixed step solvers took only **1251 steps** (see figure 16) to complete the simulation. These step difference does not make any difference in the stability. So**, Fixed step solvers are selected**.

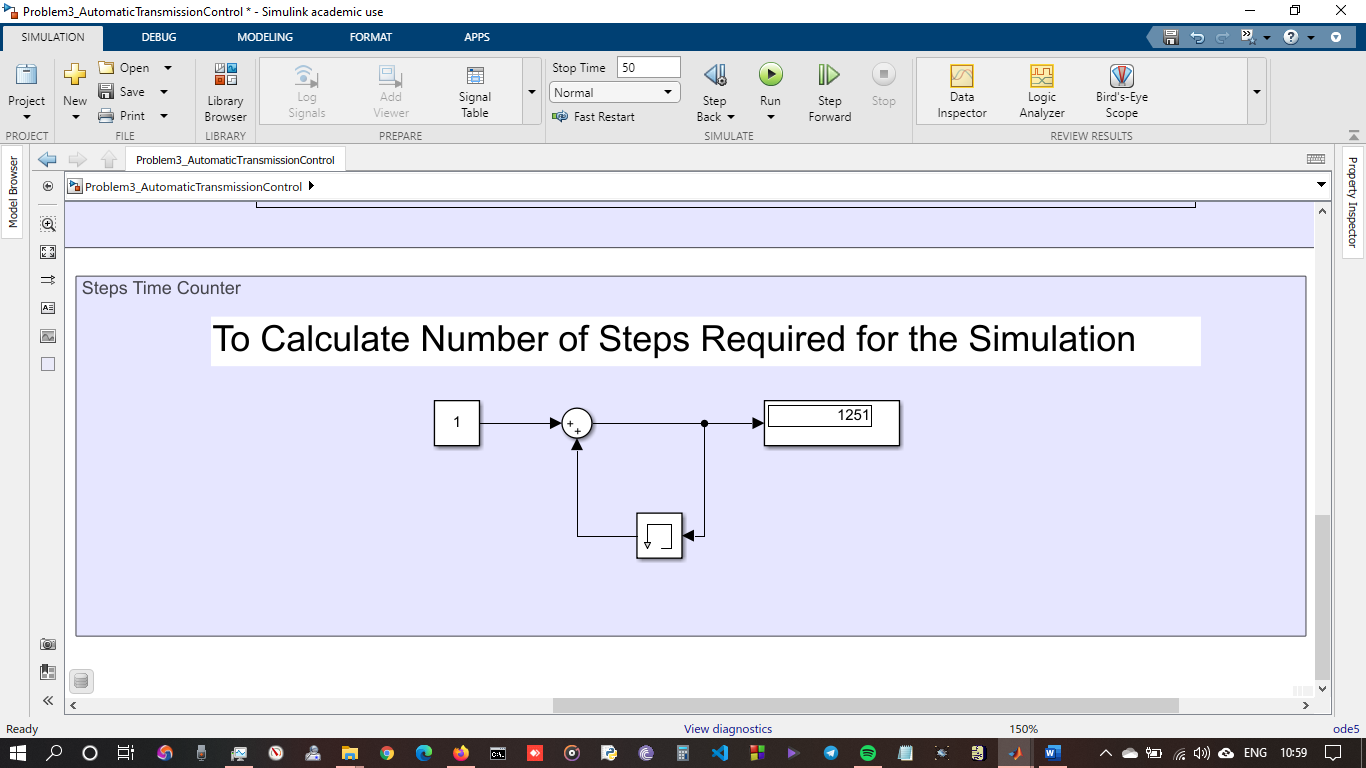


Figure 16 Steps count for Fixed Step Solvers

Fixed step size of 0.04 is taken because **0.04 sample time** is chosen for the shift logic chart. So, in order to simulate the system step size should be either 0.04 or in multiples of 0.04 which is lesser than 0.04. Since, taking the step size below 0.04 would increase simulation time. So, taking **0.04 step size is optimal.**

For selecting the solvers between stiff/non-stiff solvers, system dynamics is observed. Since there is will not be any **large changes in output for the smaller step size**, there is no need to select **Stiff solvers/Explicit solvers**. So, non-stiff solvers/Implicit solvers is used.

For selecting the solvers in non-stiff solvers, simulation elapsed time is used to compare between the solvers. **CPU time elapsed** for the solvers can be observed in the below table.

For Displaying the CPU elapsed time, System call back function is used. (See the figure 17)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sno. | Solver | Implicit/  Explicit | Fixed/  Variable Step Size | Step  Size | Step  Counts | Simulation  Time  (in Seconds) |
| 1 | Ode5 | Explicit | Fixed | 0.04 | 1251 | 1.1875 |
| 2 | Ode4 | Explicit | Fixed | 0.04 | 1251 | 0.8594 |
| 3 | Ode3 | Explicit | Fixed | 0.04 | 1251 | 0.9219 |
| 4 | Ode1 | Explicit | Fixed | 0.04 | 1251 | 1.1406 |
| 5 | Ode45 | Explicit | Variable | auto | 1258 | 0.8281 |
| 6 | Ode113 | Explicit | Variable | auto | 1441 | 0.8281 |
| 7 | Ode15s | Implicit | Variable | auto | 2938 | 0.7969 |
| 8 | Ode23t | Implicit | Variable | auto | 1316 | 1.0156 |
| 9 | OdeN | Implicit | Variable | auto | 1252 | 0.8438 |

Table 1 Experiment for Solvers

From the Experiment for solvers table, it can be observed that as predicted Fixed step Explicit solvers used lesser number of simulation step count. Since all the solvers works similar to this system. It optimal to take the solvers which takes lesser number of steps for simulation.

Simulation time also depends on the current state of the computer. Any background process also affects the simulation and simulation time. Since, all fixed step solvers simulate within certain range, any explicit solvers from fixed step can be chosen. For this model Ode5 Dormand-Prince