MODEL BASED DESIGN OF BLDC MOTOR CONTROL

SUBMITTED BY:
PRIYA PAUL
MANI RAJESH CHADUVULA
NADEEM NISSAR

DEGREE OF MASTER OF SCIENCE'S
IN ADVANCED AUTOMOTIVE ELECTRONICS ENGINEERING
(MUNER)

Table of Contents

List of Figures

1 ABSTRACT

In this report we detail and outline the steps and procedures carried out in modelling and simulating the BLDC motor control and the achieved oututs.

2 INTRODUCTION

Brushless Direct Current (BLDC) motors are one of the motor types rapidly gaining popularity. As the name implies, BLDC motors do not use brushes for commutation; instead, they are electronically commutated. BLDC motors have many advantages over brushed DC motors and induction motors.
A few of these are:
• Better speed versus torque characteristics
• High dynamic response
• High efficiency
• Long operating life
• Noiseless operation
• Higher speed ranges
In addition, the ratio of torque delivered to the size of the motor is higher, making it useful in applications where space and weight are critical factors.

3 HARDWARE

The hardware used for this project are very easily available equipment like a Windows PC and an Arduino UNO. We did code generation and PIL testing using Arduino Uno and Matlab.

4 SOFTWARE

The software used for the realisation of the project.

1)MATLAB
2)SIMULINK

Tools and software used to enable communication and documentation throughout the project.

1)Skype
2)Latex
3)GitHub
4)Draw-IO

5 MODEL-BASED DESIGN

For the execution of our project we followed the V-model which is a derivative of waterfall model, since this is the most widely used model for the approach of model-based design.

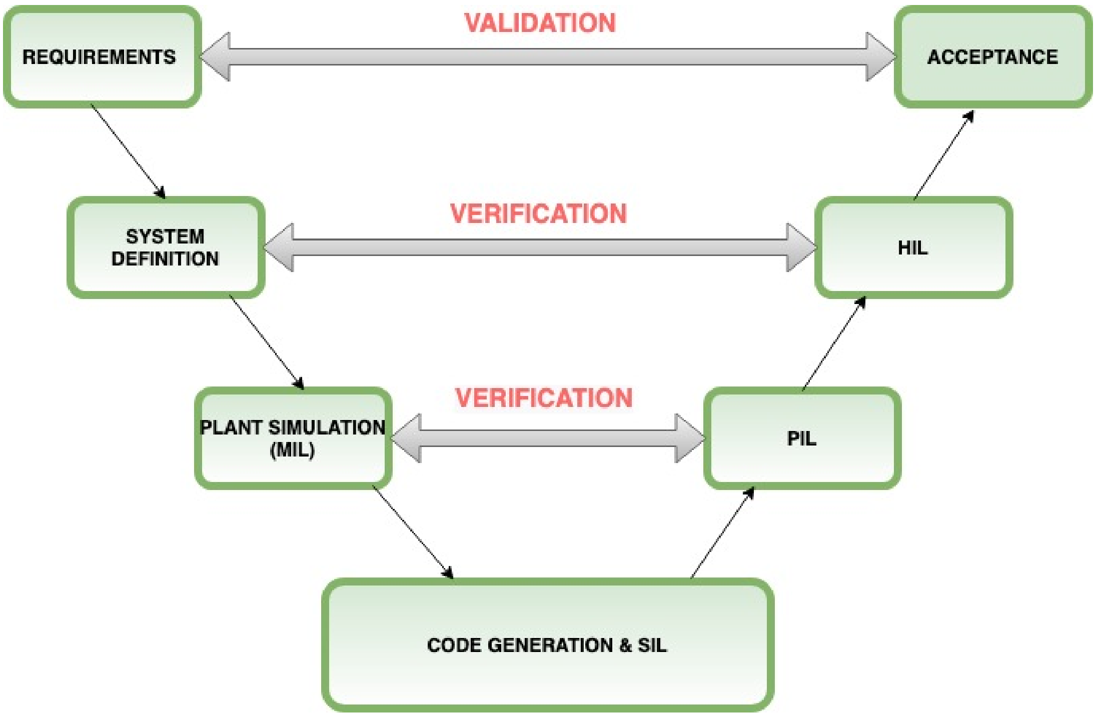

Figure 1: V-Model

Description

- **REQUIREMENTS**:
  We use the BLDC motor for the power window application within a car. The clockwise and anticlockwise rotation of the motor is to be controlled. The speed of the motor is a constant.

- **SYSTEM DEFINITION**:
  In this step we defined all the system parameters on the top level.

  1. Motor parameters and specification
  2. Controller specification
  3. Sensing element

  To name a few.

- **MIL:**
  The plant model was verified by giving different test inputs. The following test vector was given as input to the system to test the speed. [100, 300, 600,

900]RPM.

- **Automatic code generation**:
  We use two digital input pins to read the state of the switches to which they are connected and a PWM pin to generate the PWM output required to control the BLDC motor. The first switch is for on and off of the motor and the second toggle switch act as the control for the up and down movement of the windows. The pwm duty cycle is set to be a constant as the application that we are focusing on is power windows within a vehicle. After the Simulink model of the controller is completed, the c code can be generated by setting the target for the embedded c code generator to be Arduino Uno board.

- **PIL:** With the code generated we were able to run this on the Arduino Uno board at out disposal but since we could not read the output PWM we could not complete the loop.But we propose a method to read the PWM output in the later part of this document.

- **HIL:** In this step the plant should be simulated in a HIL module while the motor control should run on target hardware but since we didn't not have the hardware to perform this step we could not carry out HIL.

- **Acceptance:** The real plant is realised, integration tests are performed but again we did not have the possibility to perform this step.

6 PLANT

6.1 MOTOR

Brushless Direct Current (BLDC) motors are one of the motor types rapidly gaining popularity. As the name implies, BLDC motors do not use brushes for commutation; instead, they are electronically commutated. BLDC motors have many advantages over brushed DC motors and induction motors. A few of these are:

- Better speed versus torque characteristics
- High dynamic response
- High efficiency
- Long operating life
- Noiseless operation
- Higher speed ranges

In addition, the ratio of torque delivered to the size of the motor is higher, making it useful in applications where space and weight are critical factors.

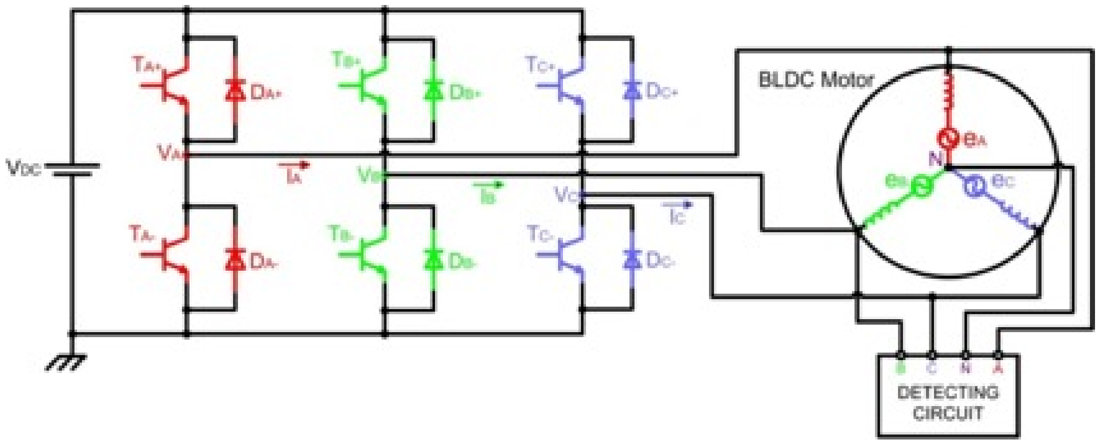

Figure 2: STATOR AND ROTOR

6.2 OPERATION

To better understand BLDC motor behaviour when an external voltage is applied, we will use the configuration shown earlier where the rotor consists of a single pole pair while the stator is made up of three coils spaced at 120 degrees. The coils, which are here referred to as phases A, B, and C, can be energized by passing a current through them. In the beginning, none of the coils are energized and the rotor is stationary. Applying voltage across two phases, A and C, generates a combined magnetic field along the dashed line. As a result, the rotor starts to turn to align itself with the stator magnetic field.

There are six possible ways of energizing coil pairs. By commutating two phases at a time, the resulting stator magnetic field is rotated, which causes the rotor to turn and end up in the positions shown as below. The rotor angle is measured with respect to the horizontal axis and there are six different rotor alignments, each 60 degrees apart from each other. This means that if the correct phases are commutated every 60 degrees, the motor will continuously spin. The term for this type of control is six-step commutation or trapezoidal control.

6.3 CONSTRUCTION OF MOTOR

STATOR AND ROTOR The stator has a coil arrangement placed in the slots that are axially cut along the inner side as shown in the fig below .The windings are connected in a star manner, each of these windings are constructed using numerous coils interconnected to form a windings, which are separated to form an even number of poles.

The type of stator windings variants in BLDC motor is trapezoidal. This differenti-

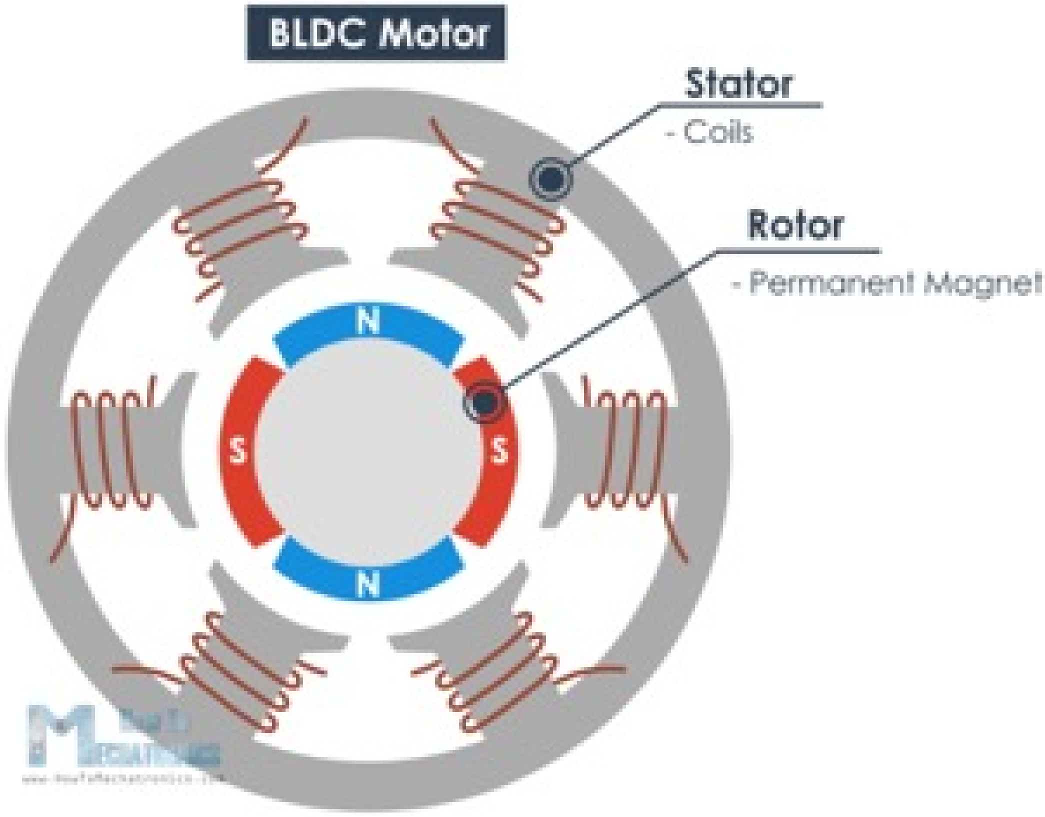

Figure 3: STATOR AND ROTOR

ation is made on the basis of the interconnection of coils in the stator windings to give the different types of back Electromotive Force (EMF). As the names indicates, the trapezoidal motor gives a back EMF in trapezoidal fashion.
The rotor is made of permanent magnet and can vary from two to eight pole pairs with alternate North (N) and South (S) poles. Based on the required magnetic field density in the rotor, the proper magnetic material is chosen to make the rotor.

From here we choose the appropriate specification of the motor to be modelled.

Specification of the MOTOR:

Number of pole pairs	4
Stator Resistance(ohm)	0.001
Stator Inductance(H)	0.00016
Magnetic flux linkage(wb)	0.18H
Speed(rpm)	1000
Torque(Nm)	0.082
Rated power(w)	500
Rotor Inertia(Kgm^2)	2

6.4 COMMUTATION

The commutation o the BLDC motor is controlled electronically. To rotate the motor the stator winding s should be energised in a sequence and is also important to know the position of the rotor to understand which winding will be energised. This rotor position is sensed using Hall effect Sensor which is embedded into the stator. Hall effect Sensor it is a device that is used to measure the magnitude of a magnetic field, where the output voltage is directly proportional to the magnetic field strength flow through it. Based on the physical position of the Hall sensors, there are two versions of output. The Hall sensors may be at 60° or 120° phase shift to each other. Based on this, we define the commutation sequence, which should be followed when controlling the motor.
Our sequence is represented in the commutation logic table.

Sequence #	Hall Sensor Input			Active PWMs		Phase Current		
	A	B	C			A	B	C
1	0	0	1	PWM1(Q1)	PWM4(Q4)	DC+	Off	DC-
2	0	0	0	PWM1(Q1)	PWM2(Q2)	DC+	DC-	Off
3	1	0	0	PWM5(Q5)	PWM2(Q2)	Off	DC-	DC+
4	1	1	0	PWM5(Q5)	PWM0(Q0)	DC-	Off	DC+
5	1	1	1	PWM3(Q3)	PWM0(Q0)	DC-	DC+	Off
6	0	1	1	PWM3(Q3)	PWM4(Q4)	Off	DC+	DC-

Figure 4: SEQUENCE FOR ROTATING THE MOTOR IN CLOCKWISE DIRECTION WHEN VIEWED FROM NON-DRIVING END

Sequence #	Hall Sensor Input			Active PWMs		Phase Current		
	A	B	C			A	B	C
1	0	1	1	PWM5(Q5)	PWM2(Q2)	Off	DC-	DC+
2	1	1	1	PWM1(Q1)	PWM2(Q2)	DC+	DC-	Off
3	1	1	0	PWM1(Q1)	PWM4(Q4)	DC+	Off	DC-
4	1	0	0	PWM3(Q3)	PWM4(Q4)	Off	DC+	DC-
5	0	0	0	PWM3(Q3)	PWM0(Q0)	DC-	DC+	Off
6	0	0	1	PWM5(Q5)	PWM0(Q0)	DC-	Off	DC+

Figure 5: SEQUENCE FOR ROTATING THE MOTOR IN COUNTER-CLOCKWISE DIRECTION WHEN VIEWED FROM NON-DRIVING END

7 CONTROLLER

In this scheme, current is controlled through motor terminals one pair at a time, with the third motor terminal always electrically disconnected from the source of power.

Three Hall devices embedded in the motor are usually used to provide digital signals which measure rotor position within 60 degree sectors and provide this information to the motor controller. Because at any time, the currents in two of the windings are equal in magnitude and the third is zero, this method can only produce current space vectors having one of six different directions. As the motor turns, the current to the motor terminals is electrically switched (commutated) every 60 degrees of rotation so that the current space vector is always within the nearest 30 degrees of the quadrature direction.

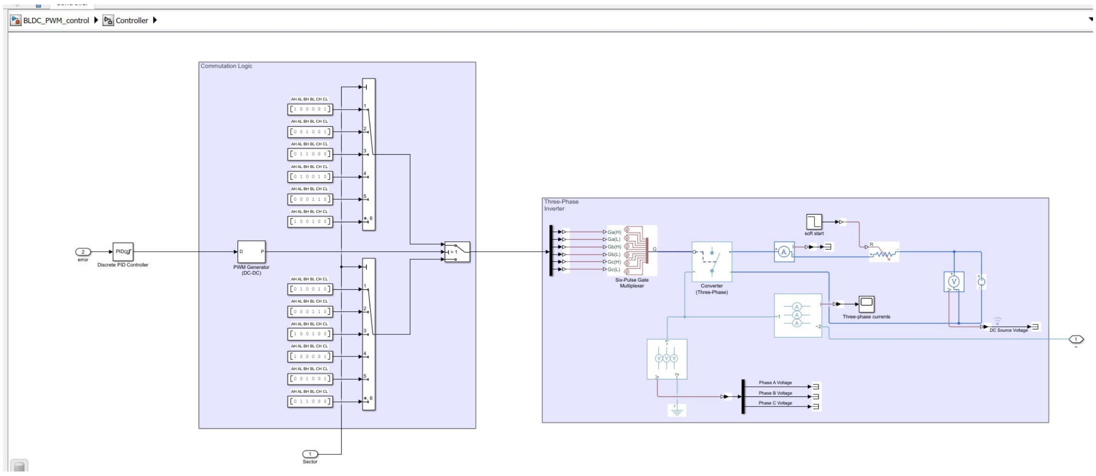

Figure 6: Controller Block

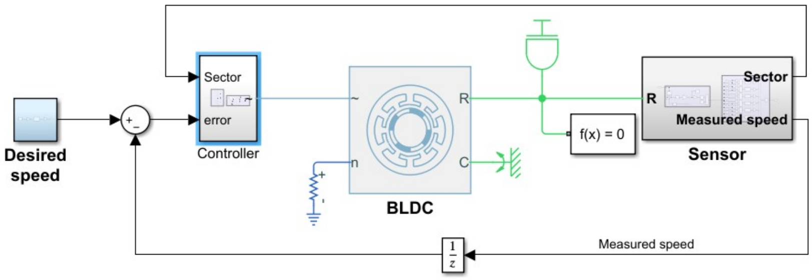

Figure 7: Simulink Block

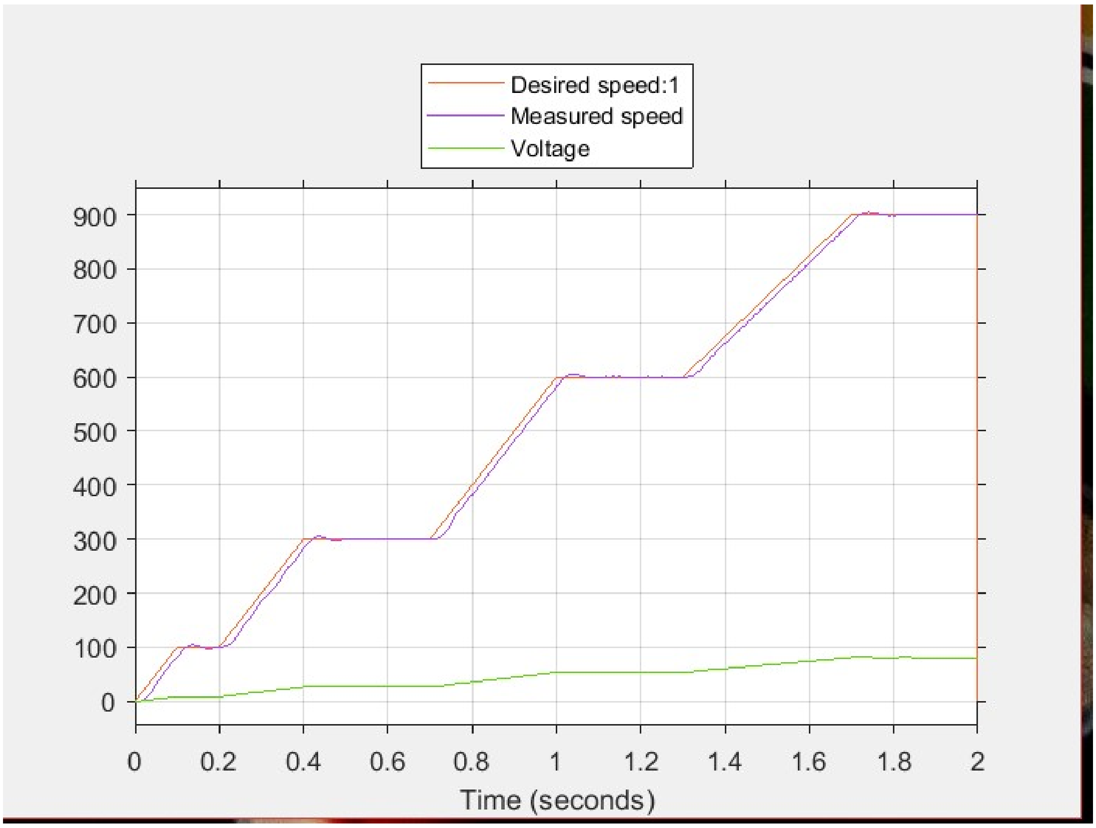

Figure 8: MIL OUTPUT.

7.1 MIL Conclusions:

After Applying the following test vector to our model in loop we have achieved the following results and here we can clearly see that the motor speed is closely following the desired speed.
Test vector in RPM: [100, 300, 600, 900]

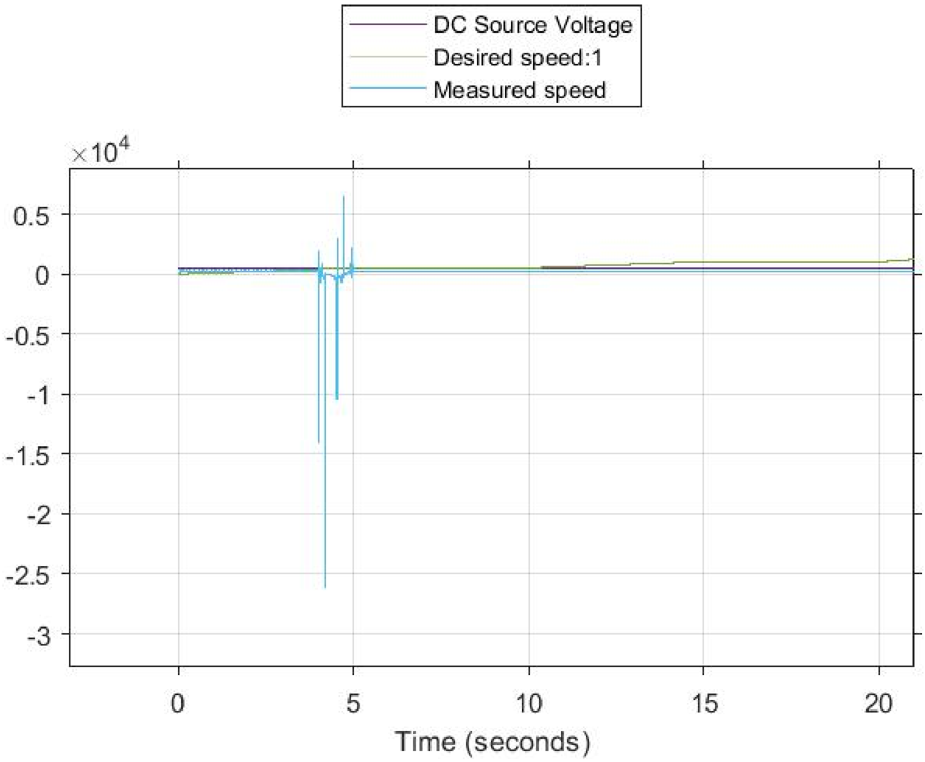

Figure 9: SIL OUTPUT

8 SOFTWARE IN LOOP.

For this step of the project we utilised the automatic code generator function of MATLAB and had it executed. We were returned with no errors which signifies that our SIL step is in accordance with the requirements.

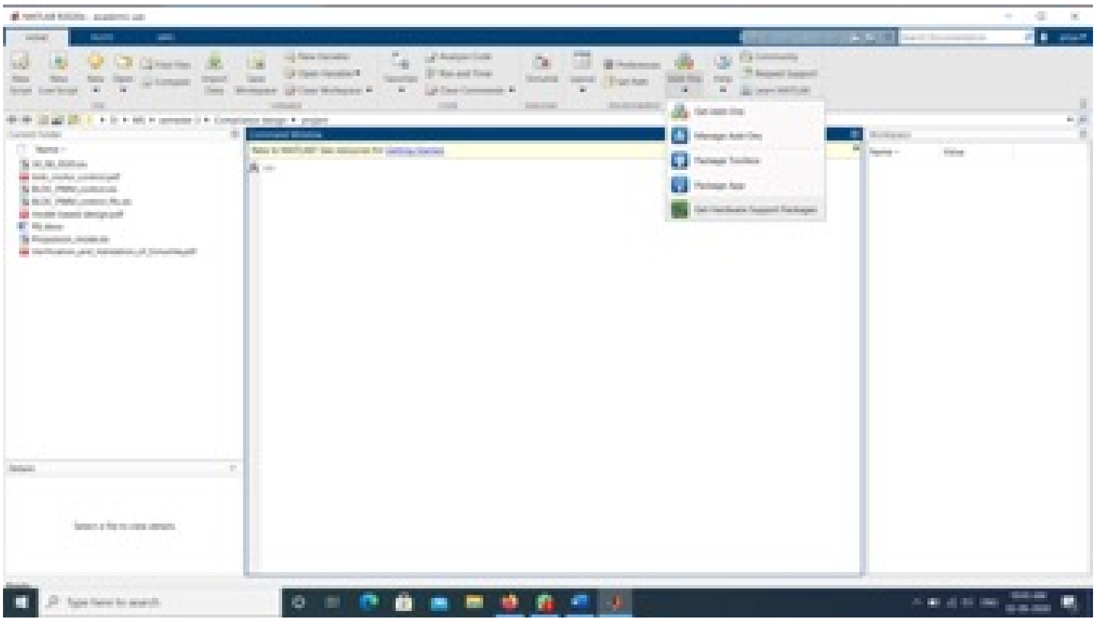

Figure 10: STEP 1.

9 CODE GENERATION AND PIL.

The last part of the project was focused on realising the PIL simulation, the goal being deploying the motor controller block to the hardware.
Inorder to generate code automatically and also to run the PIL test in Arduino, the Matlab and Simulink support package for Arduino were installed.

To install support package for Arduino hardware, follow the following steps

• On the MATLAB Home tab, in the Environment section, click Add-Ons > Get Hardware Support Packages. As shown in Figure 10.

• In the Add-On Explorer search bar, search for MATLAB Support Package for Arduino Hardware, and then click the MATLAB Support Package for Arduino Hardware add-on. As shown in Figure 11.

Similarly, also install the Simulink support package for Arduino. As shown in Figure 12.

After the support packages are installed, we can design the controller easily using the blocks available in the 'Simulink support package for Arduino' in the Simulink library. As shown in Figure 13.

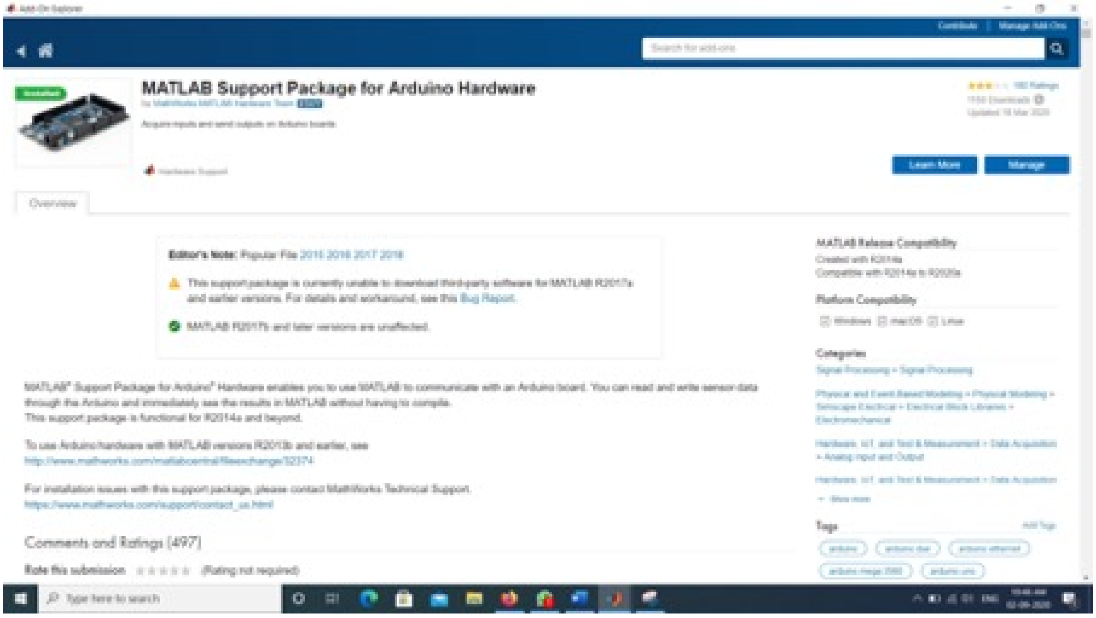

Figure 11: STEP 2.

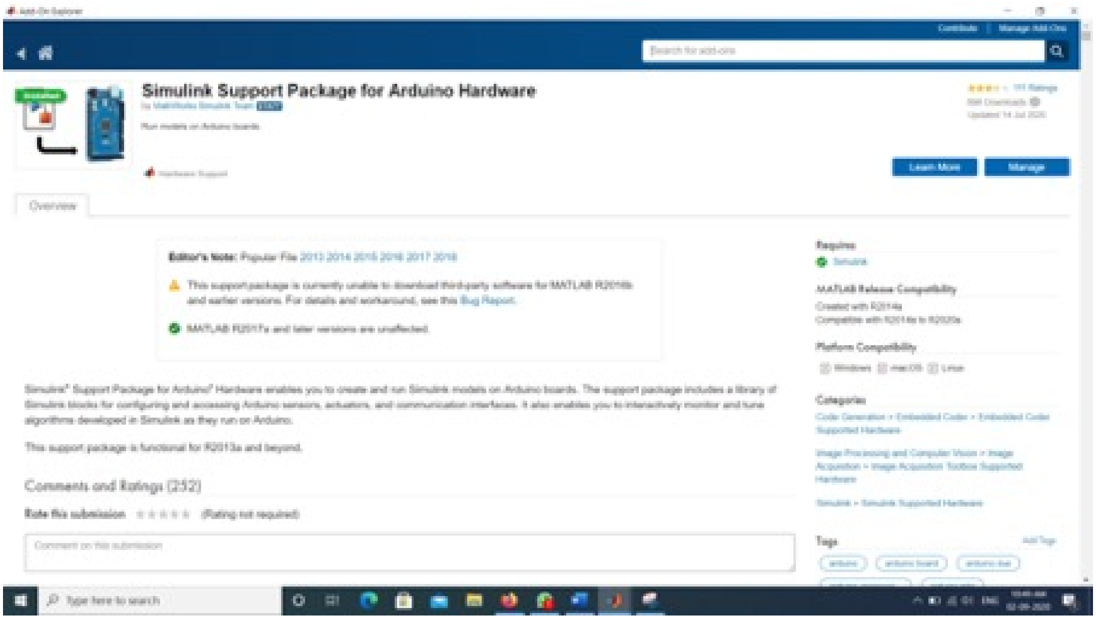

Figure 12: STEP 3.

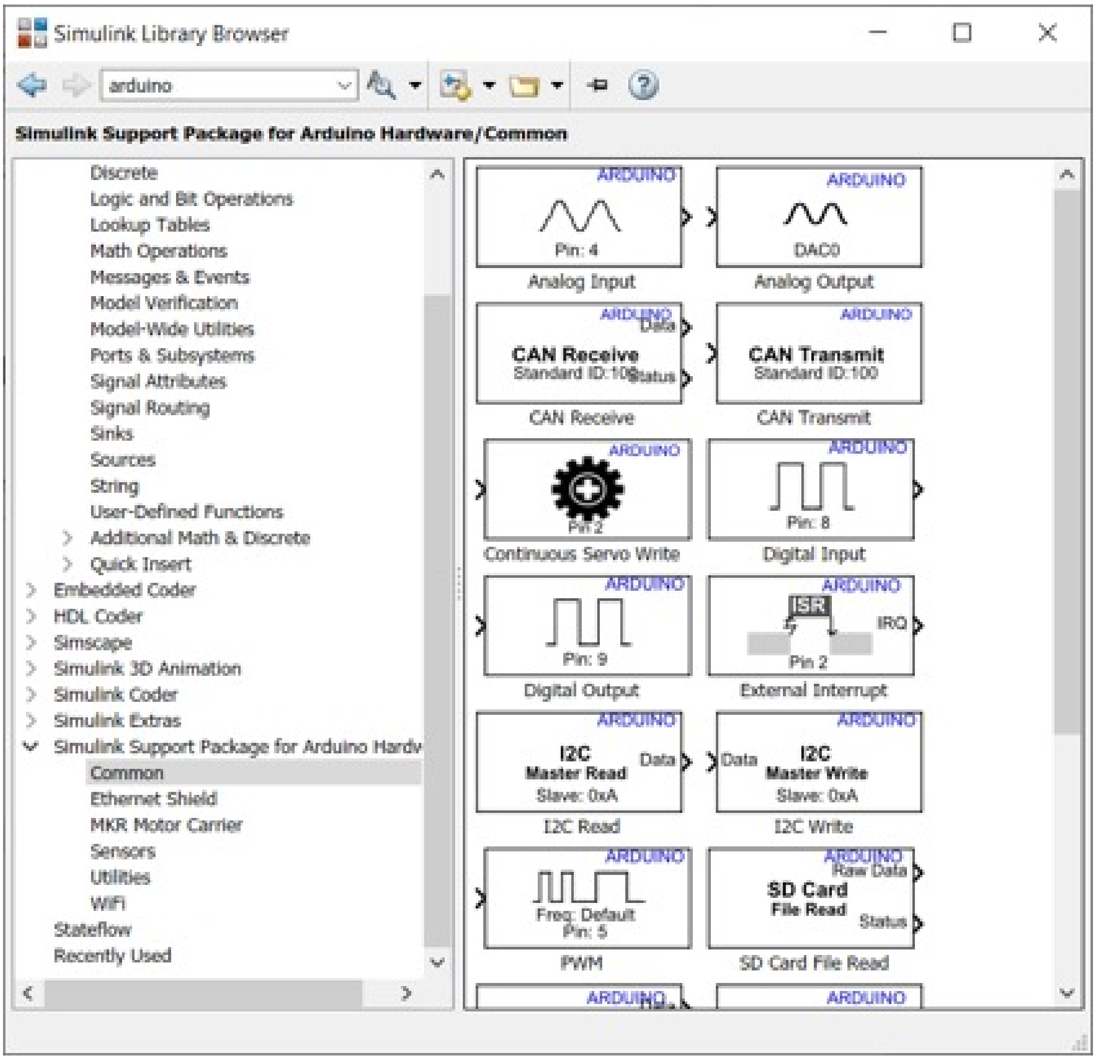

Figure 13: STEP 4.

9.1 CODE GENERATION:

We use two digital input pins to read the state of the switches to which they are connected and a PWM pin to generate the PWM output required to control the BLDC motor. The first switch is for on and off of the motor and the second toggle switch act as the control for the up and down movement of the windows.The pwm duty cycle is set to be a constant as the application that we are focusing on is power windows within a vehicle. After the Simulink model of the controller is completed, the c code can be generated by setting the target for the embedded c code generator to be Arduino Uno board.

9.2 PROCESSOR IN LOOP.

For processor in the loop, it is necessary to read PWM output generated from the Arduino output pin. This output is then fed to the driver circuit. Though simulink support package for Arduino does not provide support at the moment to read PWM output directly from the pin to Simulink, we can do this by processing the PWM output within the Arduino using PulseIn function and then return back the output value to our Simulink model using serial communication. For this a custom function can be written where an an arduino object 'a' is created and then the read command is executed within the Simulink function.

10 CONCLUSION

The model based design of BLDC motor control specifically for the application of power windows within a vehicle was done successfully, using a V model design approach. Initialy, we verified the plant model by giving a test input to the MIL. Code generation for the entire plant was done using embedded coder. Arduino UNO was chosen as the desired controller for the actual implementation and testing using PIL. The controller was designed in simulink using the blocks from simulink support package for arduino But we could not complete the loop due to issues in reading the output from arduino PWM pin to simulink.
Overall, the plant model comprising of motor and sensor was verified using MIL and SIL.