DEVELOPMETNT OF FUZZY-PID CONTROL SYSTEM FOR MULTYROTOR UAV USING FACE RECOGNATION ALGORITHM

Avetik Baskovchyan
Postgraduate student
EUA, Chair of Information Technologies and Applied Mathematics
avik.baskovchyan@gmail.com

Abstract

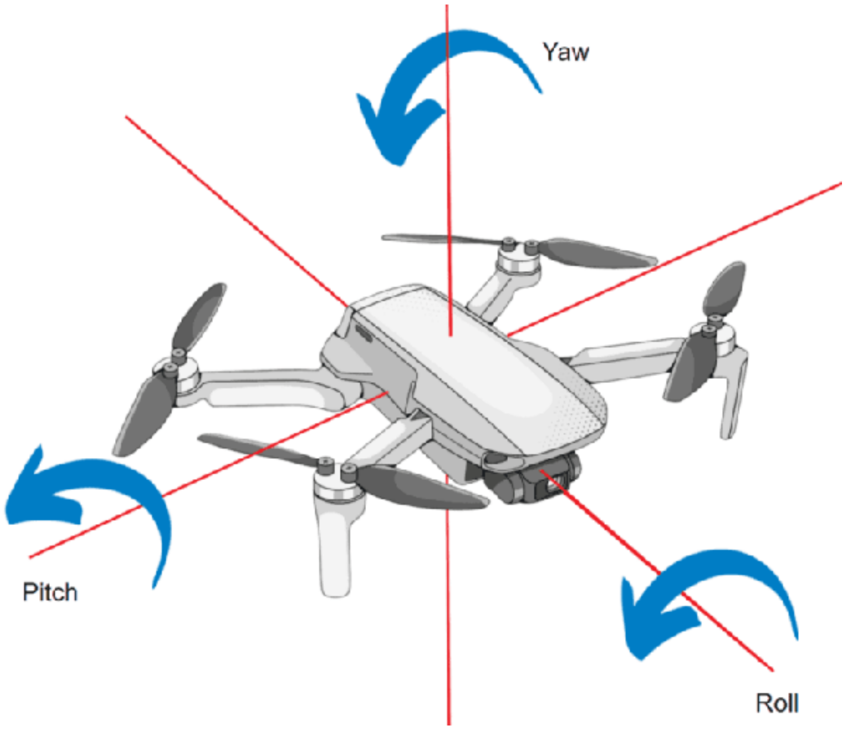
This article presents a model of the control system of a multi-rotor drone, which is a combination of 2 models, Proportional Integral Derivative and Fuzzy logic. In order to obtain the most accurate values, instead of modeling, a real multi-rotor drone, DJI TELLO, was chosen, on which the designed control system was used. The main advantage of this system lies in the development of an algorithm for the joint application of not only PID but also Fuzzy logic controllers, using the values taken by the Open CV library with the image captured by the quadcopter camera as an input value. This article presents a drone control model that solves the problem of automatic control of one of the most popular drones, the DJI TELLO drone, by taking information received from the camera as an input value. Based on the sensor data, a hybrid control system is built that uses a fuzzy logic controller to tune the parameters of the PID controller. The principle of work consists in determining the position of the face found using sensor values in a matrix system with a selected surface, according to which the created membership functions provide the required coefficients to the Proportional Integral Derivative controller with a precise calculation, with a much higher degree of accuracy compared to a separate Proportional Integral Derivative controller with.

Keywords: Drone, PID, Fuzzy logic, Automation system, Control system, Face recognizes

Introduction

Currently, unmanned aerial vehicles are widely used in various fields, one of which is the application of face recognition and the obtained results through the data provided by the camera. It can be used both in everyday life and in military work, for pursuit and search. It is necessary to have a drone control system that can work in real time with information received from sensors, taking into account the tasks assigned to it. The solution to a similar problem is provided by the control and use of a multi-rotor unmanned aerial vehicle (UAV) using programming languages. This article presents a hybrid Fuzzy-PID model of UAV control in Python, which rightly gives more accurate values than systems with PID and Fuzzy logic controllers separately [Carvajal et al, 249-270]. In the article, the DJI RYZN TELLO [Dhaval Joshi Dipankar Deb S. M. Muyeen, 9-11] drone was selected as a test UAV, through which the performance results of the designed algorithm were presented under real conditions. The Python OpenCV libray was chosen as a working environment, based on the results obtained, the Fuzzy-PID controller automatically determined the direction of movement of the quadrotor according to the following 3 parameters: roll, pitch, yaw. roll, pitch, yaw angles for a quadcopter (Figure 1).

Figure 1



roll, pitch, yow angels for quadrotor

During the execution of the work, in addition to the Python language, the Simulink package of the MATLAB software environment and the Fuzzy Toolbox [Putri et al., 65-71] were also used to obtain the graphic representation of the input and output results for the Fuzzy logic controller, whose parameter values are obtained from Python code, based on real-time drone flight.

literature review

The controllers used for drone control are various, one of them is the hybrid Fuzzy-PID controller, which is presented in [Kuantama et al., 519-532], using separate PID and Fuzzy-PID controllers, where better results are obtained for the Fuzzy-PID controller. A similar study, but for another drone AR.drone model, is presented in [Prayitno et al., 1-6], where the time difference and overshoot difference of PID and Fuzzy logic controllers is discussed. For both of the articles, overshoot (about 20%) and settling times(3-5s) are wors than in our case.

Problem setting

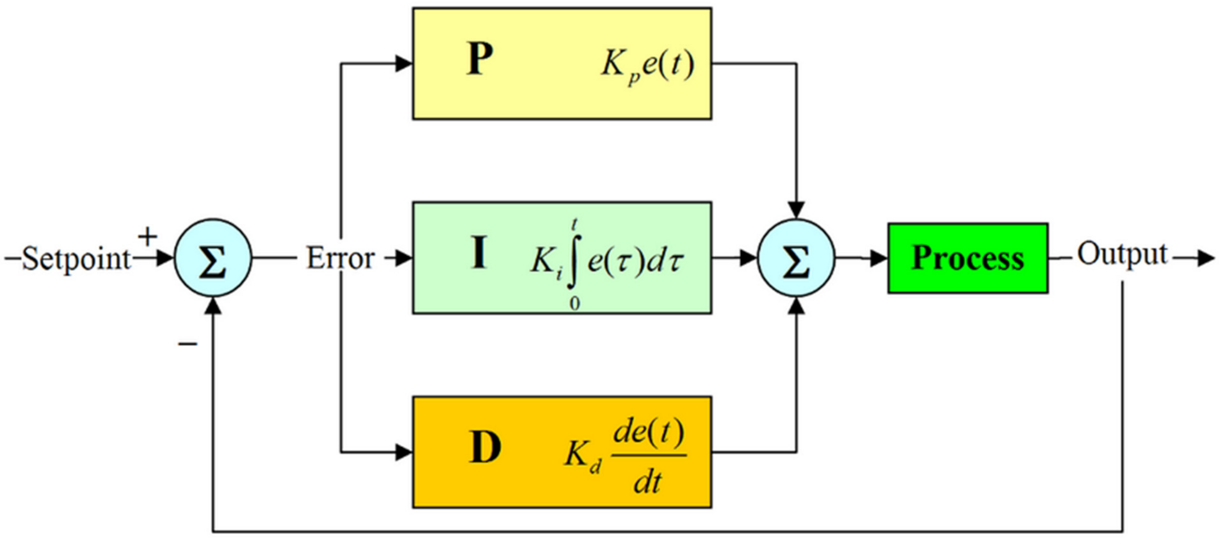
Taking into account the popularity of the DJI TELLO drone, the cheapness of the price and the parameters corresponding to the price, it becomes a problem to use the drone in the economy and many other areas. The article aims to create a more accurate control system that will be applicable for a specific DJI TELLO drone, using the Python programming language, which is acceptable for the DJI TELLO drone.

Scientific novelty: This article presents a real model of the UAV for DJI TELLO drone, that is, a control system was built based on the real values, in which the membership functions of the Fuzzy-logic controller was adjusted, through which the individual parameters of the PID system were controlled and the drone was operated with the created system in real conditions. The presented model is innovative, because a hybrid control system was studied and created, which is applicable in real life and has not been found in the studied literature.

Fuzzy-PID controller

As mentioned, a Fuzzy-PID controller was used to control the quadcopter, which is an improvement of the results of the PID controller through Fuzzy logic. The basic scheme of the PID controller is shown bellow (Figure 2).

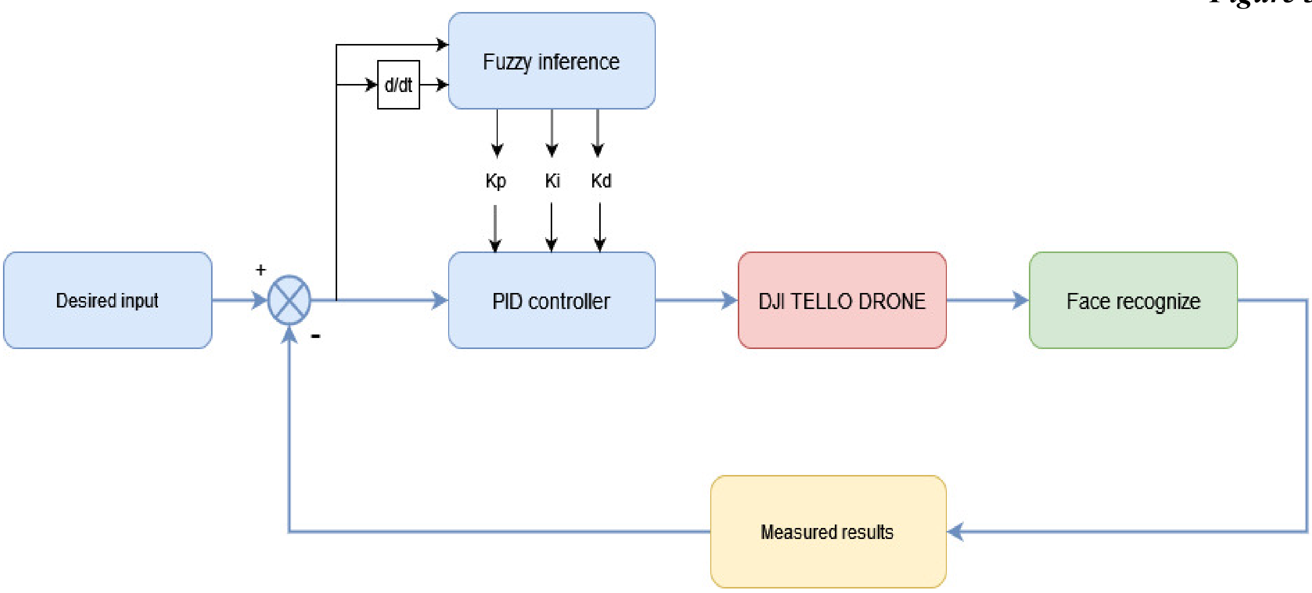
Figure 2



PID controller's basic diagram

Where P,I,D are the coefficients of the corresponding proportional, integrator, differentiator components, and e is the function of time[Heong Ang, G. Chong, Yun Li, 559-576]. As can be seen from Fig. 2, the elements P, I, D of the PID system have coefficients Kp, Ki, Kd, the automatic calculation of which is the function of the Fuzzy logic controller. The block diagram of the general control system is shown in (Figure 3).

Figure 3

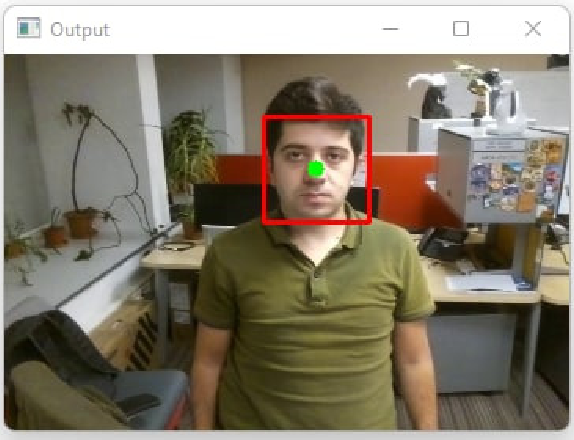


Block diagram of the designed system

As can be seen from Figure 3, the values obtained as a result of the operation of the feedback system [Gasparyan, 343-351], entering the Fuzzy logic controller with the appropriate error, automatically calculates the necessary Kp, Ki, Kd coefficients for the PID and provides the PID controller. The variables necessary for the fuzzy controller are obtained from the OpenCV [Culjak, 59-65] library, the data obtained from camera is presented in (Figure 4).

Figure 4

```
Center [202, 72] Area 4096
Center [201, 71] Area 4225
Center [200, 71] Area 3969
Center [200, 71] Area 4225
Center [200, 71] Area 3969
Center [199, 71] Area 3969
Center [199, 72] Area 4225
Center [198, 72] Area 4096
Center [198, 73] Area 4225
Center [198, 73] Area 3844
Center [197, 73] Area 3844
Center [197, 72] Area 3844
Center [197, 72] Area 3721
Center [197, 72] Area 3844
Center [197, 73] Area 3969
Center [198, 73] Area 3844
Center [198, 73] Area 4225
```



Face recognition results, area and center coordinates

The results shown in Fig. 4 show the data obtained by the quadcopter's camera, based on which the Fuzzy system observes the location of the case: near, far, left and right, performing appropriate actions by changing the direction and speed of the quadcopter as necessary. Using the camera, face recognition is obtained, as shown in Figure 4, a square is built for the recognized face, the values of which are represented by the coordinates of the center and surface. Parallel to the movement of the recognized face, the presented parameters are changed by transferring them to the controller, which determines the direction and speed of the drone's movement. The code for face recognition and obtaining necessary values (area, center) is presented in (Figure 5).

Figure 5

```python
def findFace(img):
    faceCascade = cv2.CascadeClassifier("Resources/haarcascade_frontalface_default.xml")
    imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(imgGray, 1.2, 8)

    myFaceListC = []
    myFaceListArea = []

    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 0, 255), 2)
        cx = x + w // 2
        cy = y + h // 2
        area = w * h
        cv2.circle(img, (cx, cy), 5, (0, 255, 0), cv2.FILLED)
        myFaceListC.append([cx, cy])
        myFaceListArea.append(area)
    if len(myFaceListArea) != 0:
        i = myFaceListArea.index(max(myFaceListArea))
        return img, [myFaceListC[i], myFaceListArea[i]]
    else:
        return img, [[0, 0], 0]
```

Face recognition

Figure 6 shows the algorithm of face recognition and movement direction determination using Python code.

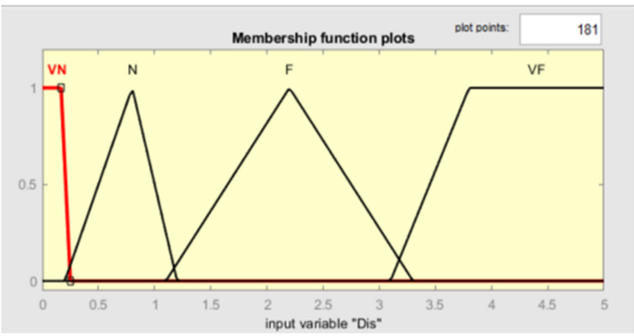
Figure 6

```
def trackFace(info, w, pid, pError):
    area = info[1]
    x, y = info[0]
    fb = 0
    error = x - w // 2
    speed = pid[0] * error + pid[1] * (error - pError)
    speed = int(np.clip(speed, -100, 100))
    if area > fbRange[0] and area < fbRange[1]:
        fb = 0
    elif area > fbRange[1]:
        fb = -20
    elif area < fbRange[0] and area != 0:
        fb = 20
    if x == 0:
        speed = 0
        error = 0
    return error
```

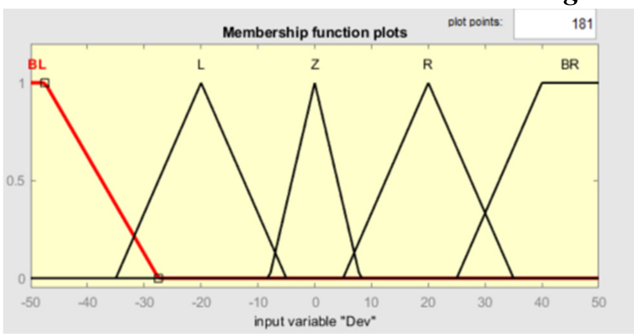
Determining the direction and speed of the drone

Based on the values obtained as a result of the work of the codes presented in Fig. 5 and Fig. 6, a Fuzzy logic system was created with the corresponding membership function, whose graphic representation using the Fuzzy GUI tool of the MATLAB software is shown in (Figure 7 a, b, c).

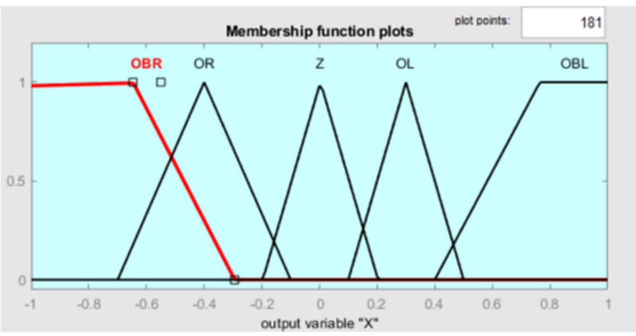
Figure 7



a



b



c

Fuzzy membership functions

Figure 7 a,b,c shows the defines membership functions of 2 inputs and 1 output for fuzzification.

For the membership functions shown in Fig. 7 (2 inputs and 1 output), the rules presented in were created table (Table 1).

Table 1

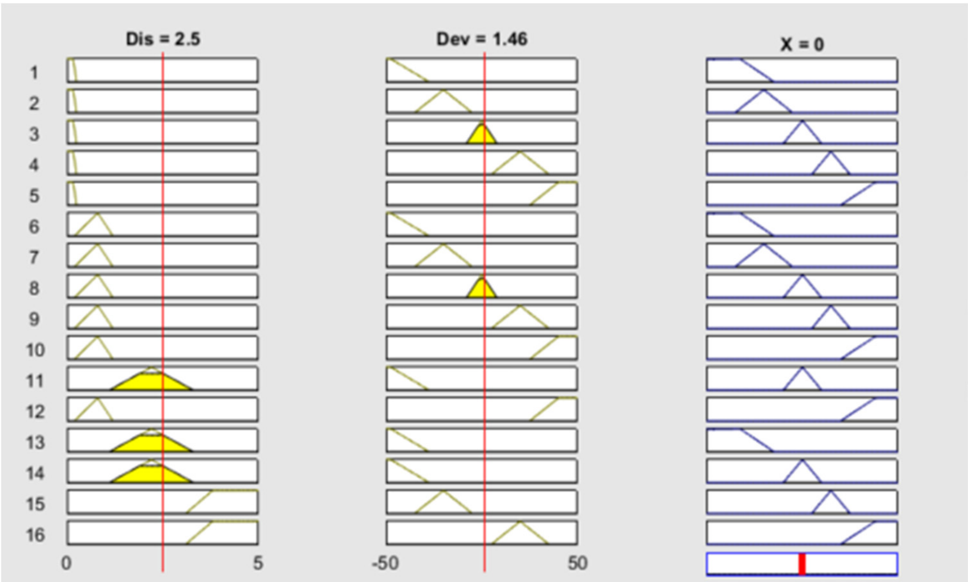
I1/I2	**BL**	**L**	**C**	**R**	**BR**
VN	OBR	OR	OR,OL,Z	OL	OBL
N	OR	OR	OR,OL,Z	OL	OL
F	OBR	OR	OR,OL,Z	OL,Z	OL
VF	OBR	OR	Z	OL	OBL

Fuzzy Inference system rules

where VN-very near, N-near, F-far, L-left, Z-zero, R-right, B-big, BR-big right

The graphical values obtained for the rules presented in Table 1 are shown below (Figure 8, Figure 9).

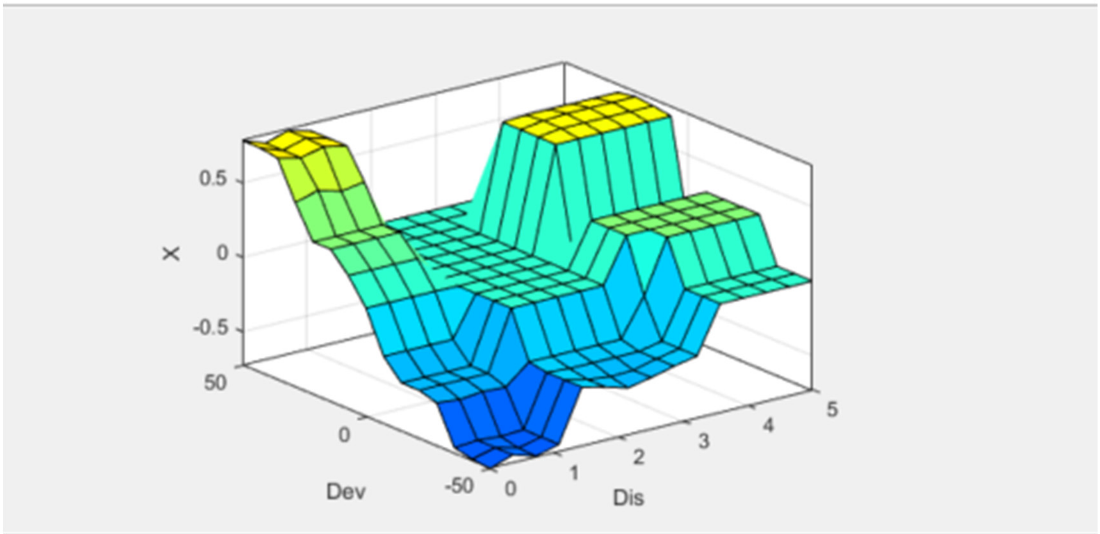
Figure 8



Graphic view of selected rules

The rules are created according to the logical approach, if Input_1 VN & Input_2 BL then OUT is BR and all other rules follow the same principle:

Figure 9



Surface diagram of membership function and rules

Fuzzy-PID controller Fuzzy and PID controller connection through the code is shown in (Figure 10).

Figure 10

```
self.P=self.kp*self.error
self.D=self.kd*self.eder
self.I=self.ki*self.esum
pid=self.P+self.I+self.D
```

P,I,D parameters with python code

Where: P,I,D are the searchable values of Proportional, Integral, Differential variables respectively, Kp-comparative coefficient, Ki-integrating component coefficient, Kd-differentiating component coefficient, error, esum-integral error, eder-process variable differential.

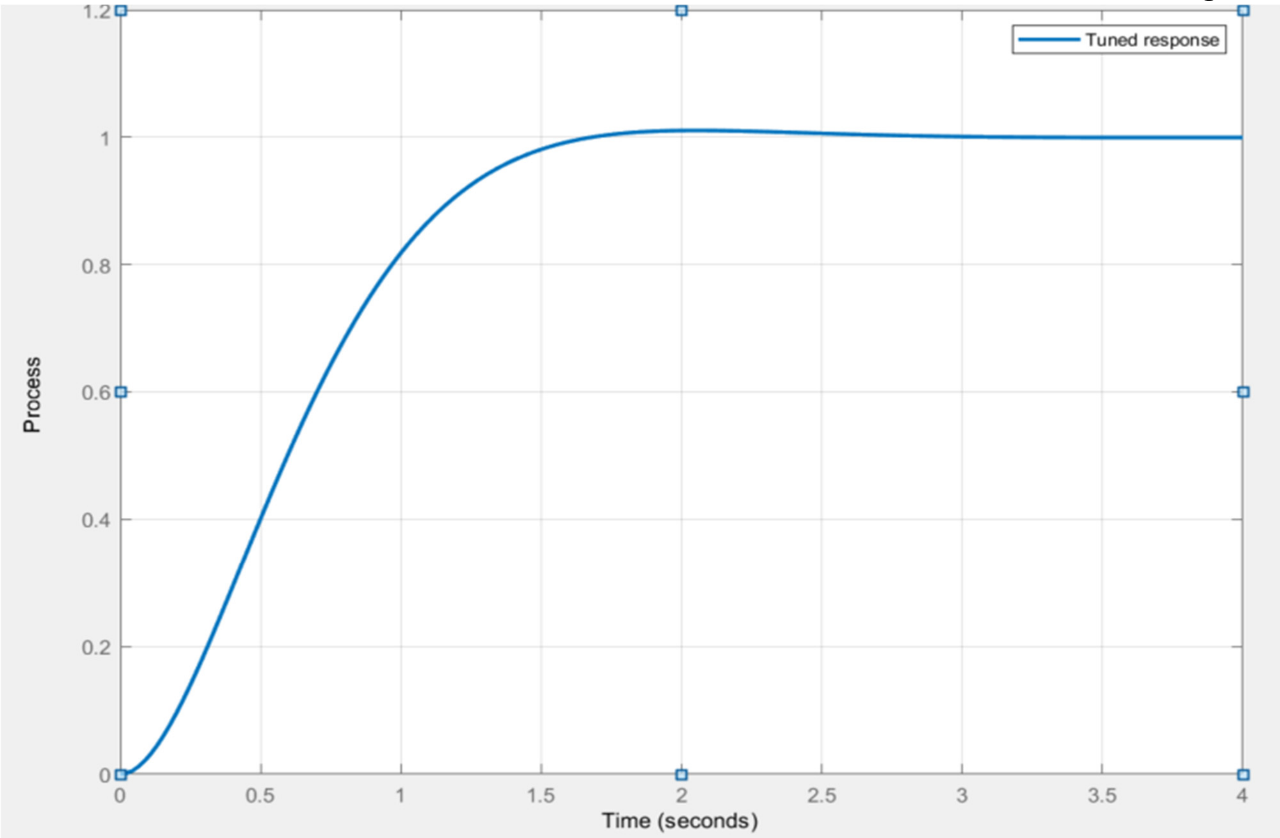
Results obtained

As already mentioned, the designed control system receives appropriate signals from the camera, i.e. the fixed distance from the quadcopter, the latter's position with respect to the center of the drone, and calculates the necessary Kp, Ki, Kd coefficients for the basic PID system using the Fuzzy logic controller. The results generated during the test flight are shown below.

Kp = 2.1, Ki = 0.899, Kd = 1.22

The output graph obtained as a result of the work of the code is shown in (Figure 11).

Figure 11



Time dependence of the process

where the dependence of the drone's flight height on time is presented, it shows that in 1.5s the drone was adjusted, reaching the pre-specified height of 1m.

The values corresponding to the graph shown in Fig. 11 are presented in (Table 2).

Table 2

Rise Time	0.971 sec
Settling Time	1.49 sec
Overshoot	1.06%
Peak	1.01
Phase Margin	70.2 deg
Closed-loop stability	Stable

Results obtained

Table 2 shows the performance results of the drone after tuning. Considering the fact that the control system presented in the article was created specifically for the DJI TELLO drone, it is not realistic to make a comparison with other models due to parameter differences. The presented system is hybrid, using fuzzy logic for the value adjustment of the PID controller. In general, it follows from the values presented in the literature review section that the obtained overshoot and settling time values are sufficient to consider the proposed problem solved.

From the values shown in Table 2, it becomes clear that the values of the control system created on the real Drone correspond to the expected values, in particular, very fast adjustment time, absence of Overshoot [Mystkowski, 1187-1196]. Thus, it can be said that the system designed using the Python language meets the requirements.

Conclusion

The combination of the control system presented in the article with the face recognition algorithm meets the expected values, automatically providing precise control for the quadcopter. Since the results of the designed model are very good in real use (overshot 1.06%, settling time 1.49sec), it can be confidently said that the work performed meets the requirements. The work was done with the help of Python, OpenCV, Matlab tools with the example of a real four-screw as a result of creating a system with practical application.

List of literature

1. J. Carvajal, G. Chen, H. Ogmen, "Fuzzy PID controller: Design, performance evaluation, and stability analysis", Information Sciences, Volume 123, April 2000, Issues 3–4, Pages 249-270,
2. Dhaval Joshi Dipankar Deb S. M. Muyeen "Inverted Docking Station: A Conceptual Design for a Battery-Swapping Platform for Quadrotor UAVs", February 2022, pp 9-11
3. Anggie Prahas Putri, Haris Rachmat, Denny Sukma Eka Atmaja "Design Selection of In-UVAT Using MATLAB Fuzzy Logic Toolbox", International Conference on Soft Computing and Data Mining, 2017, pp 65-71
4. Oleg Gasparyan, "On Application of Feedback Linearization in Control Systems of Multicopters", In book: Advanced Technologies in Robotics and Intelligent Systems, 2020 (pp.343-351)
5. Ivan Culjak, David Abram, Tomislav Pribanic, Hrvoje Dzapo, Mario Cifrek, "A brief introduction to OpenCV" 2012, pp 59-65
6. Arkadiusz Mystkowski, "Implementation and investigation of a robust control algorithm for an unmanned micro-aerial vehicle", August 2014, Robotics and Autonomous Systems 62(8), pp 1187-1196
7. A. Prayitno, V Indrawati,I Trusulaw, "Comparison of PID and Fuzzy Controller for Position Control of AR.Drone", 2017 *IOP Conf. Ser.: Mater. Sci. Eng.* 190 012006
8. E. Kuantama, T. Vesselenyi, S. Dzitac, R. Tarca, "PID and Fuzzy-PID Control Model for Quadcopter Attitude with Disturbance Parameter", INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL ISSN 1841-9836, 12(4), 519-532, August 2017
9. K. H. Ang, G. Chong, Yun Li, "PID Control System Analysis, Design, and Technology", IEEE Transactions on Control Systems Technology 13(4), 2005, pp. 559-576.

Ներկայացվել է՝ 10.10.2022թ.
Ուղարկվել է գրախոսման՝ 02.12.2022թ.