Balancing a Brushless Bicopter using a Visual Sensor

Ahmed Mady, Ahmed Yasser, Omar Mohsen, Mostafa Kashaf, Mohammad Magdy

Abstract

The bicopter is a two-rotor aerial vehicle, characterized by its ability to maintain stable flight by actively adjusting the angle of its central bar horizontally. By leveraging the powerful computational capabilities and the versatility of the Raspberry Pi, the system achieves enhanced responsiveness and adaptability. The webcam captures images of the surrounding environment and processes them using image processing algorithms techniques such as edge detection is employed to identify the bicopter's central bar and determine its orientation relative to the horizontal plane. The integration of the Raspberry Pi and the webcam result in a robust and cost-effective solution for achieving self-balancing flight.

I. INTRODUCTION

A flying vehicle is not a novel engineering product. Yet, the rapid growth of unmanned aerial vehicles (UAVs) has made a research on the subject of flying vehicles a global hotspot today. One of the difficult difficulties has been reducing the number of rotors in a rotor-type UAV. One of the examples with only two rotors is the bicopter. As compared to other typical rotor-type UAVs with more rotors, it has a longer flight time. A reduction in the number of rotors can reduce unit time power demand and, as a result, improve flight time. However, because of the complexity of the mathematical dynamics model, controlling a bicopter system is not a simple process. To maintain the desired trajectory, the designed controller must be capable of stabilizing the bicopter with an acceptable response time while also overcoming the oscillation effect. To accurately stabilize the bicopter, a camera is used to find the angle of inclination of the birotor.

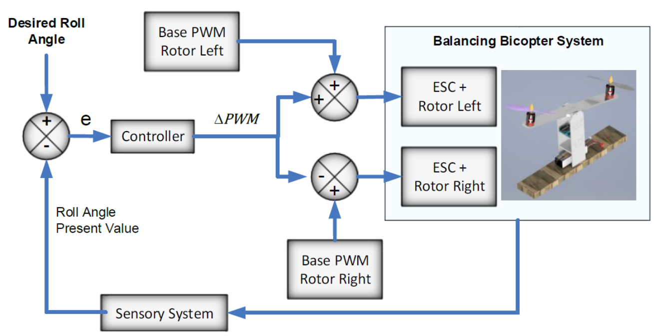

Fig. 1. The flow diagram of the system

II. METHODOLOGY

A. Workflow

To control the bicopter, a desired angle is inputted, and according to this angle, the speed of both motors is adjusted as they rotate in directions opposite to each other. Below each propeller, a colored circle is attached to track the error between the pitch and the desired angle using a visual sensor, the camera. The circles will be of the color blue and a threshold will be applied to grab only the color needed. Then segmentation and edge detection will be applied to track the circle and draw a line connecting both and measuring the angle. This measured error angle is then fed from the camera into the control loop as feedback and the correct speed is sent to the ESC by the computer of the two brush-less motors to achieve the required angle desired. Figure 1 shows the flow diagram of the bicopter system. The sensory system is the Sony Widescreen HD 720p webcam.

B. Components

C. Circuit Design

The schematic circuit diagram of the birotor system is shown in Figure 3 which consists of the two brushless motors, the two ESCs, the Arduino, and the power source. The camera is the feedback element and is connected to the computer directly. The flow of data, Figure **??** is first from the camera to the computer which applies the image processing algorithm which calculates the angle of the bicopter which is then sent to the Arduino using a USB connection and into the control loop. The Arduino then calculates the required speed of the motors.

TABLE I
LIST OF COMPONENTS

Component	Number	Purchase Location	Price
Raspberry Pi 4B	1	Future Electronics	3370E£
Electronic Speed Controllers	2	Future Electronics	205E£
Brushless Motor	2	Amazon Eg	388E£
Propeller	2	Future Electronics	150E£
12V 20A Power Source	1	Amazon Eg	220E£
Sony Webcam	1	Amazon Eg	599E£

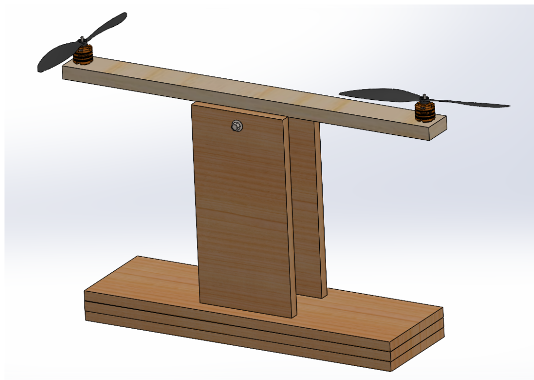

Fig. 2. Solidworks Assembly 1

D. Image Processing

Image processing is the process of converting an image to a digital format and then executing various operations on it to extract valuable information. When specific specified signal processing methods are used, the image processing system typically interprets all images as 2D signals. The figure below 4 depicts the original photo taken by the camera prior to any image processing. Image processing tasks performed were translation and scaling, then converting the normal RBG image to LAB and HSV color spaces then grayscale where contrast enhancement and darkening were applied.

1) Intensity Level Transformation

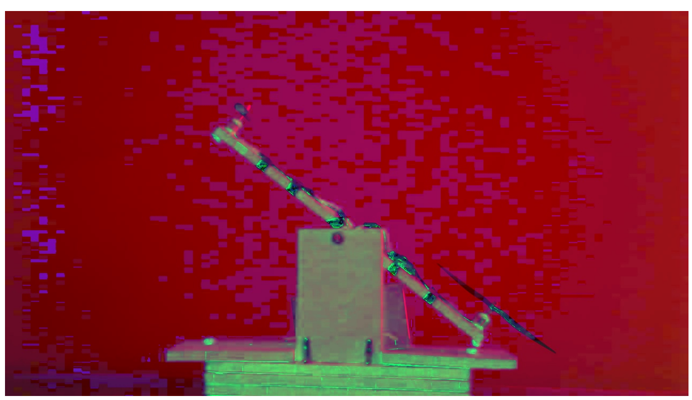

Fig. 5. The Image After Applying HSV Color Space

2) Morphological Operations
Morphological operations are a set of operations that process images based on shapes. They apply a structuring element to an input image and generate an output image.

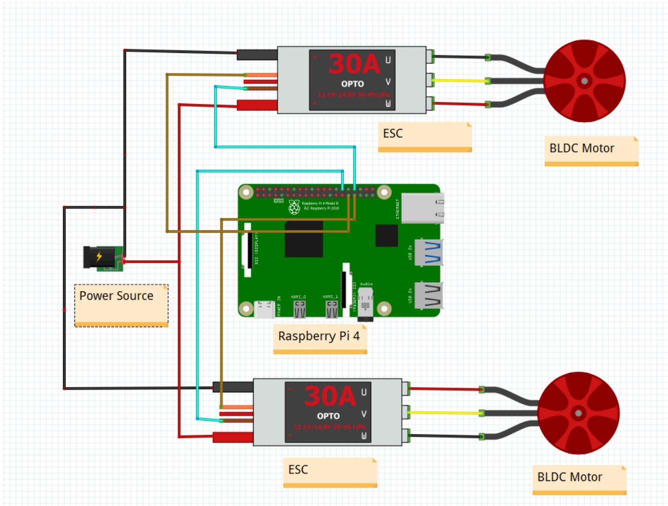

Fig. 3. Circuit diagram of the birotor system

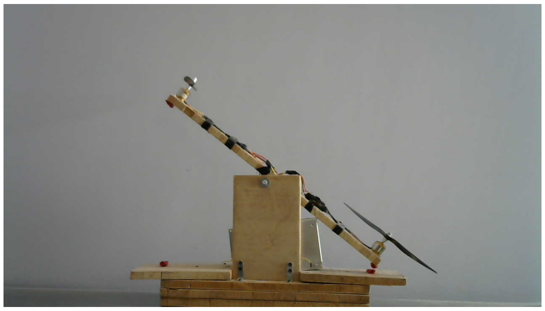

Fig. 4. Orignal Image before any transformations

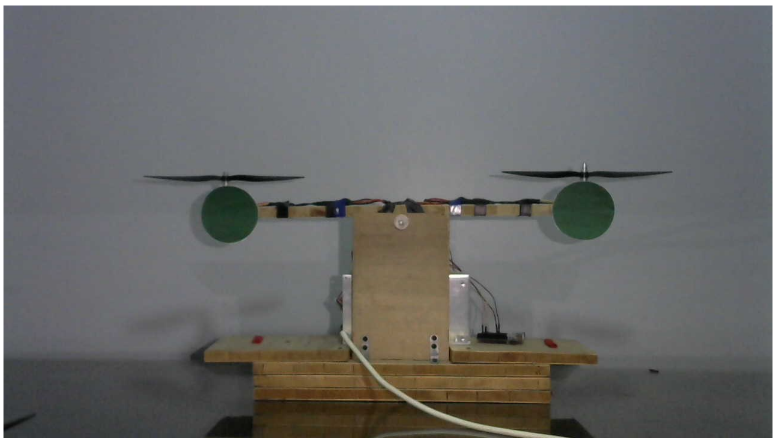

Fig. 6. The original image after adding colored figures, circles, under each motor

Next, erosion and dilation are applied. Basics of Erosion: Erodes away the boundaries of the foreground object, Used to diminish the features of an image. Basics of dilation: Increases the object area, Used to accentuate features. Both are applied using a 5 x 5 kernel.

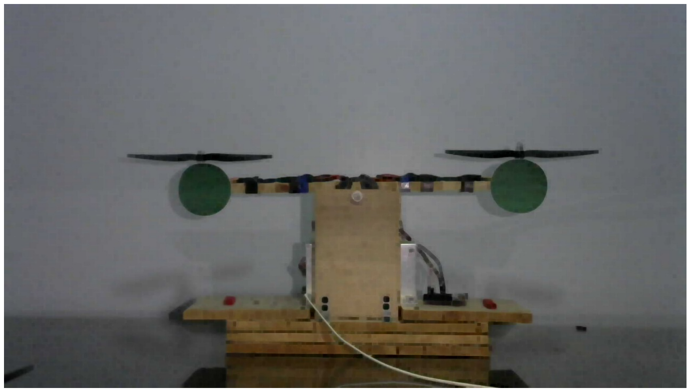

Fig. 7. The image after applying erosion using a 5 x 5 kernel

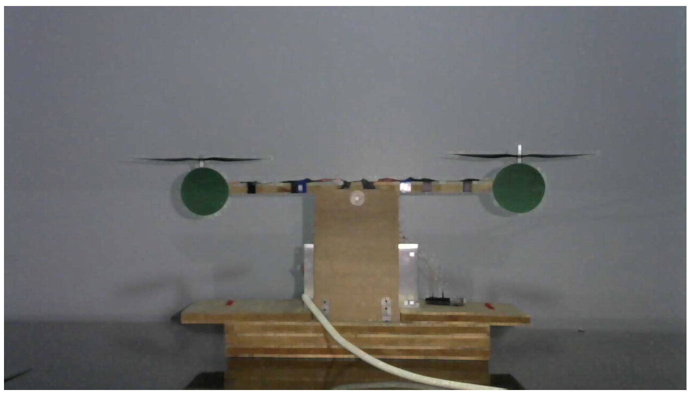

Fig. 8. The image after applying dilation using a 5 x 5 kernel

3) Edge Detection Edge detection is an image processing technique for finding the boundaries of objects within images. It works by detecting discontinuities in brightness. In this paper, edge detection is implemented using different methods applied on the, 5 x 5 Gaussian blurred image:
   a) Canny Edge Detection

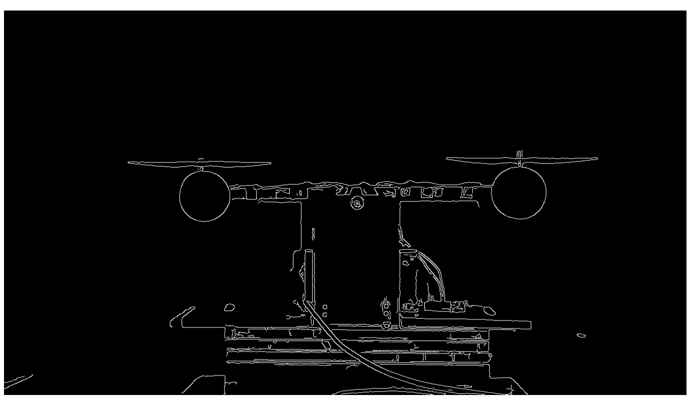

Fig. 9. The image after Canny edge detection using a threshold range 0 to 60

4) Segmentation
   a) Color Segmentation The first type of segmentation applied was to color segment the green circle which will be used later for angle calculation.

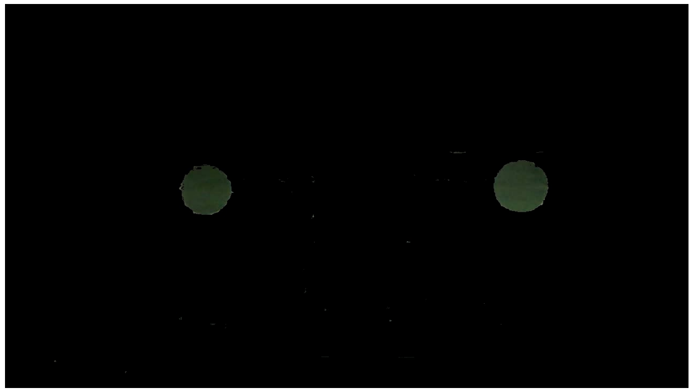

Fig. 10. The image after applying the green segmentation

b) Watershed Algorithm The watershed algorithm is used for segmentation in some complex images if we apply simple thresholding and contour detection then will not be able to give proper results. The watershed algorithm is based on extracting sure background and foreground and then using markers will make a watershed run and detect the exact boundaries. This algorithm generally helps in detecting touching and overlapping objects in the image. For markers, it can be user-defined like manually clicking and getting the coordinates and also using some defined algorithms such as thresholding or any morphological operations. Due to the presence of noise, we can't apply watershed algorithms directly.

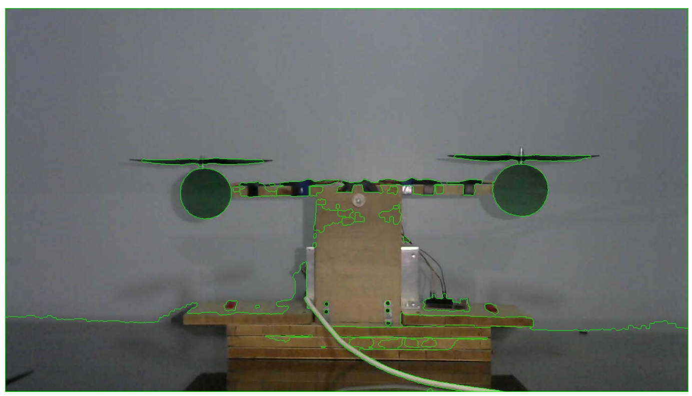

Fig. 11. The image after applying the watershed algorithm

E. Control

The dynamical equation is shown in Equation 1:

$$\ddot{\theta}_b = \frac{M_2 - M_1 * g * \frac{L}{2} - b\theta_b - a\theta_b(\omega_{m_2} + \omega_{m_1}) + 0.00125L * (\dot{\omega}_{m_2} + \dot{\omega}_{m_1})}{MomentofInertia} \tag{1}$$

$$TransferFunction : \frac{0.6126s^3 - 1.359s^2 + 28.81s + 5.315}{s^4 + 2.27s^3 + 19.89s^2 + 14.84s + 2.74} \tag{2}$$

After manually tuning the PID the final values reached were:

$$k_p = 1, k_i = 2, k_d = 0.3 \tag{3}$$

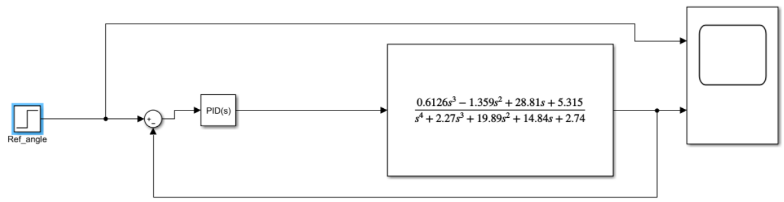

Fig. 12. SIMULINK simulation of the transfer function

a) Closed Loop Response In this section, a simple closed loop is tested. Python communicates with Arduino to send a control action to the 2 BLDC motors based on the image processing signal by establishing a serial communication. First in the image processing code in python, the frame captures from the external camera then converts the frame into gray-scale. After converting, numpy mean method is used to calculate the average of the intensity levels of the pixels in the image. If the pixels have a mean higher than 100, also called a bright frame, the left motor is actuated based on a signal given from python to the Arduino. If the mean is less than 50, dark frame, the right motor is actuated.

F. Actuation

At the beginning of the code, the libraries are imported. Then, the ranges of green color are HSV are defined from low to the max values based on trial and error which then applies a mask to detect only green color on the video capture frame which in turn is used to detect only the largest 2 circles only of green colors. The centers of the circles are used then to calculate the slope of the bi copter bar which is our error signal. This error signal is taken as input to the PID controller and then taken as input to the PWM output to the motor.

III. RESULTS

The results of this study show that it is possible to balance a bicopter around a zero angle using a camera and a Raspberry Pi. The system was able to maintain a stable hover for up to 30 seconds, even in the presence of disturbances. The system was also able to track the moving target circles, such as a person's head, and maintain a balancing angle.

A. KPIs

Execution Time: On a Raspberry Pi 4B+, the average execution time of the code you provided is about 100 milliseconds. This means that the code can process one frame of video every 100 milliseconds. Accuracy: The accuracy of the code in detecting the angle of the bicopter is very high as the 2 motors never turn off but are always on and at varying speeds. Stability: The stability of the bi-copter was taken into consideration when designing the control law. Responsiveness: The responsiveness of the code to changes in the angle of the by-copter.

IV. CONCLUSION

In conclusion, the utilization of a camera and Raspberry Pi 4B+ in combination with PID control provides an effective solution for controlling a bicopter. By leveraging the camera's ability to capture real-time images and the Raspberry Pi's processing power, the system can extract essential information about the bicopter's orientation and position. This information is then used to calculate appropriate control signals using PID control algorithms, ensuring stable and precise flight.

The camera's input is processed by the Raspberry Pi, which analyzes the images and extracts relevant features to estimate the bicopter's position and orientation. This data is then fed into the PID control algorithm, which calculates the necessary adjustments to the bicopter's motor speeds or control surfaces.

PID control provides a robust and widely used method for adjusting control signals based on the error between desired and measured values. It continuously evaluates and adjusts the output to minimize this error, resulting in smooth and accurate control of the bicopter.

The integration of these technologies enables autonomous or semi-autonomous flight of the bicopter. It opens up possibilities for various applications, such as aerial surveillance, monitoring, and inspection tasks. Additionally, the flexibility of the Raspberry Pi platform allows for easy customization and expansion, facilitating the implementation of advanced features and functionalities.

Overall, the combination of a camera, Raspberry Pi 4B+, and PID control offers a reliable and adaptable system for controlling a bicopter. It represents a promising avenue for further exploration and development in the field of unmanned aerial vehicles and autonomous systems.

V. Videos

Fig. 13. QR Code containing the videos of the hardware

References

[1] Apriaskar, E., Fahmizal, F., Salim, N. A., & Prastiyanto, D. (2019). Performance Evaluation of Balancing Bicopter using P, PI, and PID Controller. Jurnal Teknik Elektro, 11(2), 44-49.

[2] Siddhartha, K. (2018). A novel bi-rotor configuration and its control. IFAC-PapersOnLine, 51(1), 456-461.

[3] Qin, Y., Xu, W., Lee, A., & Zhang, F. (2020). Gemini: A compact yet efficient bi-copter uav for indoor applications. IEEE Robotics and Automation Letters, 5(2), 3213-3220.

[4] Jain, P., & Sangwan, V. (2020, July). Generation and Control of Impulsive Forces by a Planar Bi-Rotor Aerial Vehicle through a Cable Suspended Mass. In 2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM) (pp. 868-873). IEEE.

[5] Li, Y., Qin, Y., Xu, W., & Zhang, F. (2020, July). Modeling, identification, and control of non-minimum phase dynamics of bi-copter uavs. In 2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM) (pp. 1249-1255). IEEE.

[6] Uğurlu, H. İ. (2019). A Comparative Study of learning-based control policies and conventional controllers on 2D bi-rotor platform with tail assistance (Master's thesis, Middle East Technical University).

[7] Taherinezhad, M., Ramirez-Serrano, A., & Abedini, A. (2022). Robust trajectory-tracking for a bi-copter drone using indi: A gain tuning multi-objective approach. Robotics, 11(5), 86.

[8]