ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 9, Issue 3, March 2020

Auto-Tuning High Current Dual-Channel Motor Control Unit with Communication

Shreyas Borse [1], Pinak Jani [2], Ritik Jain [3], Kartik Bhadesiya [4]

U.G. Student, Department of Electronics and Telecommunication, K.J. Somaiya College of Engineering, Mumbai, India[1,2,4]

U.G. Student, Department of Information Technology, K.J. Somaiya College of Engineering, Mumbai, India [3]

ABSTRACT: The paper presents the design and development of a dual-channel motor control unit. It is developed with auto-tuning feature and encoder support for both loads. DC motors are controlled using encoder feedback and the auto-tuned result is obtained by implementation of system-identification technique and real-time hardware simulation using MATLAB Simulink. The obtained result is then passed to the motor drive equations for control application. The complete development with the results of the dual-channel motor control unit is explained and clarified in this paper.

KEYWORDS: DC Motor driver, Encoder, Control System, MATLAB Simulink, Auto-tuning, System-identification.

I. INTRODUCTION

DC motor control is very important in automation, robotics and electromotive applications. PID (Proportional – Integral – Derivative) controller is a well-known controller implemented for motor control applications. The PID controller requires fine manual tuning of kp, ki, kd constants, which is tedious at times.[7] The motive behind this was to develop a single module that performs the tasks of motor control and thereby making it simpler by implementing auto-tuning and motor control algorithms that support high current drive applications. The module is designed to support UART communication for IoT based industrial applications. The concept proposes the idea of implementing motor control by real-time simulation to obtain accurate run-time control parameters. The module is designed to communicate with other similar modules by creating a network that is connected over a single UART(Universal Asynchronous Receiver Transmitter) channel and controlled by a single master microcontroller. Further, it explains the functions that are built to process the constants resulting from the model and executing on the hardware we built.

II. DESIGN AND DEVELOPMENT OF BOARD

The motor-controller is developed using STM VNH5019A-E and Arduino Micro ATmega32u4. The basic purpose of integrating the microcontroller was for the auto-tuning of motors and for establishing communication with external controllers. As a rudimentary stage of design, a small break-out of IC VNH5019 was made with an approximate area of the patch to sustain maximum drive current load about 30A. The required biasing components for the ICs according to parameters are taken from the schematic provided in the datasheet of VNH5019[1] see fig 1.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 9, Issue 3, March 2020

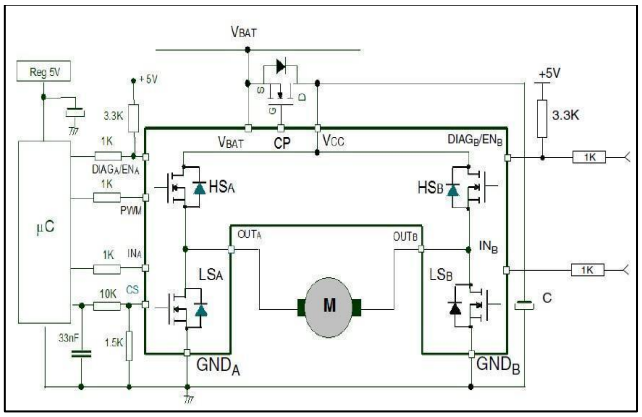

fig.1 Circuit diagram [1]

After testing of the initial breakout of the IC, certain conclusions were drawn which helped in the development of design of the final module. The final design was made by integrating all different modules required for creating the motor-controller. The modules included are On-board microcontroller, reverse polarity protector, Voltage regulation, ISP Programming, Led, external UART output port for communication with master. The design details consisted of three major parts a) Reverse Polarity Protection b) Power and Microcontroller Interface c) Schematic and 3D model.

A. Reverse Polarity Protector

The reverse battery protection is practiced by inserting a NMOS in the right direction of the positive supply line and protects the load against the battery misconnection. In case of reversal of the battery terminals no voltage is supplied forward and the connection is cut preventing harm to any other electronic component[2].A charge pump circuit provides input to the gate terminal of the NMOS. Whenever battery polarity is reversed the diode to the ground as shown in fig.2 is reverse biased and thereby disconnecting thereby charge-pump input and turning the NMOS off. The charge pump input to the NMOS in this application is provided by the VNH5019 IC itself.

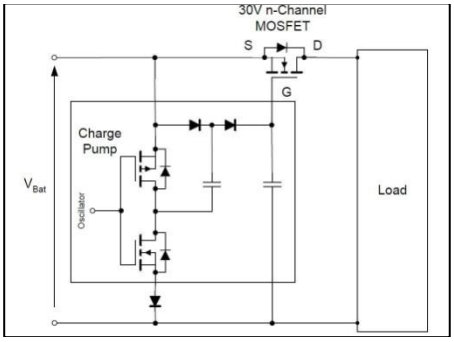

Fig.2 [2] NMOS for S.C. protection.

So, as per electrical requirements, the external Power NMOS used for the reverse battery protection should have the following characteristics:[1]- BVdss > 20 V(for a reverse battery of -16 V); R_{DS} (on) < 1/3 of H-bridge total R_{DS} (on) [1].Considering the above parameters a power NMOS used is CSD18511KTT 40-V N-Channel Power MOSFET [3].

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 9, Issue 3, March 2020

B. Power And Microcontroller

The max input power voltage of the motor-controller is 24V, for stable design parameters of the module. The input power is then regulated by LM7805-SOT223[4] voltage regulator to 5V. This regulated 5V output is fed via a 3-pin switch to power the microcontroller Arduino Micro-ATMega32u4 [5] as well as for the status LEDs and to power encoders. The Arduino Micro performs the auto-tuning task when interfaced with MATLAB Simulink in external mode. The main purpose of considering Arduino Micro as the on-board microcontroller is because there is an Arduino Hardware add-on that Simulink supports which is lucid to develop and deploy. Also, Arduino Micro has the provision of UART communication and five external interrupt pins that are required to interface with incremental encoders. The onboard microcontroller is also fed with different drive functions which makes the motor controlling task very easy for naive users. We have built functions including acceleration, de-acceleration, 2-wheel drive equation, position tuning which are used for embedded robotics applications.

C. Schematic And Model Of Motor Driver.

The PCB schematic is designed as per the features discussed above, and it is divided into two parts.

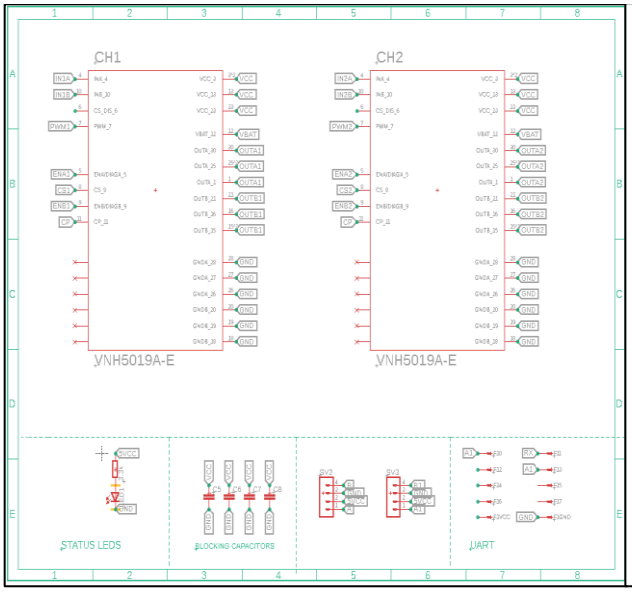

fig. 3.1 Schematic (Part A)

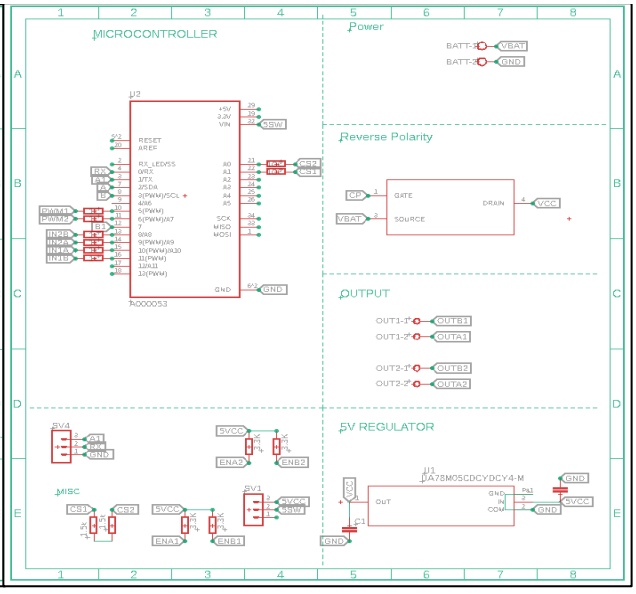

fig. 3.2 Schematic (Part B)

The schematic (Part A) consists of the breakout of VNH5019-AE along with other components like the blocking capacitors, status Led, Encoder connectors, UART FRC connector. The blocking capacitors are for prevention of sudden current spikes generated to damage the other peripherals. These blocking capacitors are connected in parallel to the input power which helps in dead stop or stalling conditions.[1] In such conditions of PWM (Pulse Width Modulation) operation, there is a high spike of current for a few seconds which may malfunction other components; these blocking capacitors help in filtering such spikes or noises and preventing harm to other electronics. Also, these capacitors are selected depending upon the max load current and possible over-voltage that is generated by the motor inductance also known as back-emf. Considering the max load current to be 30A for the motor controller, three 500uF capacitors are connected in parallel. The Encoder relimates are the connectors for the incremental encoder feedback they consist of a four-pin relimates which are i) VCC (5V) ii) GND iii) Channel A iv) Channel B. The A and B channels are the encoder outputs are connected to the external interrupt pins of the onboard

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 9, Issue 3, March 2020

microcontroller, which then determines the RPM from these inputs from encoders. The VNH5019A-E ICs are interfaced with microcontrollers via biasing resistors (1210 smd package). Following are the pin connections done between the motor driver IC and Arduino Micro:

VNH5019-AE MOTOR DRIVER IC	ARDUINO MICRO PINS
IN1A	D10
IN1B	D11
PWM1	D5
IN2A	D9
IN2B	D8
PWM2	D6
EN1A	PULLED UP VIA 3.3Kohm
EN1B	PULLED UP VIA 3.3Kohm
EN2A	PULLED UP VIA 3.3Kohm
EN2B	PULLED UP VIA 3.3Kohm
CS1	A1
CS2	A0

fig.4 Pin mapping

As mentioned in the schematic, these pins are connected via different resistors used for biasing. The resistor in series to INA, INB and PWM pins are 1kΩ resistors. The ENA, ENB which are the enable pins are pulled up to 5V via 3.3kΩ resistors and the current sense pins CS are connected via 1kΩ series resistor and pulled down via 10kΩ resistor to the respective Arduino Micro pin (fig.3.2).[4][1]. Decoupling capacitors are connected to the input and output of LM7805 with 100uF at the input and 10uF output, both these capacitors (smd package 1206). The status LED (smd package 1206) is present to indicate the power of the module. The gate of reverse polarity MOSFET is connected to the charge-pump pin of the motor driver IC and the source is connected to the input battery terminal (VBAT). The drain of the NMOS CSD18511KTT is the VCC output which is supplied further[3].Screw terminals are used for the input from the battery and for the output to the motors. A dual-layered board layout is designed with all the electrical parameters and efficient placement of the components to fulfill the complexity of multiple traces. Different trace widths are set considering the max amount of current flowing through it.The designed unit has maximum trace patches of 5.5mm width to drive high current up to 30A on 4OZ the copper thickness of the board. The layout has top and bottom layered ground planes along with vias of 0.3 mm diameter. The overall dimension of the motor-controller is 83x84mm. The complete design of the board and model is as shown in fig. 5 and fig.6 respectively.

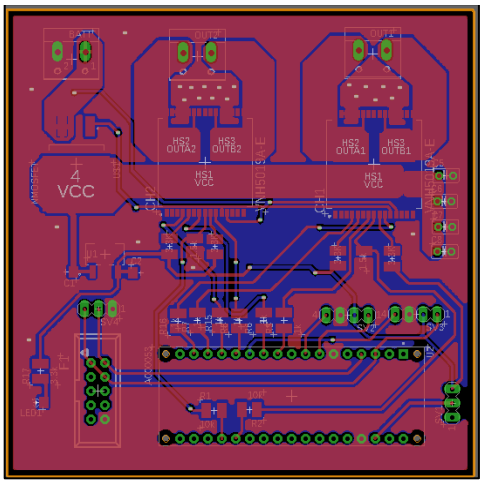

fig.5 Board layout.

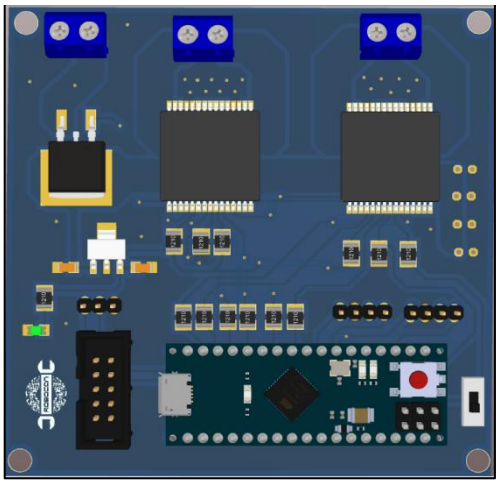

fig.6 3D Model top View .

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 9, Issue 3, March 2020

IV. DEPLOYED MATLAB SIMULINK MODEL

A MATLAB Simulink model is deployed on the onboard ATMega32u4 to obtain the auto-tuned PID constants to control the motor. These constants are then fed to the motor control equations to perform the desired operation. PID tuning is possible in two ways 1) Physical tuning 2) Model-based tuning. Physical tuning is the manually tweaking of constants by running the trial and error approach on physical hardware, whereas the model-based tuning is creating a mathematical model of the system by different equations and then simulating the model to obtain the PID constants[8]. The physical tuning is rather time-consuming and tedious; also the model-based tuning requires a lot of mathematics. In order to overcome these problems,a system-identification process is implemented on the motor control unit. This process builds a mathematical structure of dynamic systems by knowing the input and output response of the system.

MATLAB supports an add-on system-identification app that helps in such physical hardware-based model tuning. This system-identification app develops a transfer function based on the input and output response provided. Motor-controller tuning is done with the help of encoder feedback. The encoder output is in the form of high-low pulses, these pulses are converted to ticks and then to determine the speed in RPM. A Simulink model is deployed to determine the output speed response at a particular input. This model is developed with the help of Arduino Hardware Add-on. The Arduino Hardware add-on features various blocks with different functions like digital input, digital output, PWM, External interrupt, etc. These blocks along with other Simulink blocks are used to generate the output speed response in real-time running on external mode.

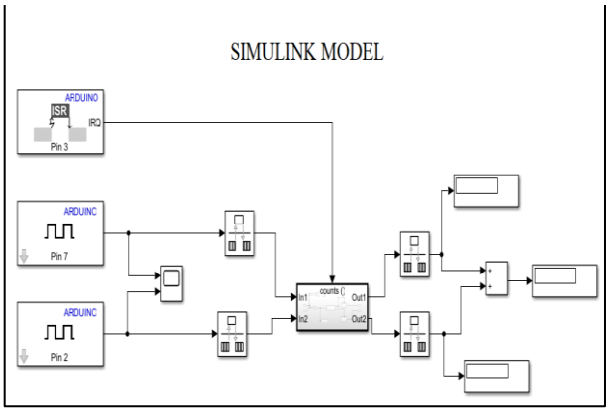

fig.7 Simulink Model

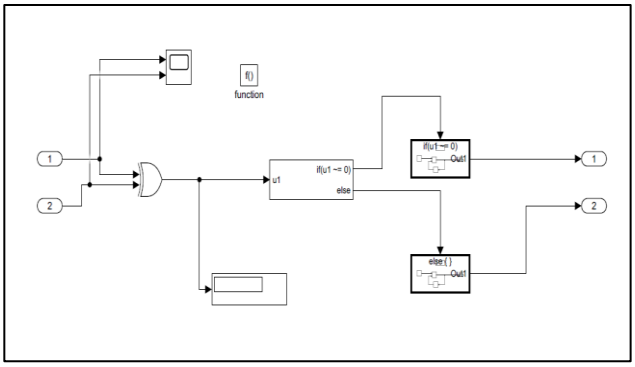

fig.8 Subsystem

The block consists of subsystem counts(), this is subsystem is expanded (fig 8).This subsystem takes the input from both the encoder channels on external interrupts and then performs the operation to increase or decrease the encoder count which is then subsequently converted in terms of speed. This count function has two more conditional subsystems which are, the if and else subsystem; they do the task of incrementing and decrementing based on the provided condition. Following is the expansion of these conditional system :

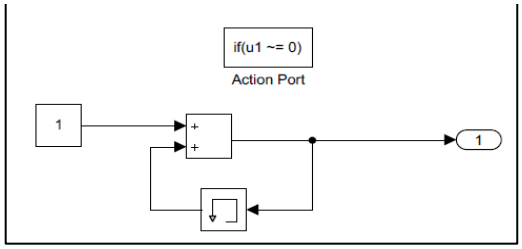

fig . 9.1

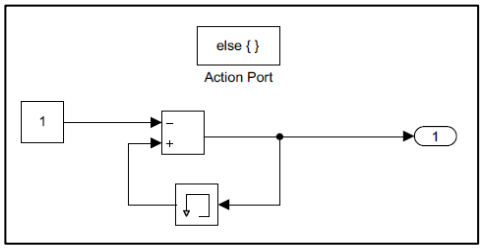

fig. 9.2

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 9, Issue 3, March 2020

The input and output from this Simulink block are sent to the Matlab workspace at real-time when simulated in external mode and the input and output response graphs are plotted accordingly.

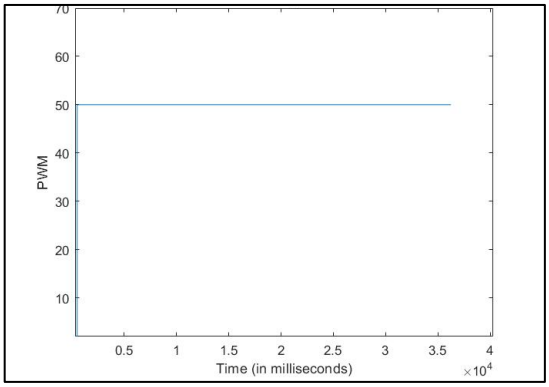

fig.10 Input PWM

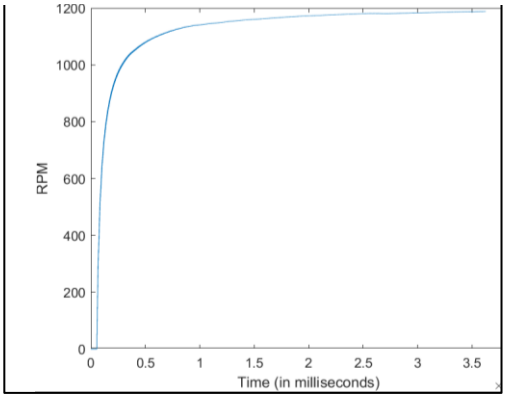

fig.11 Response of RPM

These graphs are then used in the system identification method to determine the transfer function of the system based on the number of poles and zeros.

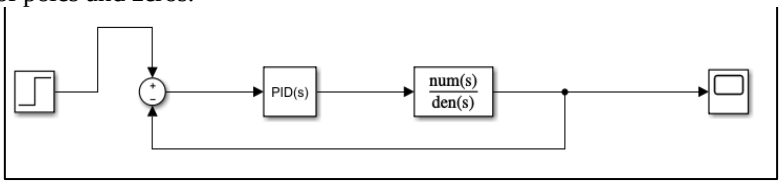

fig.12

This transfer function is then put in as a model and the PID constants are developed by the auto-tune PID block of Simulink model as shown in the fig.12. The PID constants are then obtained as a result of the auto-tuning process[8]. The resultant constants are then passed to the functions as parameters to perform the motor control applications.

V. FUNCTIONS

To control and drive the hardware according to the need, the algorithms are built and the codes are executed on ATMega32u4. UART is used to communicate between the other master microcontroller and the motor-controller unit that we built. Each instruction to be fed is broken down and packed together in a block before transmission.

Length of Block (8-bits)
Address (8-bits)
Instruction Code (8-bits)
Parameters (0-64 bits)
Checksum (8-bits)

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 9, Issue 3, March 2020

Length of blocks represents the number bits the block will contain, it is required as the blocks are continuously transmitted so to differentiate between the blocks the length is required.Each motor controller has its own unique address by which it is addressed. The addresses can be between 128 to 132 and none of the controllers should have the same address if it is being used in series as it would lead to a conflict. If all the units are connected in series only one should be connected to a microcontroller which acts as a master and forms a communication channel. A total of five units can be connected together in series. Connecting more than five controllers in series would cause a delay between the communication of microcontroller and the motor-controller. Each function has an instruction code representing it, and the set of parameters associated with the execution of those functions.If some of the bits are missed as a transmission error, it is detected and the entire block can be discarded by the checksum block. In such a case a request for re-transmission is sent to the master microcontroller from the motor controller unit. Few functions are explained below as per the working with communication with master.

drive_M1 (Address, PWM) and drive_M2 (Address, PWM**)** are used to rotate the channel 1 motor and channel 2 motor respectively, the address of the particular motor-controller is given, along with the PWM. At PWM 0 the motor stops and at PWM 1 to 127 the motor moves in a clockwise direction and PWM -1 to -127 the motor moves in an anti-clockwise direction. Block received by the motor-controller.

4	Address	30/31	PWM	Checksum

dist_M1(Address,dist,p,i,d) and dist_m2(Address,dist,p,i,d) are used to rotate the motor of channel 1 and channel 2 respectively for a fixed encoder count. The PID constants are supposed to be used from the results we achieve from the matlab model, so that the motor rotates precisely for the given counts.The position of the motor can be verified using the read encoder function.Block received by the motor-controller[6].

7	Address	Dist	36/37	P	I	D	Checksum

readEnco1 (Address) and readEnco2 (Address) are for reading the encoder counts of channel 1 encoder and channel 2 encoder respectively assigned with instruction code 32/33, the address of the particular motor-controller is given. This function returns the value of the encoder counts. The encoder connected should be an incremental type[6]. Block received by the motor-controller –

3	Address	32/33	PWM	Checksum

As only 8bits can be transmitted at a time the encoder reading is transmitted in base 256 number system.

N	Bit N	Bit (N-1)	...	Bit 1	Bit 0

reset_Enco1(Address) and reset_Enco2 (Address) are providing the reset counts of the encoder of channel 1 encoder and channel 2 encoder respectively, the address of the particular motor-controller is given. So now when you read the encoder counts it will start from 0. Block received by the motor-controller –

3	Address	34/35	PWM	Checksum

The communicated message block is broken down to parameters by onboard ATMega32u4. Before running the functions, the block is checked for any errors using the checksum bit. The instruction code is checked and the corresponding function is executed using the parameters if there are any. For example, the instruction is 30 and there would be a single parameter, then the following code will get executed using the parameter as PWM magnitude.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 9, Issue 3, March 2020

```
void driveCH1()
{
  if(parameter1>127)
    parameter1=127-parameter1;
  if(invert_dir_CH1==1)
    parameter1=-parameter1;
  if(parameter1>=0)
 {
 analogWrite(CH1_pwm,parameter1*2);
digitalWrite(CH1_dir1,HIGH);
digitalWrite(CH1_dir2,LOW);
 }
  else
  {
analogWrite(CH1_pwm,-parameter1*2);
    digitalWrite(CH1_dir1,LOW);
    digitalWrite(CH1_dir2,HIGH);
  }
}
```

VI. EXPERIMENTAL RESULTS

The printed circuit board of the unit as per the discussed design and parameters is shown below(fig.13). The testing of the unit is done on load (motor) which has peak current of 30Amps and a maximum speed of 4000 RPM resulting in real-hardware implemented auto tuning response plot.(fig.14)

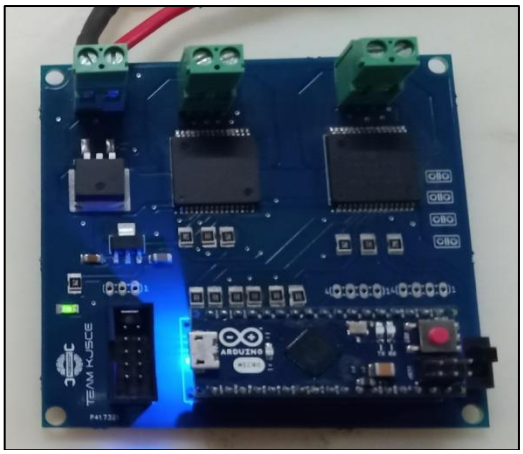

fig.13 Motor driver unit

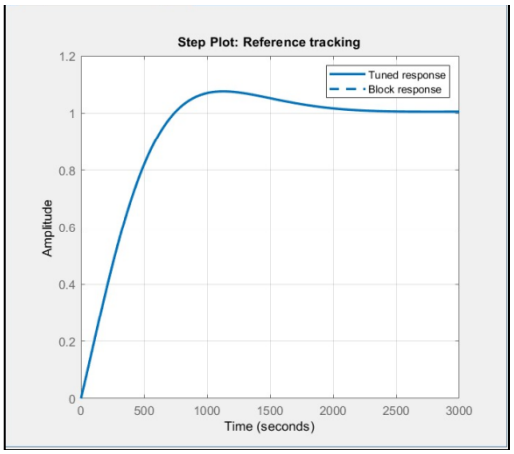

fig.14 The Auto-Tuned Response vs Time

VII. ACKNOWLEDGEMENT

We are very thankful to PCB Power for providing us with the printed circuit boards as per our specifications, we thank K.J. Somaiya College of Engineering for providing us with the required essentials and test equipment along with the resources for this project. We would also like to thank our Prof. Dr Irfan Siddavatam for their guidance and encouragement throughout the research and development process.

VIII. CONCLUSION

We have implemented a dual-channel motor-controller unit which is successfully auto-tuned for both loads using the system-Identification process and controlled using the drive control algorithms. It is inferred that the PID tuning is possible in a simpler yet accurate approach using the system-identification on MATLAB simulink as compared to the physical and model-based tuning. Also, it is feasible in robotics, industrial automation for high current

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 9, Issue 3, March 2020

applications. It is practically concluded that the motor-controller efficiently satisfies the high-current load requirements and it can be further developed to be controlled using IoT applications.

REFERENCES

[1] STMicroelectronics,"Automotive fully integrated H-bridge motor driver", Datasheet, Jan.2017, https://www.st.com/resource/en/datasheet/vnh5019a-e.pdf

[2] Semiconductor and System Solutions - Infineon Technologies, "Automotive MOSFETs Reverse Battery Protection Rev2",datasheet,Application Note, 2.0, June 2009 https://www.infineon.com/dgdl/Reverse-Batery-Protection-Rev2.pdf?fileId=db3a304412b407950112b41887722615

[3] Texas Instruments,"CSD18511KTT 40-V N-Channel NexFET™ Power MOSFET", csd18511ktt datasheet, July 2017 http://www.ti.com/lit/ds/symlink/csd18511ktt.pdf.

[4] Texas Instruments,"LM340, LM340A and LM7805 Family Wide VIN 1.5-A Fixed Voltage Regulators", SNOSBT0L Datasheet, February 2000 [Revised Sept. 2016].http://www.ti.com/lit/ds/symlink/lm340.pdf.

[5] https://store.arduino.cc/usa/arduino-micro

[6] Joshi, Bharat & Shrestha, Rakesh & Chaudhary, Ramesh. (2014). Modeling, Simulation and Implementation of Brushed DC Motor Speed Control Using Optical Incremental Encoder Feedback.

[7] PATIL, M. (2014). Modelling and simulation of Dc drive using PI and PID controller. IJIREEICE, 2263-2266. https://doi.org/10.17148/ijireeice .2014.21213

[8] Reddy, H.K. & Immanuel, J. & Parvathi, C.S. & Bhaskar, P. & Sudheer, L.S.. (2011). Implementation of PID controller in MATLAB for real time DC motor speed control system. Sensors & Transducers. 126. 110-118.

[9] Hassan, Ali & Al-Shamaa, Nabil & Abdalla, Kasim. (2017). Comparative Study for DC Motor Speed Control Using PID Controller. International Journal of Engineering and Technology. 9. 4181-4192. 10.21817/ijet/2017/v9i6/170906069

[10] Mezher, L. S. (2019). Position control for dynamic DC MOTOR with robust PID controller using MATLAB. *International Journal of Advanced Trends in Computer Science and Engineering*, *8*(3), 936-942. https://doi.org/10.30534/ijatcse/2019/92832019.