Overload Protection of BLDC Motor Using Arduino

Suchismita Ghosh(1), Sudeep Samanta(1), Visvapa Banik(1), Rahul Pramanick(1), Sital Mondal(2)

1 Department of Electrical Engineering, MCKV Institute of Engineering, 243, G. T. Road, Liluah (N), Howrah-711204, India.
2 Electrical Maintenance Department, Shyam Sel & Power Limited, S S Chambers,5 C.R. Avenue, Floor 2nd & 3rd, Kolkata-700072, West Bengal, India.

Abstract: This paper explores utilizing IoT for the study and control of Brushless DC (BLDC) motors, which are common in industrial applications like robotics and manipulators. IoT enables communication between these motors and other internet-enabled devices, allowing for fine-tuning of control parameters to achieve desired outputs.

Keywords:- Overcurrent, Overload, Relay, Arduino UNO, ESC, and BLDC Motor.

1 INTRODUCTION

A Brushless DC Motor (BLDC) is an electric motor powered by direct voltage supply and commutated electronically instead of by using brushes like in conventional DC motors. BLDC motors offer a high-efficiency alternative to brushed DC motors. The working principle of a BLDC motor is based on Lorentz force law, which states that whenever a current carrying conductor is placed in a magnetic field it experiences a force **[1, 2]**. With the advancement of technology, BLDC motor has been used in car, fan, electric vehicle, drone, lathe machine and robotic applications. For an example, permanent magnet brushless DC motors are best suited for position control and medium sized industrial drives due to their excellent dynamic capability, reduced losses and high torque/weight ratio **[4]**. So, depending on its features like high speed, high torque and noiseless operation, the wide range of utilities are developed. While the motor is used in conveyor, lathe machine and/ or vehicles it is necessary to keep its loading under a tolerable limit. Moreover, before it's applications, it is necessary to know the various parameters like voltage, current, input power, output power, torque, speed, motor efficiency, temperature etc.

The previous research work includes the implementation of IOT which is a type of network where multiple devices are connected which technology is based on the communication between various devices to devices as well as to the cloud **[3]**. In recent study, MATLAB/ Simulink model of BLDC motor drives **[5, 6]** took the interest of the researchers, where fuzzy logic has been applied **[5]**. Phase commutation in automotive applications are also developed along with the simulation of BLDC motor drive model **[7]**. DSP implementation of a PV system supplying BLDC motor drive has been investigated using the machine learning **[8]**. Reduce cost controllers for BLDC motors are introduced which are more in demand and many schemes and algorithms for reduced cost controllers have been reported in the literature, which shows the cost reduction of controllers for BLDC drives by different two approaches **[9-10]**.

With this backdrop, this paper studies about the performance of BLDC motor under different load conditions, and its overload protection using Arduino Uno. The prime objective of this is to establish a simple algorithm which enables to protect the BLDC motor for overload condition, and hence a hardware model of BLDC motor with its loading arrangement and along with connections with Arduino has been developed. Here, at first the methodology has been described, and circuit diagram is provided as well. The component specifications and hardware modeling is also shown in next consecutive two segments. The algorithm which is implemented that is also provided next section. After that the result is shown which is obtained in real time and followed by the discussion is presented.

2 METHODOLOGY

At the very beginning, supply is given from a battery of 12V to the motor controller which actually runs the motor. Here a current sensor is used which measures the current between the battery and motor controller. Current sensor is further connected with the Arduino UNO. An LCD display which is attached with Arduino UNO, is used to display current drawn by the motor. One variable resistor is used to create variation in the motor speed and is connected to the Arduino UNO. After this whole controlling process, a relay module will be connected between the power supply and motor controller for breakdown purposes. The relay module is also connected to the Arduino UNO. In this paper, one threshold current value has been fixed and programmed in Arduino UNO in such a way, so that when the current value exceeds, the relay starts to trip and the motor gets stopped. In case, if ESC module fails, this whole work will give the back-up protection for the BLDC motor and its driver circuit. Here, the simple block diagram is shown in **Figure 1**.

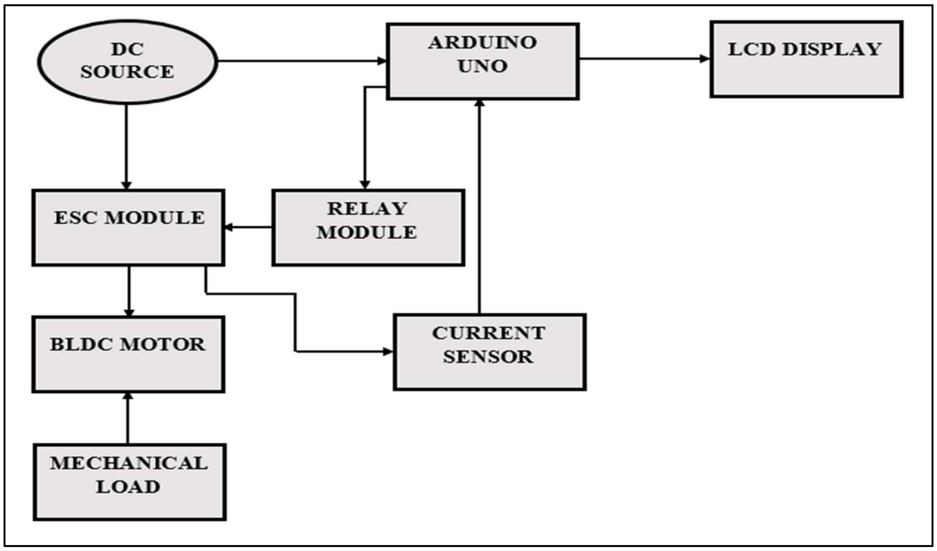

Fig1: Block Diagram

3 CIRCUIT DIAGRAM

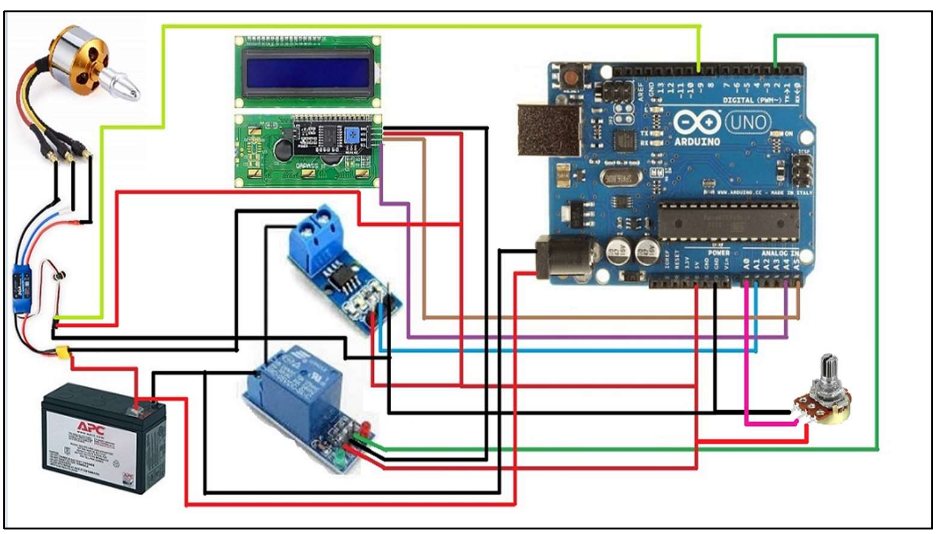

Fig 2: Circuit Diagram

The connection of this work shows that the battery supply is given to the ESC Controller which actually runs the BLDC motor and Arduino is also getting the supply from the same battery. A current sensor is connected to ESC to measure the current value. Here, the Vcc and GND pin of current sensor is connected to the Arduino's Vcc and GND pin and the remaining pin of current sensor is connected to the Arduino's A1 analog pin.

As mentioned previously, in Arduino programming provided below, a threshold current value has been set. So whenever the current value exceeds, the relay will trip. Preferably this set value should be lesser than the threshold current value of ESC module beyond which the connection from supply to the motor gets interrupted and motor may stop. To give the backup protection due to overload, this circuit will work efficiently.

Here, the Vcc and GND of the relay is connected with the Arduino's Vcc and GND pin. The other pin of the relay module is connected with the Arduino's digital pin no 2.

One LCD display module is connected to show the overload condition of the motor. Here the Vcc and GND of the display module is connected with the Arduino's Vcc and GND pin, and other two pins are connected with the Arduino's A4 and A5 analog pins.

The detailed circuit connection considering different pins is shown in **Figure 2**.

4 COMPONENT SPECIFICATIONS AND HARDWARE MODELING

The connection of the components used in this work is already described. In **Table 1**, the specifications of the components along with their manufacturer's name and market price in INR have been provided for cost estimation. With these components, the real time model is prepared and the experiment is done on the same. This arrangement is shown in **Figure 3**. Here, the load applied to the motor exceeded the rated value and thus the LCD monitor displays the overload occurred.

Table 1: Component Specifications

Serial No	Name	Specification	Maker's Name	Quantity	Price (INR)
1	BLDC MOTOR	Model:A2212 10T, Motor KV 1400, Maximum efficiency 80%, Current Capacity: 16A/60S	Generic	1	550/-
2	ELECTRONIC SPEED CONTROLLER	30Amp	Generic	1	450/-
3	ARDUINO UNO R3	Microcontroller: ATmega328P, Operating Voltage: 5V, Input Voltage: 7-12V, Digital I/O Pins: 14, Analog Input Pins: 6, SRAM: 2 KB, EEPROM: 1 KB, Clock Speed: 16 MHz	SHANIWAR PATH Ltd.	1	625/-
4	LEAD ACID BATTERY	12V, 7Ah	EXIDE INDUSTRIES	1	1155/-
5	CURRENT SENSOR	ACS712, 30Amps	Crocus Technology	1	190/-
6	RELAY MODULE	5V, SPDT	OEM Pvt. Ltd.	1	80/-
7	CONNECTING WIRE	M-M, M-F	-	1+1+1	80/-
8	LCD MODULE	I2C 16x2 LCD DISPLAY	TFT LCD	1	400/-
9	VARIABLE RESISTOR	10 KΩ	-	1	25/-
TOTAL					**Rs. 3555/-**

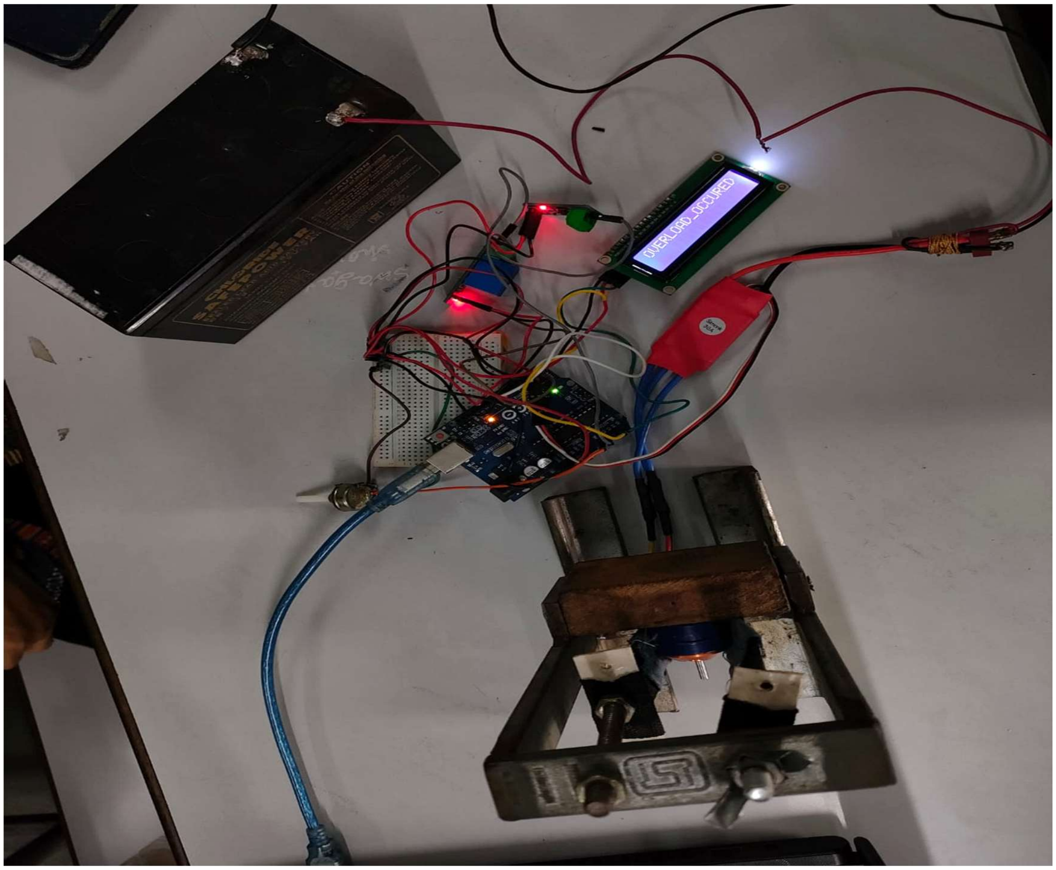

Fig 3: Hardware Representation of Real-time Model

4.1 FLOW CHART

For ease of understanding, a flowchart is shown in **Figure 4**, which represents the processing of the designed model. The rated current chosen in this work is 1.5 Amp. According to this flowchart, the programming in Arduino is provided in **Annexure 1**.

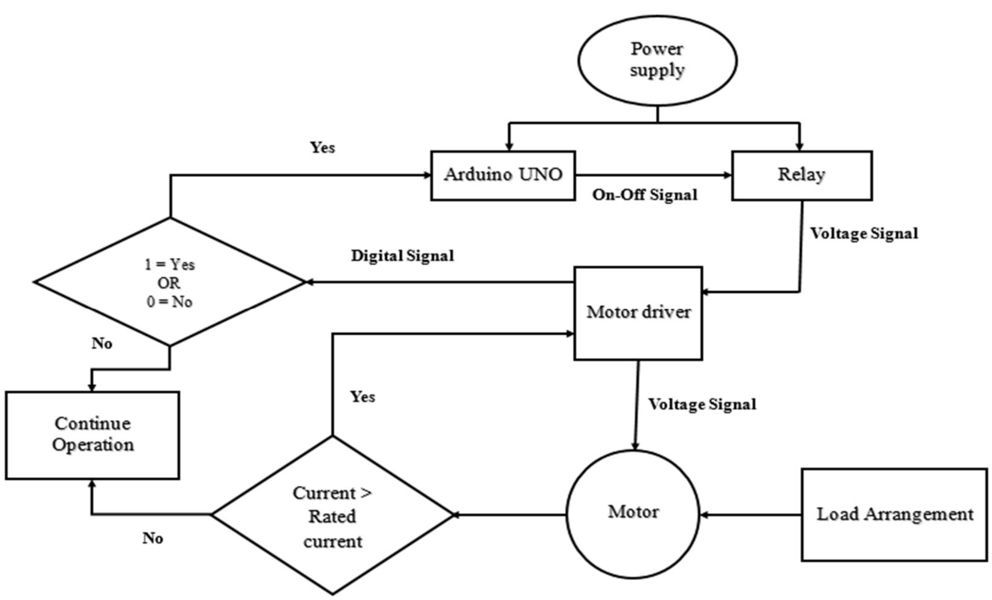

Fig 4 : Flow Chart

4.2 DATA COLLECTION AND ANALYSIS

The connection diagram presented in **Figure 2**, and the hardware arrangement shown in **Figure 3**, provides the graphical plot of current for load variation while simulated via Arduino programming. The graph in **Figure 5** represents the No Load Condition of the BLDC motor. Here the X axis represents the time in ms and the Y axis represents the Current (A). In the No load condition, the motor is drawing near about 0.05A. **Figure 6**, with the same axis, represents the transient behavior of the motor when a very small amount of load is applied. It even shows the characteristics while the load is increased gradually. Hence, **Figure 6** represents the running condition of BLDC motor in on-load conditions. In on-load condition (load of 4.5 unit) the maximum current drawn by the motor is near 0.61A as seen in the above figure.

Figure 7 has the extended part of **Figure 6**, which shows the sudden change of current to a very small value. It happened during the overload caused by the increasing load.

In **Figure 7**, at the position of 235 ms the gradual increase of the load becomes so high and is beyond its rated value. Here, the relay coil trips showing the display as “OVERLOAD OCCURED”. Here the current suddenly falls near to zero and gradually it becomes absolutely zero. This display in LCD is shown in **Figure 3**.

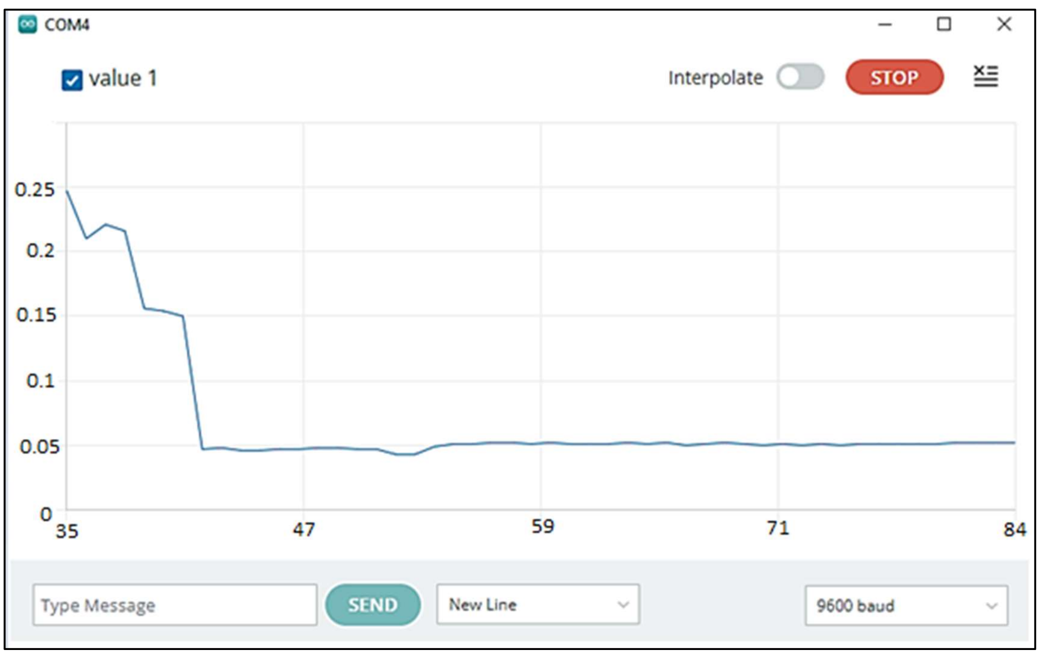

Fig 5: No Load Condition

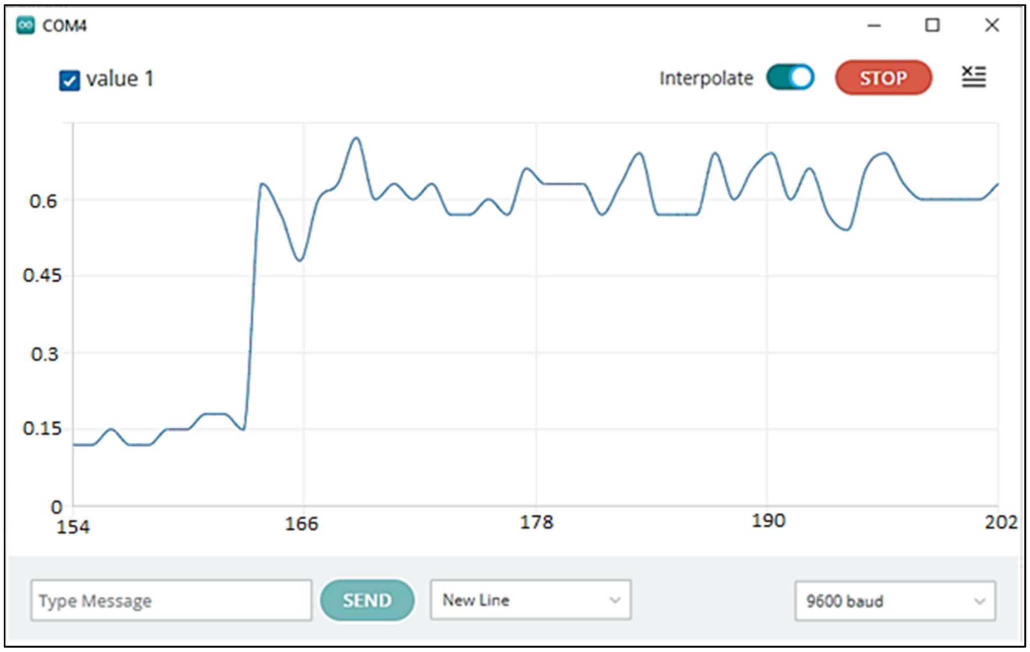

Fig 6: On Load Condition

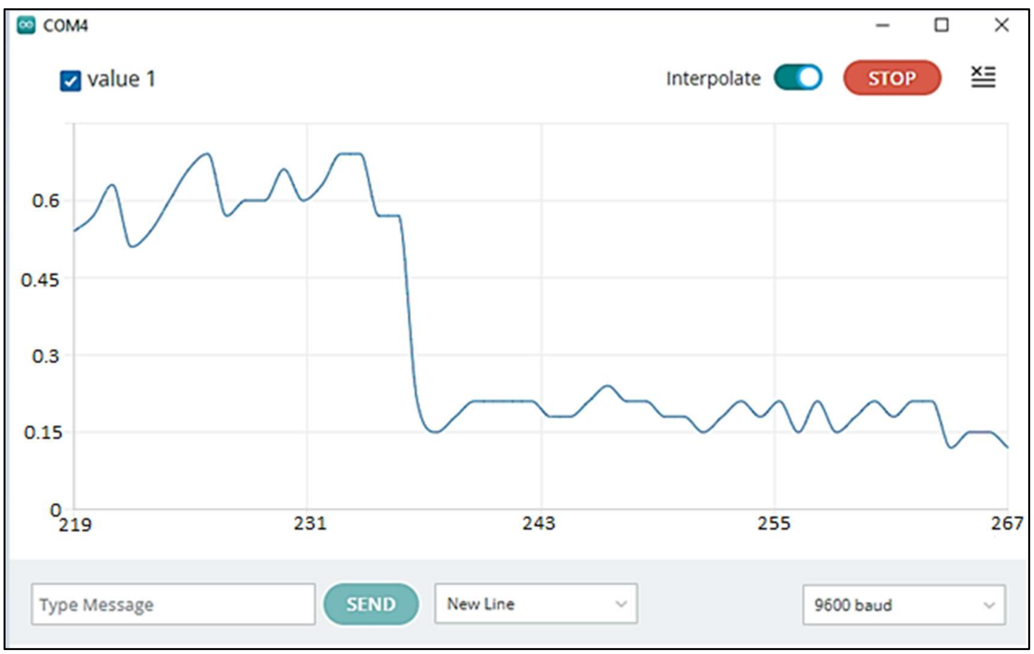

Fig 7: Over-Load Condition

Apart from the transient behavior shown in **Figure 5**, **Figure 6**, **Figure 7**, the steady value of speed and current has been recorded and is shown in **Table 2**. The load applied can be calculated using the following formula:

T=(s1-s2) x G x R N-m ;

where, s1, s2 are the measured value of scale attached with the load to measure the applied load,

G and R comes from the usual convention.

In this work the torque has been calculated in terms of scale difference i. e., (s1-s2) only. Hence it has been represented as in unit of load. However, had the radius of the disc attached to shaft been measured in meter, then torque could be calculated in N-m.

In **Table 2**, different positions of potentiometer have been set and according to different set values, the change in speed and current have been observed. Here, in the potentiometer, in its three connectors (Vcc, ground and, analog signal sender) the terminals Vcc and ground is connected in the Arduino's Vcc and ground port and the analog signal sender is connected in the Analog port of Arduino. And the supply to the motor is controlled by changing the resistance value of variable resistor through Arduino.

Figures 8-11 shows several graphical plots in one frame to get the comparative analysis. In **Figure 8** and **Figure 9**, speed and current change with respect to resistance is observed consecutively for different load variation. The speed control of BLDC motor is achieved by varying the duty cycles (PWM Pulses) to varying input voltage (0V-5V) by the variable resistance in analogue pin of microcontroller's and delivers the desired output to switch the motor drives to control the speed of the BLDC motor.

Figure 10 shows the speed-load characteristics and **Figure 11** shows input current-load characteristics. These show that the speed decreases and input current increases with load increases from 4.4 to 4.6 unit. The motor characteristics can be observed from all of these graphical plots.

SL. NO.	LOAD	POT POSITION	SPEED (RPM)	CURRENT (A)
1	Load 1 4.5unit	position 1 (0.23k)	3340 rpm	0.61A
		position 2 (1.428k)	4166 rpm	0.67 A
		position 3 (3.183k)	7243 rpm	1.27 A
2	Load 2 4.6 unit	position 1 (0.23k)	0 rpm	0.52 A
		position 2 (1.428k)	1830 rpm	1.07 A
		position 3 (3.183k)	1970 rpm	2.01 A
3	Load 3 4.4 unit	position 1 (0.23k)	4253 rpm	0.52
		position 2 (1.428k)	6130 rpm	0.63
		position 3 (3.183k)	8990 rpm	1.19 A

Table 2: Data Table

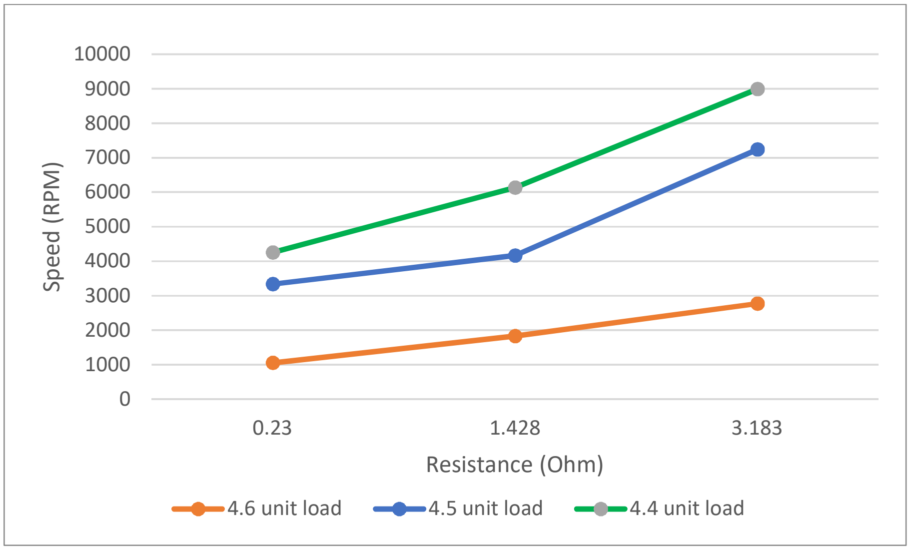

Fig 8: Variable resistance vs speed for different motor loads

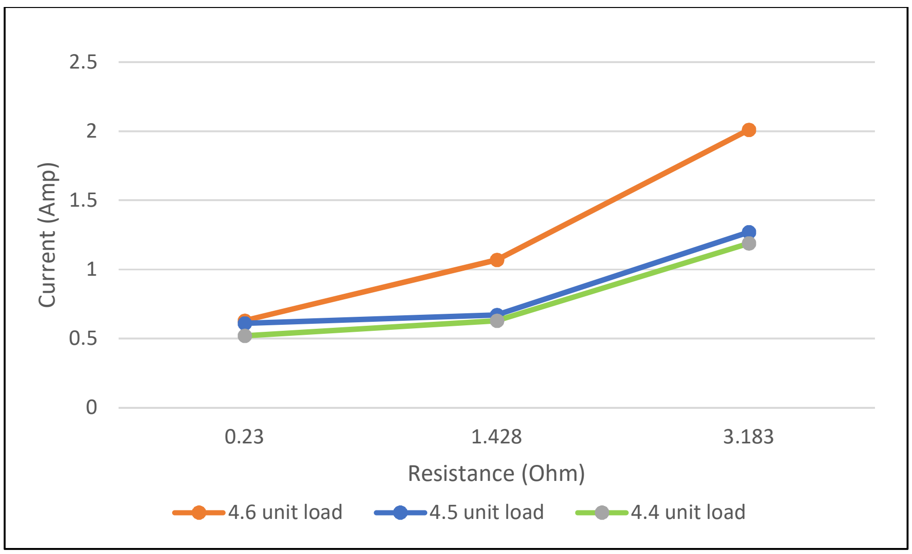

Fig 9: Variable resistance vs current for different motor loads

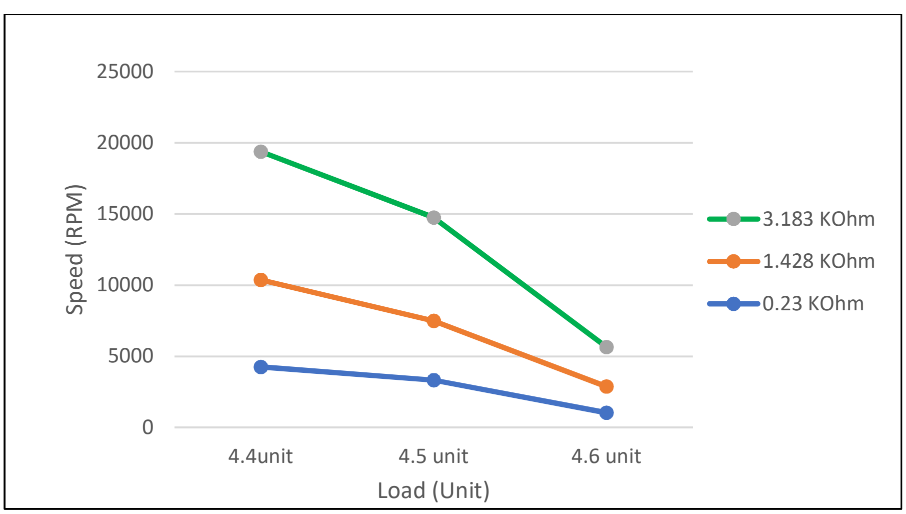

Fig 10: Speed variation w.r.t different loads

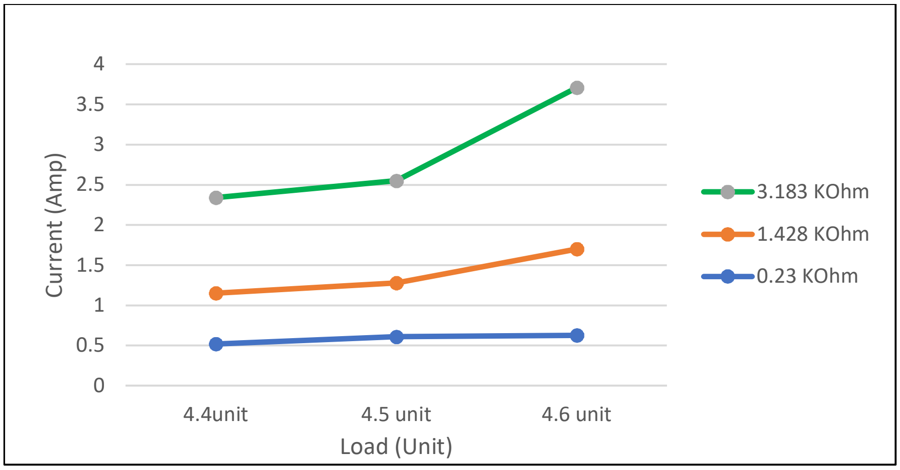

Fig 11: Input current in w.r.t different loads

5 DISCUSSION

This paper shows the design and implementation of an ARDUINO-based protection technique in BLDC motor in real-time. The model is run with ARDUINO programming and thereby the simulation result is shown to study the motor characteristics during overload condition. Moreover, the graphical plots are added with the data obtained during the running condition of the motor. Here, the modeling and simulation is clearly demonstrated and it shows that during overload condition, when the load current exceeds the set value, the LCD displays the condition before the ESC controls over the circuit. Even, if the ESC fails, the overload condition can still be detected with the model presented here. The circuit even trips by the relay coil followed by the performance of display unit. Hence, from simulation results and evaluation, it was observed that the system's behavior for overload conditions is better in terms of equipment safety. The proposed solution proves its feasibility by tripping the motor at the above condition and thereby, ensure the safety of equipment in a long-term run.

Annexure 1: PROGRAM

```
#include<LCD_I2C.h>
LCD_I2C lcd(0X27, 16, 2);
int ampValue;
int digiPin=7;
void setup() {
  pinMode(digiPin, OUTPUT);
  Serial.begin(9600);
  lcd.begin();
  lcd.backlight();
  lcd.setCursor(1,0);
  lcd.print("Welcome");
}
void loop() {
```

```
  unsigned int x=0;
  float ampValue=0.0, Samples=0.0, AvgAmp=0.0, AmpValueF=0.0;
  for(int x=0;x<100;x++){
    ampValue=analogRead(A1);
    lcd.setCursor(0,0);
    Samples=Samples+ampValue;
    delay(3);
  }
AvgAmp=Samples/100.0;
AmpValueF=(2.5-(AvgAmp*(5.0/1023.0)))/0.066;
if (AmpValueF>1.5){
  digitalWrite(digiPin,LOW);
  lcd.print("OVERLOAD_OCCURED");
}else{
  digitalWrite(digiPin,HIGH);
  lcd.print("AmpValue=");
  lcd.print(AmpValueF);
  lcd.print("A");
}
Serial.println(AmpValueF);
delay(100);}
```

5 REFERENCE

[1] Miller T. J. E., et al., "*Design of Brushless Permanent-Magnet Motors*," Magna Physics Publishing and Clarendon Press, Oxford, England, 1994.
[2] Seo J. M., et al., "*A study on brushless DC motor high torque density*," WASET, International Journal of Power Electronics and Drive System (IJPEDS), Indonesia, vol. 58, pp. 225-229, 2011.
[3] Bahga Arshdeep and Madisetti Vijay "*Internet of Things: A Hands-On Approach*" Orient Blackswan Private Limited, New Delhi, 2015.
[4] Singh Bhim and Singh Sanjeev, State of the art on permanent magnet brushless DC motor drivers, Journal of Power Electronics, New Delhi, vol. 9(1), pp. 3-17, 2009.

[5] Mehmet Çunkas and Omer Aydoğdu, Realization of fuzzy logic controlled brushless DC motor drives using matlab/simulink [J]. Mathematical and computational applications, Turkey, vol. 15(2), pp. 218 -229, 2010.

[6] Lee B.K. and Ehsani M., Advanced Simulation Model for Brushless DC Motor Drives, Electric Power Components and Systems, United Kingdom, vol. 31, pp. 841–868, 2003.

[7] Hong W., Lee W., and Lee B.K., Dynamic Simulation of Brushless DC Motor Drives Considering Phase Commutation for Automotive Applications, International Electric Machines & Drives Conference (IEMDC), Houston, Texas, 2007.

[8] Akkaya R., Kulaksız A.A. and Aydogdu O., DSP implementation of a PV system with GA-MLP-NN based MPPT controller supplying BLDC motor drive, Energy Conv. and Management, United Kingdom, vol. 48, pp. 210-218, 2007.

[9] Singh Bhim, Singh B.P., and Jain K., Implementation of DSP based digital speed controller for permanent magnet brushless dc motor, Journal-EL, New Delhi, vol. 84, pp. 16-21, 2003.

[10] Lee B.K., and Ehsani M., Advanced BLDC motor drive for low cost and high performance propulsion system in electric and hybrid vehicles, IEEE electric machines and drives conference, United States, pp. 246-251, 2001.