MODELLING OF A DC MOTOR USING SIMULINK

Name: Madhav R

B. Tech in Mechanical Engineering

NIT Trichy

ABSTRACT:

DC motor acts as an energy conversion actuator that converts electrical energy (of source) into mechanical energy (for load). These motors are extensively applied for robotic manipulations, cutting tools, electrical tractions, etc. The torque-speed characteristics of DC motors are most compatible with most mechanical loads. In the present work, the transfer function and Modelling of a DC motor are performed by using generalized equations in MATLAB and Simulink.

It directly provides rotary motion and, coupled with wheels or drums and cables can provide translational motion to Electric Vehicles. The electric equivalent circuit of the armature and the free-body diagram of the rotor are shown in the following figure•

The input of the system is the voltage source (V) applied to the motor's armature, while the output is the rotational speed of the shaft. The rotor and shaft are assumed to be rigid.

INDEX:

1. INTRODUCTION TO SIMULINK:

Simulink is a simulation and model-based design environment for dynamic and embedded systems integrated with MATLAB. Simulink, also developed by MathWorks, is a data flow graphical programming language tool for modelling, simulating, and analysing multi-domain dynamic systems. It is basically a graphic block diagramming tool with a customizable block libraries.

It allows you to incorporate MATLAB algorithms into models as well as export the simulation results into MATLAB for further analysis.

Simulink supports −

- System-level design
- Simulation
- Automatic code generation
- Testing and verification of embedded systems

There are several other add-on products provided by MathWorks and third-party hardware and software products available for use with Simulink.

The following list gives a brief description of some of them −

- Stateflow allows developing state machines and flow charts.
- Simulink Coder allows the generation of C source code for real-time implementation of systems automatically.
- xPC Target, together with x86-based real-time systems, provide an environment to simulate and test Simulink and Stateflow models in real-time on the physical system.
- Embedded Coder supports specific embedded targets.
- HDL Coder allows to automatically generate synthesizable VHDL and Verilog.
- SimEvents provides a library of graphical building blocks for modelling queuing systems.

Simulink is capable of systematic verification and validation of models through modelling style checking, requirements traceability and model coverage analysis.

Simulink Design Verifier allows you to identify design errors and generate test case scenarios for model checking.

2. MATHEMATICAL MODELLING FOR A DC MOTOR:

In general, the torque generated by a DC motor is proportional to armature current and the strength of the magnetic field. Assuming magnetic field to be constant, we get

Torque = $K_i * i$

Where, K_i is the constant of proportionality. Similarly, the back emf is proportional to angular velocity. Hence,

$e = K_e * \omega$

Now, based on KVL and Newton's 2nd law,

$(J * \alpha) + (b * \omega) = (K * i)$

$[L * d(i)/dt] + [R * i] = V - (K * \omega)$

3. MODEL OF DC MOTOR USING MATLAB AND SIMULINK:

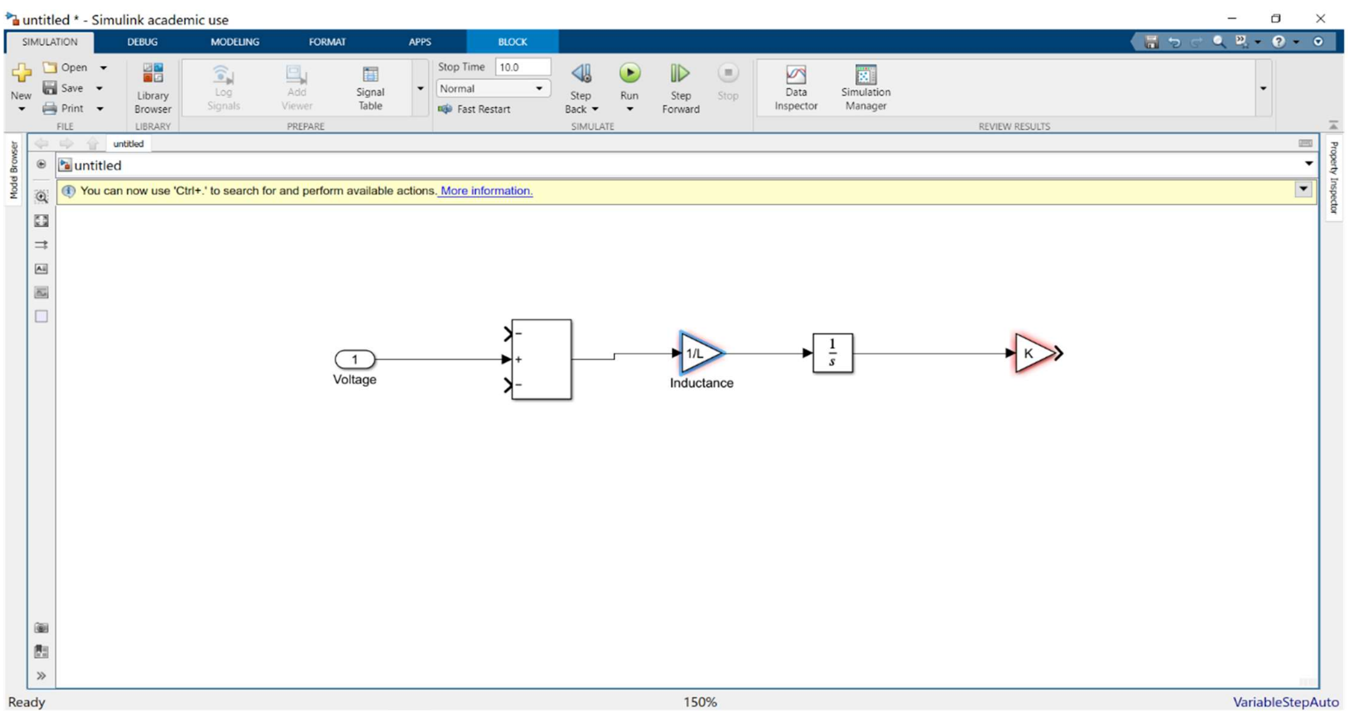

Figure 1: Calculation of the equation obtained from Kirchhoff's law

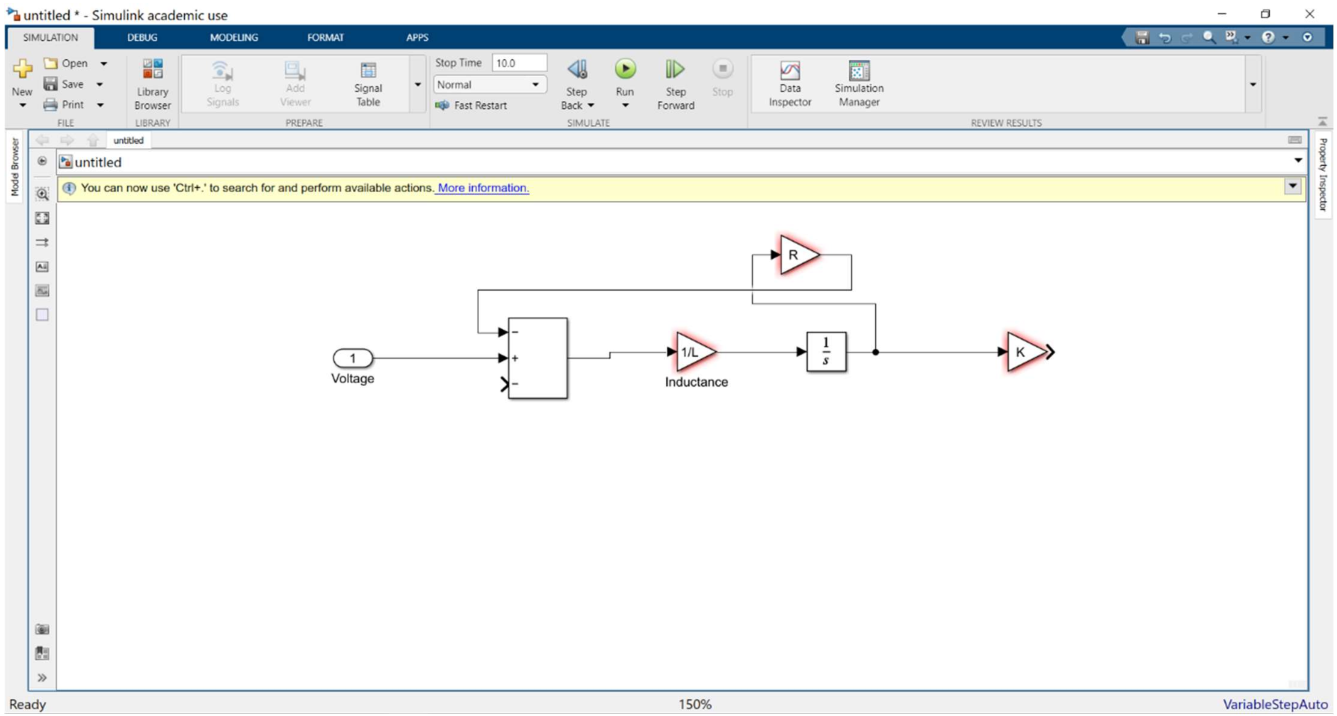

Figure 2: Connecting the blocks to form a closed loop diagram to form the second equation.

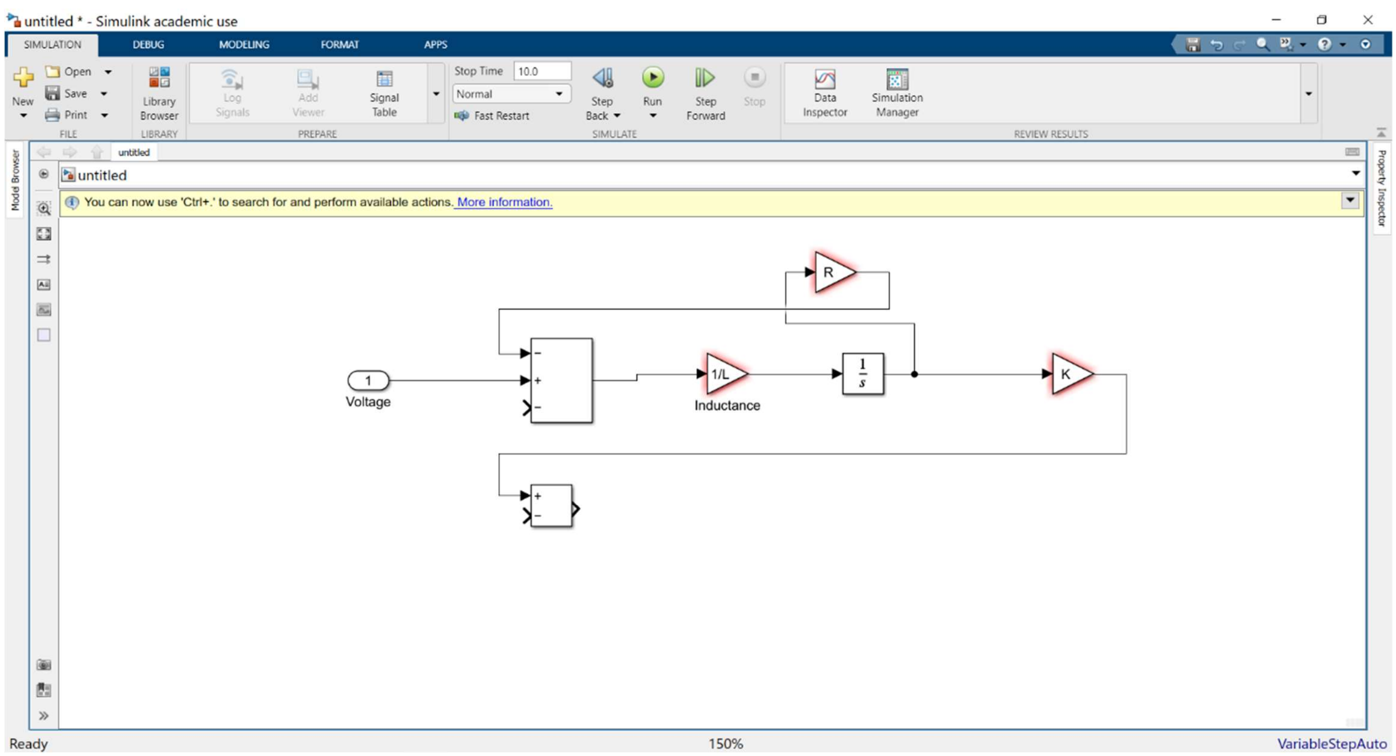

Figure 3: Taking the output of this block and feeding it to the secondary system.

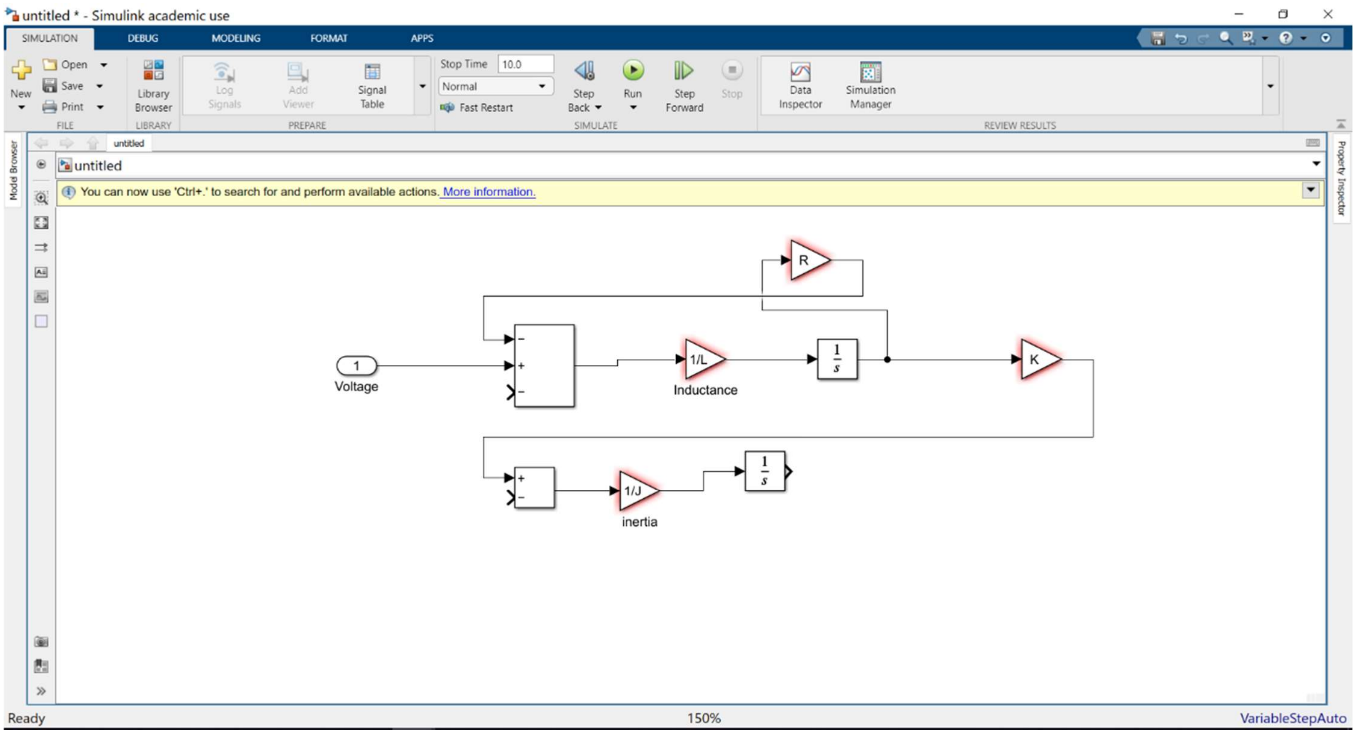

Figure 4: Forming the second torque equation.

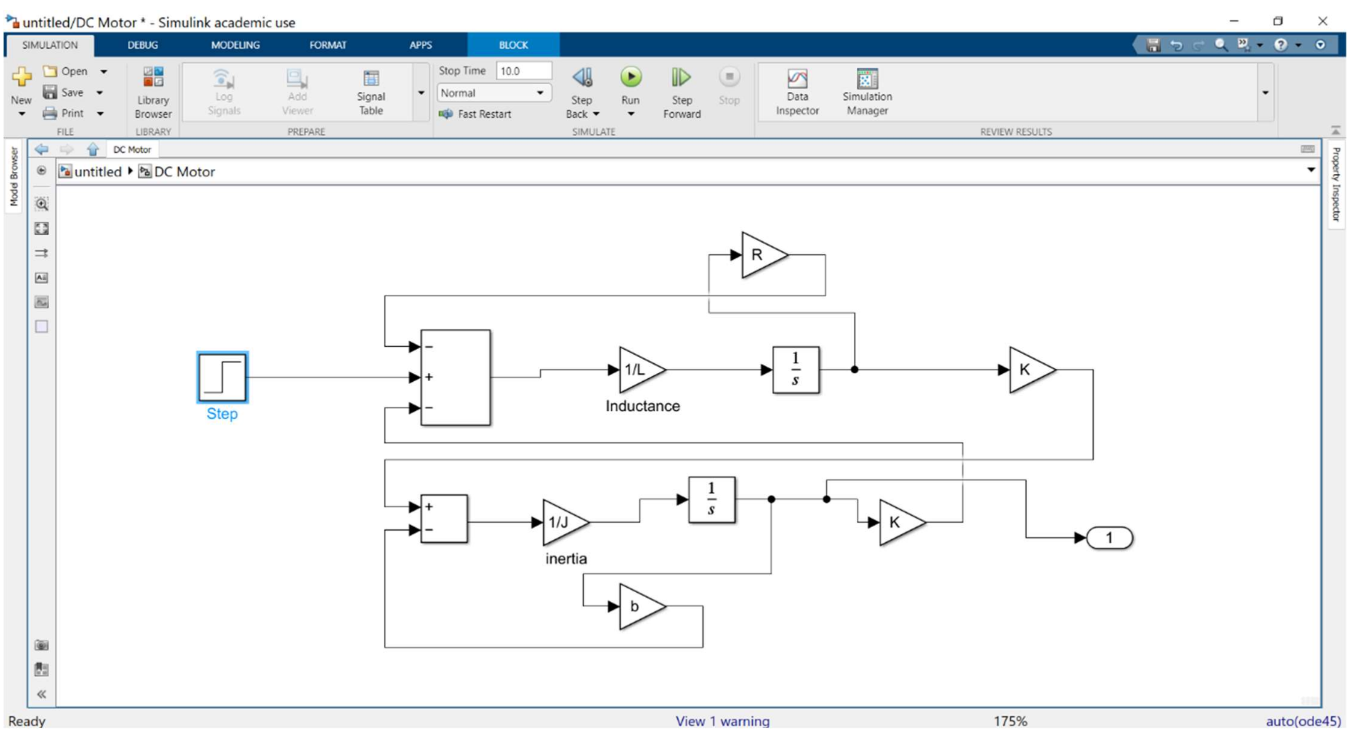

Figure 5: Completing the torque equation by forming the appropriate closed loop.

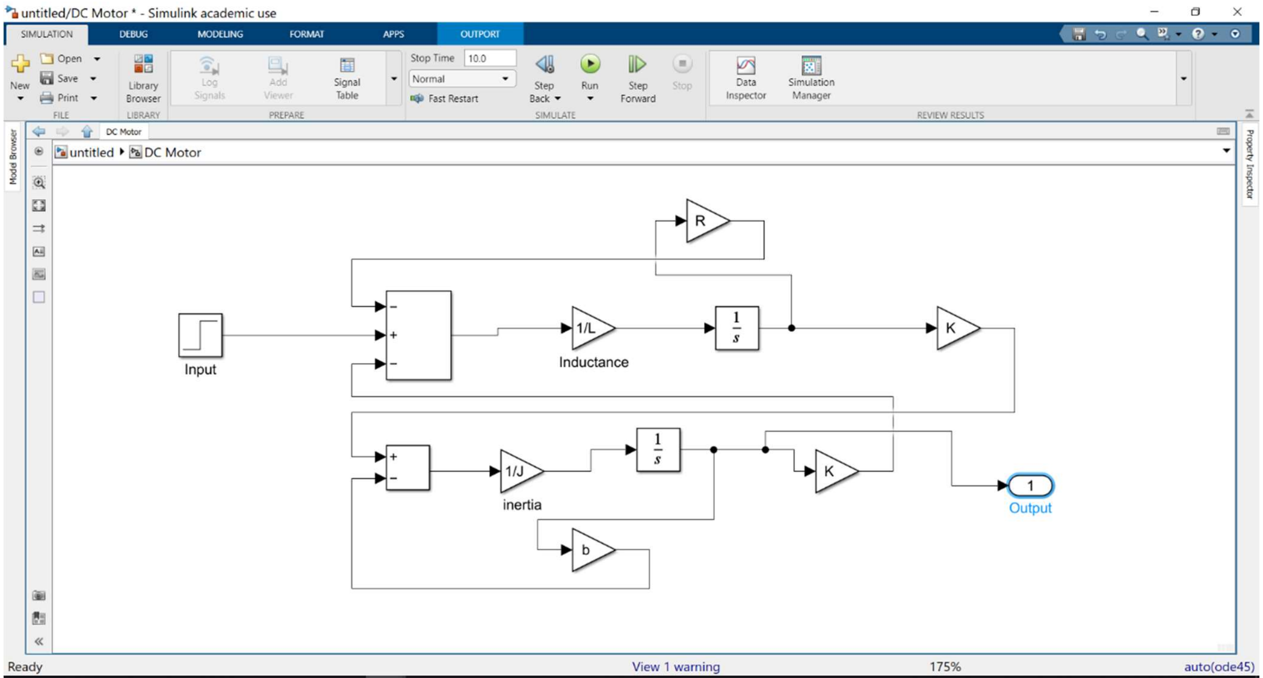

Figure 6: Rebranding the input and the output blocks.

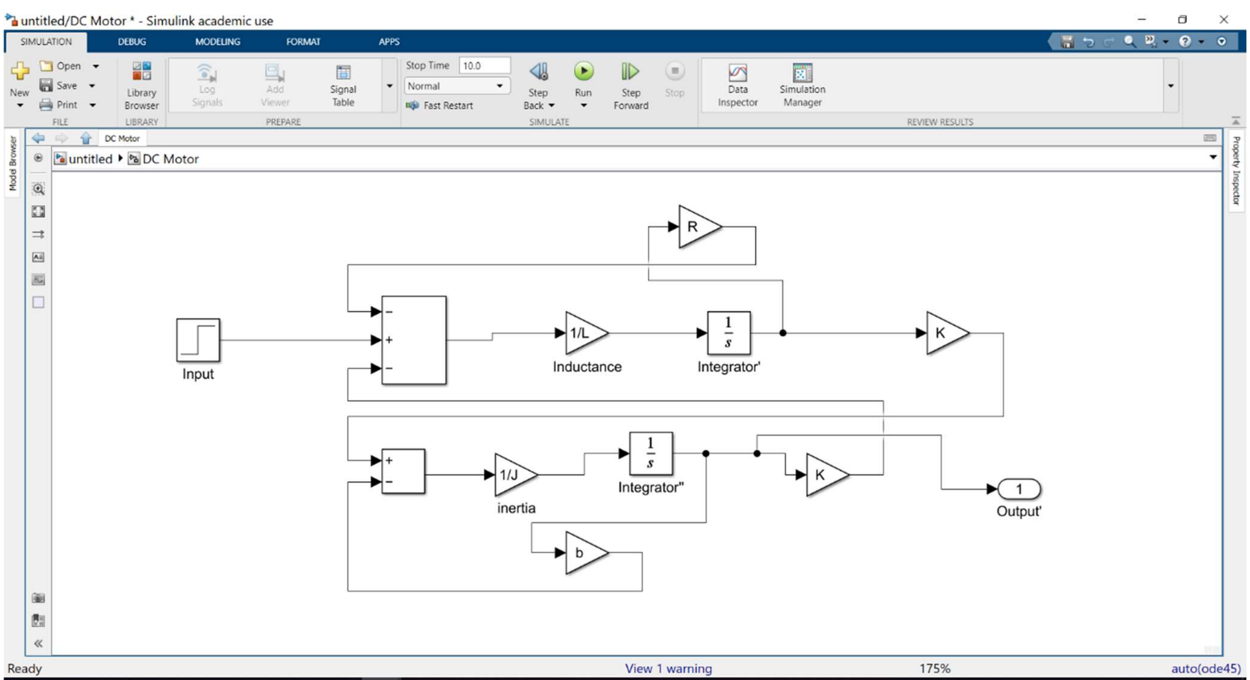

Figure 7: Rebranding the variables.

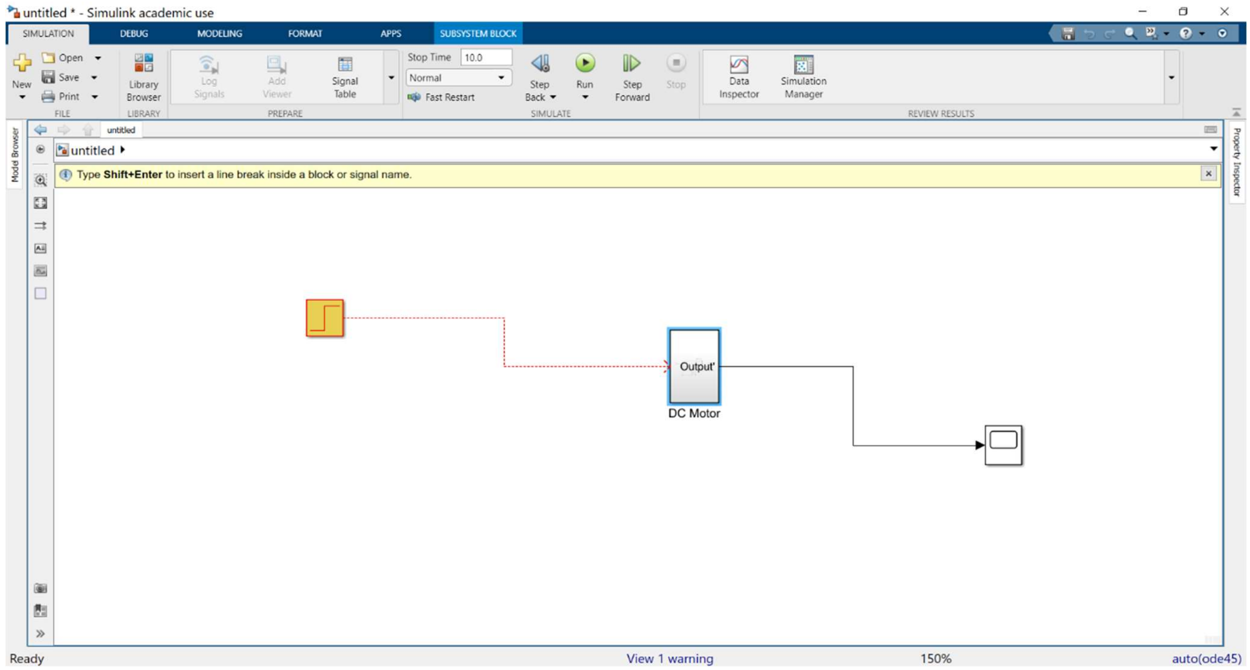

Figure 8: Representing the entire system as a system and its corresponding sub systems.

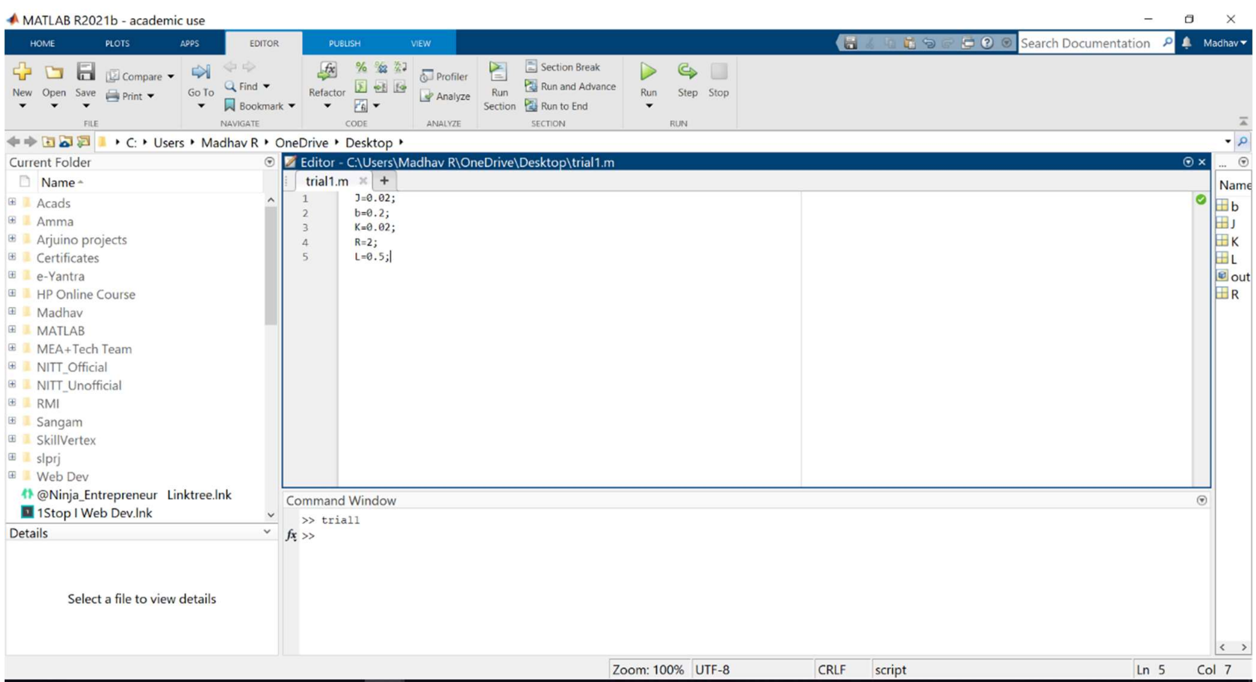

Figure 9: Assignment of the variables used for Simulink in MATLAB.

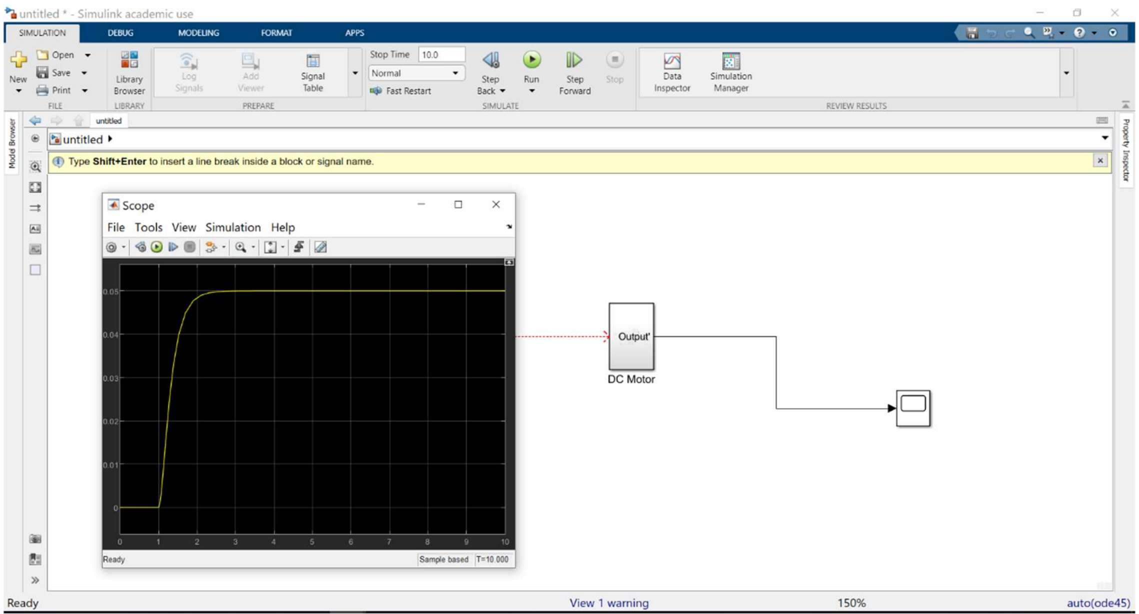

Figure 10: OUTPUT graph that represents the rotational speed of the shaft.

5. RESULTS AND DISCUSSION:

The simulation of a practical DC motor is done in MATLAB successfully.