Control Law Development and Implementation for a One Degree-of-Freedom Copter

Ketan Anand, Nitish Bhat, Pavan N, Shreyas RK and K. M. Vanitha

Dept. of Electronics and Instrumentation Engineering
M. S. Ramaiah Institute of Technology, Bangalore
anand_ketan@outlook.com

Abstract. This report presents the design, implementation, and analysis of a modified PID controller to stabilize the angular position of a 1-DOF copter system. A dynamic model of the system was obtained using classical mechanics. Simulation was performed to showcase the effectiveness of feedback control in stabilizing the system. The control algorithm was implemented on the ESP32 microcontroller. The controller performed well within the linear operating region of ±30° but showed increased oscillations, settling times and overshoot beyond that range. However, the controller could not guarantee stabilization for hover angles beyond 50°.

Keywords: Control Algorithms, Wireless Communication, Mechanical Actuation, PID Algorithm.

1 Introduction

The degrees of freedom (DOF) of a mechanical system essentially encompasses information of how many parameters are required to define the position of a rigid body in n-dimensional space. In this paper we present the implementation of a 1-DoF copter system which allows the movement of a propeller-shaft system in only one direction, that is rotation about the pitch axis. Since the analysis and development of a 1-DoF system is relatively simple, it makes it easier to understand system integration and control strategies utilized for flight control systems.

Control algorithms such as PID help in achieving stability [8], and are employed by many industries to achieve optimized, automatic control of systems. Hence, understanding a relatively simple controller such as PID gives more insights into how control strategies can be developed and improved. The modified PID algorithm implemented in our system is discussed in Sec. 4, with an overview of the hardware setup and software flowchart in preceding sections. The formulation and implementation of this modified PID algorithm is the crux of this paper.

A motor is an electrical machine that converts electrical energy into mechanical energy and is widely used in modern devices from washing machines to printers. Motors are a fundamental part of our system and knowledge of how to drive them effectively can prevent damage of system elements. It serves as a vital actuator in the proposed 1-DoF copter setup.

Real-time applications that need strict timing deadlines and reliability often need fast processing of real-time data and determinism in execution. Due to the advent of powerful embedded systems, performing multiple tasks concurrently is possible and allows for the possibility to build computationally intensive systems. The proposed 1-DoF copter system simultaneously collects an input from the user, measures hover angle of the copter and displays the system details, as presented in Sec. 3 and 5. Carrying out multiple tasks in this manner, is possible by utilizing an RTOS setup.

2 Methodology

2.1 Block Diagram of the System

The methodology can be split into two parts: First, the hardware subsystem which essentially comprises the data acquisition system, the OLED display system and the motor driver system. Second, the software subsystem which consists of an RTOS running on a microcontroller [9]. The microcontroller (ESP32) is central to the system architecture and is used to control the motor driver, OLED display, communicate with the user wirelessly and sample the current angle. The user can input the desired hover angle and perform "on the fly" tuning parameter (Kp, Ki, Kd, B) changes via the Bluetooth serial interface. The current angle given by the angle sensor is sampled by the microcontroller after passing through a signal conditioning circuit. Based on user input and current angle sensed, the microcontroller controls the motor driver by generating an appropriate control signal. The motor driver drives the motor based on the control signal which in turn propels the compound pendulum system to hover at the desired angle. The copter parameters such as current angle, desired angle and control parameters are displayed on the OLED display by the microcontroller.

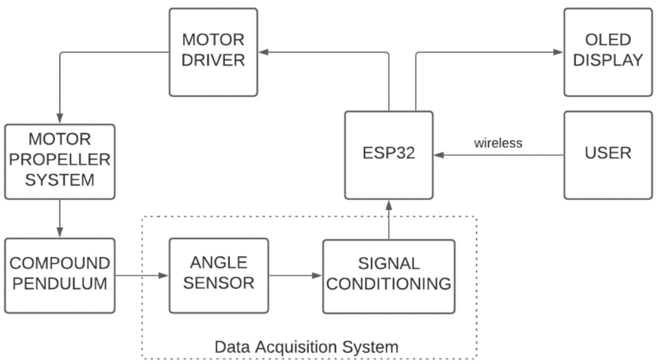

Fig. 2.1. Block Diagram of the 1-DoF Copter System

2.2 Compound Pendulum

It consists of a wooden beam with a coreless DC motor attached to one end and a potentiometer knob attached to the other. A 45mm propeller is attached to the motor shaft which generates thrust when the motor rotates clockwise. The end that is connected to potentiometer acts like a pivot and when the copter rotates about the pivot, it changes the resistance of the potentiometer which gives an indirect measurement of angle [1].

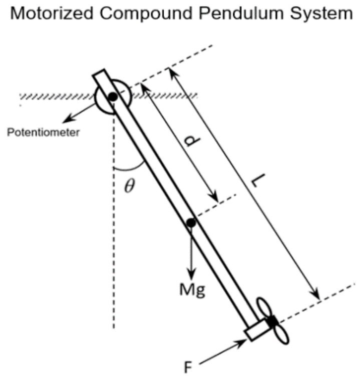

Fig. 2.2. Structure of the Compound Pendulum

2.3 OLED Display

The OLED display is used to monitor real-time system information such as control parameters (Kp, Ki, Kd, B), control signal (PWM) duty cycle, current angle and desired angle.

Fig. 2.3. OLED Display Used to Display Various Parameters

2.4 Angle Sensor

A 10k potentiometer is used as an angle sensor for the copter system. A change in potentiometer resistance occurs due to the rotation of the wooden beam about the pivot, which produces a corresponding voltage change that is processed by the signal conditioning circuit [2]. This change in voltage is sampled by the microcontroller ADC after signal conditioning.

Fig. 2.4. 10k Potentiometer Used as Angle Sensor

2.5 ESP32

The ESP32 is a dual core, low-power microcontroller with integrated Bluetooth functionality. It is a powerful microcontroller capable of performing multiple tasks such as handling user input and controlling peripherals concurrently.

Fig. 2.5. ESP32 Microcontroller

2.6 Motor Driver

The motor driver controls the motor voltage and direction based on the inputs from the microcontroller. It has a H-bridge topology but the motor is made to rotate in a single direction (clockwise). A breakout board houses the motor driver IC and has power supply decoupling capacitors installed.

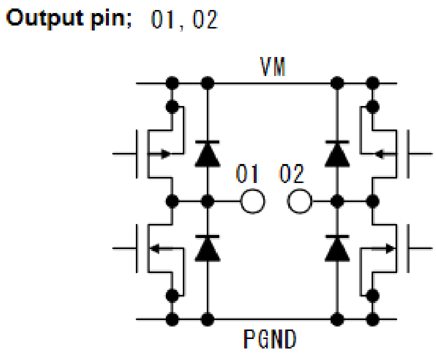

Fig. 2.6. Internal H-bridge Circuit of TB 6612

2.7 Motor Propeller System

A 45mm propeller is attached to the motor shaft which generates thrust when the motor is energized and rotates clockwise.

3 Hardware and Software Implementation

3.1 Hardware Selection

Serial Bluetooth terminal is a mobile application used to send information using Bluetooth between 2 paired devices. Here, the angle information is sent from a mobile device to the ESP32. The entire programming of the microcontroller was done on Visual Studio Code due to its user-friendly interface, relative ease of programming and at the same time, the versatility it provides [3].

The selection of the ESP32 was due to the availability of integrated Bluetooth, extensive number of GPIO pins, availability of integrated ADC, DAC channels and ability to provide high performance concurrency between multiple threads. The selection of TB6612 motor driver IC was due to it meeting the ratings of the coreless DC motor with a max current handling capacity of 1 A, support of low motor voltages and higher efficiency due to the usage of MOSFETs as opposed to L298N that uses BJTs [4]. The selection of the OLED display was due to a smaller number of pins that are needed to control it since it uses the I2C protocol.

As shown, the ESP32 is central to the architecture and controls the other peripherals. The GPIO pins 12, 13 are used to control the motor driver where pin 12 determines whether the motor turns clockwise or turns off completely and pin 13 controls the PWM input to the driver. The OLED uses the I2C protocol and is powered by the internal regulator of the ESP32 and the SCL and SDA pins are controlled by GPIO22 and

GPIO21 respectively [5]. GPIO36 of the ESP32 is used to acquire angle information from the data acquisition system.

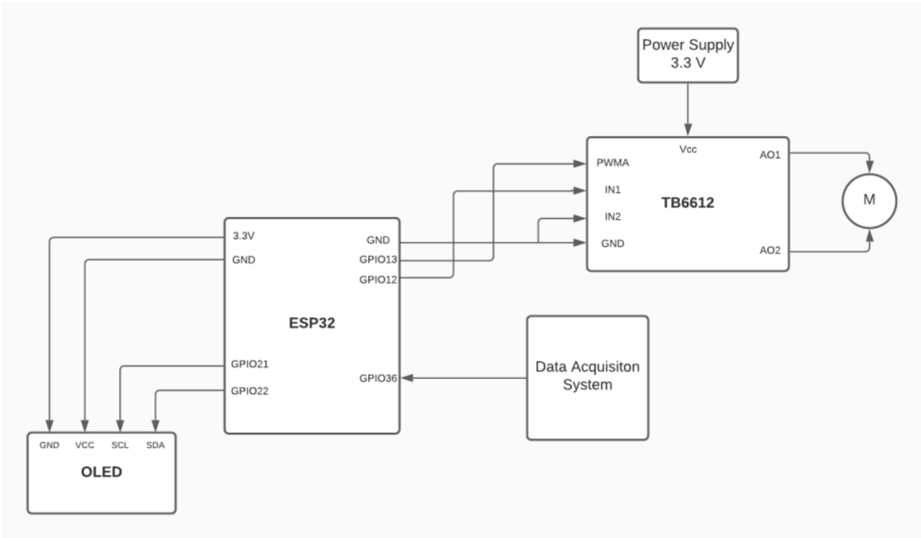

Fig. 3.1. Diagram of Hardware Interfacing

3.2 Software

A Real-Time Operating System (RTOS) is an operating system that is designed to provide a deterministic execution pattern intended to be used in real-time applications having strict timing deadlines [6]. An RTOS provides the illusion of simultaneous execution by rapidly switching between multiple threads. Therefore, an RTOS provides a platform for a microcontroller to perform several tasks such as handling user input and controlling peripherals concurrently.

RTOS in this system performs mainly three tasks concurrently:

1. Getting input from the user
2. Getting the current angle of the copter, running the PID algorithm and updating the PWM duty cycle to drive the motor.
3. Update the OLED display with current angle, setpoint and copter parameters.

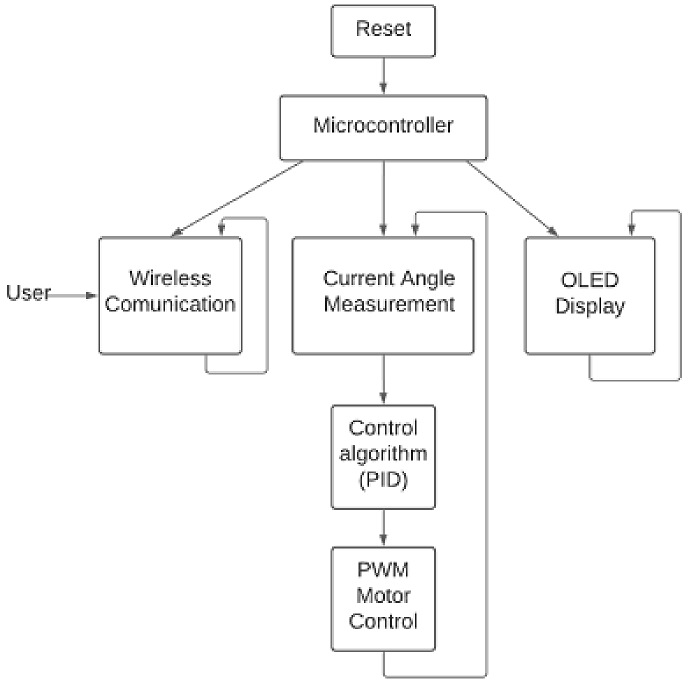

Fig. 3.2. RTOS Setup

4 Control Architecture

4.1 PID Controller

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt}$$

where, K_p is the proportional gain, K_i is the integral gain, K_d is the derivative gain, $e(\tau)$ = SP – PV(t) is the error (SP is the setpoint, and PV(t) is the process variable), t is the time or instantaneous time (the present), τ is the variable of integration (takes on values from time 0 to the present time)

We modify the PID algorithm in the following ways:

1. Clamping (limiting) integrator output to prevent integral windup
2. $K_d \frac{de(t)}{dt} = -K_d \frac{dPV(t)}{dt}$ when setpoint is constant. This eliminates derivative kick
3. $K_i \int_0^t e(\tau)\, d\tau \rightarrow \int_0^t K_i e(\tau)\, d\tau$ eliminates bump during "on the fly" parameter (K_i) change
4. Utilization of a weighting factor B (between 0 and 1) to obtain a ratio between proportional on error and proportional on PV to reduce proportional kick

5 Implementation

The system comprises two essential parts: The user interface and the copter system. And the user can communicate with the system via the Bluetooth serial terminal as per the list of commands.

5.1 Flow Description

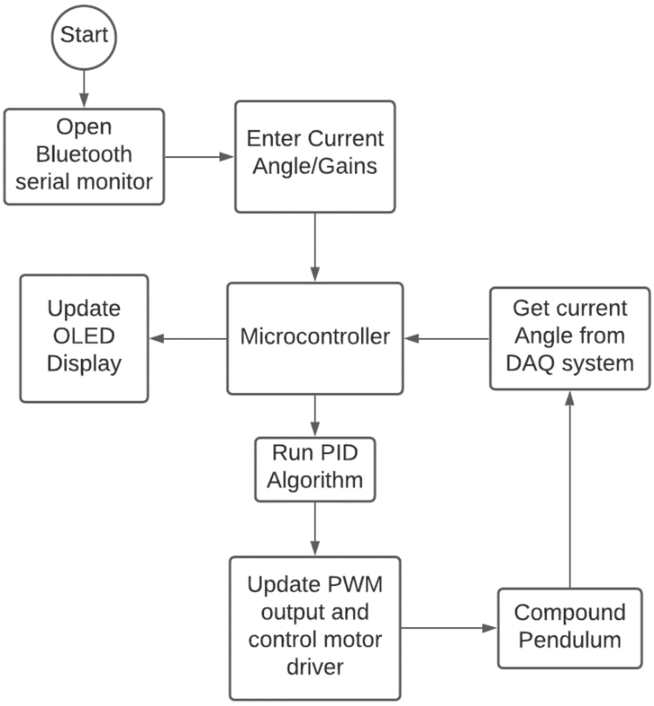

Fig. 5.1. Flowchart of System Processes

The sequence of processes occurring is as follows:

1. START
2. “Pair” is pressed to pair the microcontroller and the user’s device.
3. Setpoint angle/gains are entered in Bluetooth serial monitor.
4. The current angle of compound pendulum is continuously monitored for any setpoint change or disturbance applied to the system.
5. The microcontroller evaluates the PID algorithm using the user set parameters and current angle and updates the PWM duty cycle output.
6. The PWM duty cycle is then used to control the speed of the motor which in turn changes hover angle of the system accordingly.

7. The current angle and the control parameters are updated to the OLED Display.
8. STOP

5.2 Component Interfaced

1. The microcontroller: ESP32 is utilized in this system to automate the process and communicate with external peripherals.
2. OLED Display: OLED Display is used to show the current angle and the control parameters via the I2C communication protocol.
3. TB6612: The motor-driver is interfaced with the ESP32 to control the motor.

6 User Interface – Serial Bluetooth Terminal

6.1 Working

The system communicates with the user through the Bluetooth serial terminal. First pair the two devices using Bluetooth [7]. This can be done by turning ON Bluetooth in phone settings and pair the devices. Once the devices are paired open the Bluetooth serial terminal application as shown below in Fig. 6.1. Select the '1-DoF copter' to connect with the system.

Fig. 6.1. Entering Commands to the 1-DoF Copter System

6.2 Commands to Control the System

Instructions	Description
<number>	Any number entered is taken as setpoint angle to the system and is limited to -90 to 50 degrees.
kp <number>	Change the Proportional gain of the system
ki <number>	Change the Integral gain of the system
kd <number>	Change the Derivative gain of the system
beta<number>	Change the weighting factor (between 0 and 1) to determine percentage of proportional on error and proportional on PV the system.
off	Bring it back to the initial state.

6.3 Data Acquisition System

The data acquisition system consists of a 10k potentiometer that changes resistance when the copter system rotates [9]. This change in resistance changes the voltage across the potentiometer wiper which is sampled by the ADC after passing through a low pass filter and a voltage follower [10]. The voltage follower acts as an impedance matching device.

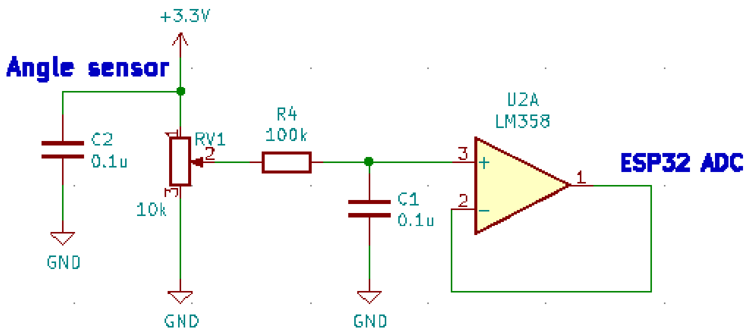

Fig. 6.3. Schematic of the Data Acquisition System

7 Setup of the 1-DoF Copter System

Fig. 7.1 shows the assembly of the system. Initially, the ESP32 is connected to the PC and the peripherals are interfaced to the ESP32. The microcontroller is programmed via USB [11]. After all the connections are made, supply is turned on and system is let free for functioning. The closed loop control of the copter system is done using the PID algorithm modified as shown before. The required setpoint and K_p, K_i, K_d, B values are entered using the Bluetooth serial terminal [12]. The current angle sampled by the microcontroller ADC and the setpoint entered by the user are used to compute the error signal. The PID algorithm uses the error signal to produce a PWM duty cycle which is then sent to the motor driver to control the motor voltage. The system parameters are updated continuously and displayed on the OLED display.

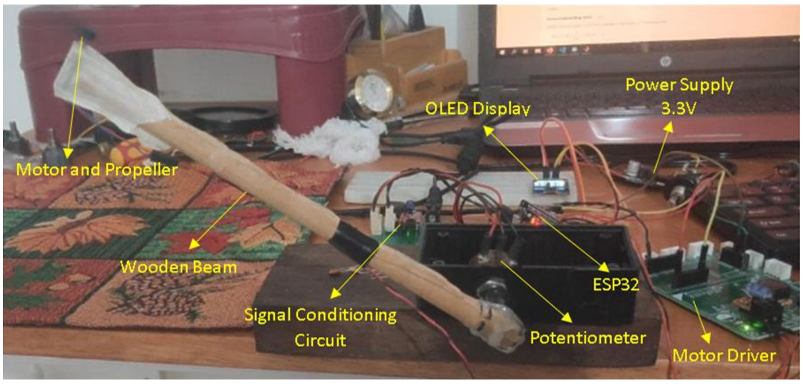

Fig. 7.1. Complete 1-DoF Copter System Setup

8 Results

The hardware and the software subsystem for the 1-DoF Copter has been designed and implemented. The system was successful at hovering at any angle given by the user within the range -90° to 50°. The system was able to self -correct and had satisfactory settling times and overshoot when disturbances were applied at various hover angles using the following gains:

PID Parameters	
K_p	0.8
K_i	0.45
K_d	0.08
B	0.55

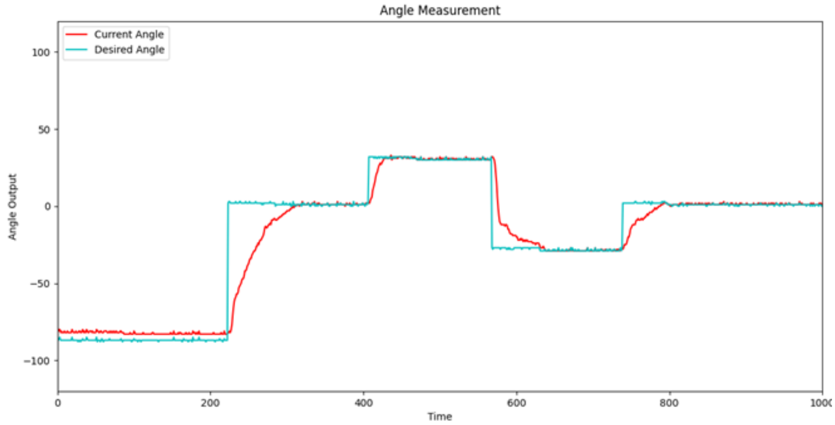

Fig. 8.1. PID Tuned Response

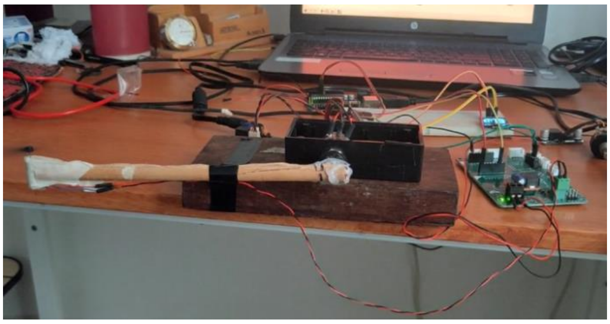

Fig. 8.2. Copter Hovering at an Angle of 0°

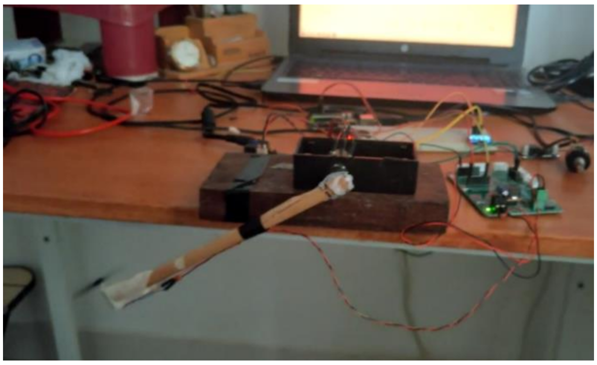

Fig. 8.3. Copter Hovering at an Angle of -30°

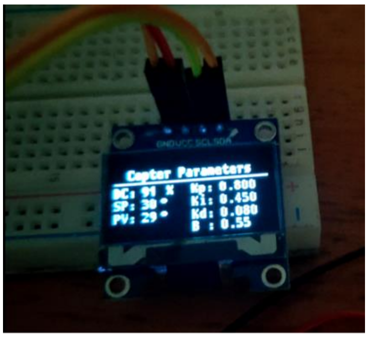

Fig. 8.4. OLED Display for a User Input of 30°

Fig. 8.5. Commands Given to the System

9 Conclusion

The 1-DoF copter system was modelled using classical mechanics and a linearized model was obtained. The mechanical setup, data acquisition system, display system and motor driver system were designed and implemented in hardware [13]. The microcontroller was programmed to run an RTOS architecture to control the hardware subsystem [14]. The PID algorithm was modified and implemented within the RTOS architecture.

The designed system performed well in the linear region of operation (± 30°) and could stabilize between -90° and 50° and "on the fly" tuning parameter changes can be performed on the system [15]. The system was found to stabilize quickly under external disturbances with an average settling time of 3s (within 5% of final value). The system was observed to attain stability with an average overshoot of 2° within the linear region of operation.

References

1. Halliday, David, Robert Resnick, and Jearl Walker, “Simple Harmonic Motion”, in Fundamentals of Physics, 8th ed. New York: Wiley, 1997, pp. 387 – 402.
2. Dorf, Richard C., and Robert H. Bishop, “Feedback control system characteristics”, in Modern Control Systems, 12th ed. Upper Saddle River, NJ: Prentice Hall, 2001, pp. 234 – 303.
3. Scherz, Paul and Monk, Simon, “Microcontrollers”, in Practical Electronics for Inventors, 4th Edition. New York: McGraw-Hill Education, 2016, pp. 880-883.
4. D. Tilbery and B. Messner, Control Tutorials for MATLAB and Simulink - Home, 2011. [Online]. Available: https://ctms.engin.umich.edu/CTMS/index.php?aux=Home. [Accessed: 17-May-2021].
5. C Feng, “IDF Programming Guide”, ESP. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/. [Accessed: 17-May-2021].
6. P. Corporation, “Pololu - 9. Dealing with Motor Noise”, Pololu Robotics & Electronic [Online]. Available: https://www.pololu.com/docs/0J15/9. [Accessed: 17-Jul-2021].
7. TOSHIBA, “TOSHIBA Bi-CD Integrated Circuit”, TB6612FNG datasheet, June. 2007.
8. M. King, “Process control: A practical approach”, Southern Gate, Chichester, West Sussex, United Kingdom: Wiley, 2016.
9. T. Wescott, “Applied Control Theory for Embedded Systems”, Burlington: Elsevier Science, 2011.
10. “FreeRTOS API categories,” FreeRTOS, 08-Jan-2020. [Online]. Available: https://www.freertos.org/a00106.html. [Accessed: 18-Jul-2021].
11. Abara, Daniel Nsor, “The development of a model and controller of a vertical take-off and landing (VTOL) system Using LabVIEW and Simulink”, The University of Manchester, 2015.
12. R. Mane, K. A. Mahapatro and A. D. Gundecha, "Single Degree of Freedom Helicopter Model: Laboratory Setup Design," 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), 2019, pp. 1-6.
13. X. Yang and X. Zheng, "Adaptive NN Backstepping Control Design for a 3-DOF Helicopter: Theory and Experiments," in IEEE Transactions on Industrial Electronics Vol. 67, no. 5, pp. 3967-3979, May 2020.
14. S. Chaudhary and A. Kumar, “Control of twin rotor mimo system using 1-degree-of-freedom pid, 2-degree-of-freedom pid and fractional order pid controller,” in Proceedings of Third International Conference on Electronics Communication and Aerospace Technology (ICECA), Gorakhpur, India, 2019.
15. A. Kumar and B. Jayaraman, “Modelling and Control of a 1-DoF Helicopter Model using LabView”, IIT Kanpur, April 2017.